

Electrical Monitoring of Electromechanical Systems

by

Daisy Hikari Green

B.S., University of Hawai‘i, Mānoa (2015)

S.M., Massachusetts Institute of Technology (2018)

Submitted to the

Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

© Massachusetts Institute of Technology 2022. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by.....
Steven B. Leeb
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by.....
John S. Donnal
Associate Professor, United States Naval Academy
Thesis Supervisor

Accepted by
Leslie A. Kolodziejcki
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Electrical Monitoring of Electromechanical Systems

by

Daisy Hikari Green

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Electromechanical systems provide the world’s backbone for generating and using energy. Electromechanical systems can also experience an innumerable set of failures, causing induced wear and wasted energy, or eventually a complete failure of a critical piece of equipment or system. Degradation or other faults are often associated with subtle but observable changes in electrical consumption. A nonintrusive load monitor (NILM) is a convenient tool for electrical monitoring, in which all loads connected downstream of an electrical panel are monitored with a single set of current and voltage sensors. If collated in a useful way, nonintrusive electrical data can make diagnostic information more easily attainable and improve the efficient operation of critical machines.

Ensuring correct nonintrusive identification of load operation is a challenge in varying operating conditions and fault scenarios. Most nonintrusive load monitoring research assumes that data is static over time. Also, ground truth labels are a scarce resource in industrial scenarios. Thus, a pattern classifier must train on a limited dataset not representative of long-term operation. This thesis employs an understanding of the physics and time-dependency behind changing load behavior to inform pattern classification. New statistical feature extraction techniques are presented for loads with time-varying operation. Results are demonstrated with laboratory experiments and case-studies from NILM installations onboard various marine microgrids.

Thesis Supervisor: Steven B. Leeb

Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: John S. Donnal

Title: Associate Professor, United States Naval Academy

Acknowledgments

I am deeply appreciative to the many people who have supported my Ph.D. journey. I first want to thank my advisor Professor Steven Leeb for his unceasing guidance and support throughout my time at MIT. I am very grateful for the opportunity to work in your lab, and for everything I was able to learn. I would like to thank Professor John Donnal for his continued mentorship and technical advice, and for providing the foundation on which this work is built upon. I would like to thank my thesis committee, Professor James Kirtley, Professor Les Norford, and Professor Steven Shaw, for their valuable time, feedback, and support.

I am grateful for the support of the crews of USCGCs Spencer, Escanaba, Marlin, and Thunder Bay, and USS Indianapolis. This research would not have been possible without the invaluable field data from these ships and the enthusiasm and support from the crews for our research. Financial support for this research was graciously provided by The Grainger Foundation, the Office of Naval Research Structural Acoustics program, and the Office of Naval Research NEPTUNE program.

I owe many thanks to those with whom I've had the privilege of collaborating with over the years. The work in this thesis is a result of the collaborative efforts of many. I want to thank all the Coast Guard officers: Greg Bredariol, Tom Kane, Steve Kidwell, Joey O'Connell, Brian Mills, Devin Quinn, Andrew Moeller, Isabelle Patnode, and Mike Bishop. Thank you for the hard work and effort you put into our research, including the countless ship visits, and your patience as I asked you many questions. Thank you to Peter Lindahl for your continued mentorship and guidance starting from my first year as a graduate student. Thank you to Andre Aboulian for guiding me in the ways of NILM when I started. Thank you to Katherine Camenzind for your work on the noncontact voltage sensor research. Thank you to Rebecca Agustin for your work on the adaptive classifier research. Thank you to Erik Saathoff for sharing your power electronics expertise. Thank you to Łukasz Huchel for your technical advice and lessons. Thank you to Kahyun Lee, it was a pleasure to work with you. Thank you to Aaron Langham, it has been really wonderful working with

you. I look forward to seeing you expand and build upon many of NILM ideas we worked on. Thank you to Tommy Krause for sharing your enthusiasm and knowledge of electric machines; thank you for your encouragement, time, and friendship, and for always believing in me.

Thank you to everyone else in the LEES/GEM laboratory, including Eric Ponce, Dan Monagle, Adedayo Aderibole, Kevin Kircher, Manuel Gutierrez, Spencer Shabshab, David Otten, Jessica Boles, Tony Zhang, Mike Ranjram, Mohammad Qasim, Sajjad Mohammadi, and everyone else I have been lucky to know. I am truly grateful I was able to work with such amazing and kind people.

I want to express gratitude to my professors and mentors. I want to especially thank Professor Anthony Kuh at the University of Hawai'i for encouraging me to go to graduate school and for the continued support. Thank you to Professor Magdy Iskander and Professor Zhengqing Yun at the University of Hawai'i for providing me with my introduction to research.

I am very thankful for all the friendships I made here, and also for my friends from Hawai'i. Mahalo to my 'ohana, especially my parents, sister, grandparents, aunts and uncles, for your unconditional love and support. I love you all very much.

Thank you,

Daisy Green

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

1	Introduction	25
1.1	Nonintrusive Load Monitoring Background	26
1.2	Shipboard NILM Installations	29
1.2.1	USCGCs Spencer and Escanaba	31
1.2.2	USS Indianapolis	33
1.2.3	USCGC Marlin	33
1.2.4	USCGC Thunder Bay	33
1.2.5	Equipment Condition Monitoring	34
1.3	Contributions and Organization	41
2	Power System Measurement and Preprocessing	43
2.1	Data Preprocessing	44
2.2	Resolution Analysis	46
2.2.1	Noise-free Preprocessor Performance	48
2.2.2	Practical Preprocessor Performance	52
2.2.2.1	Prequantization Noise	52
2.2.2.2	Transfer Characteristic Linearization	54
2.2.2.3	Effective Number of Bits	56
2.2.3	Demonstration with Hardware	60
2.2.4	Design Guide	62
2.2.4.1	Measuring System Noise	63
2.2.4.2	Choosing the Number of Bits	63
2.2.4.3	Choosing the Sampling Rate	65

2.2.4.4	Ability to Resolve Small Loads	66
2.2.4.5	Design Scenarios	67
2.3	Measurement Hardware	68
2.3.1	Frequency Signatures	70
2.3.2	Data Acquisition, Processing, and Filtering	72
2.3.3	Integration with NILM	75
2.4	Beyond Radial Distribution Networks	78
2.4.1	Ring Bus: Shipboard Examples	80
2.4.2	Synchronization	83
2.4.3	Alignment Methods	85
2.4.3.1	Zero-Crossing Delay Tracking	85
2.4.3.2	Voltage Correlation Alignment	87
2.4.3.3	Adjacent Power Event Alignment	89
2.4.3.4	System Power Reconstruction	90
2.4.3.5	Method Choice	93
2.4.4	Experimental Verification	94
2.5	Chapter Summary	101
3	Physics-Informed Feature Extraction and Selection	103
3.1	Multi-scale Framework	103
3.1.1	Event-scale Separation	106
3.1.2	Event Detection and Feature Extraction	106
3.1.2.1	Geometric Methods	107
3.1.2.2	Statistical Methods	111
3.1.2.3	Continuous Methods	115
3.1.3	Event Mapping and Event Confirmation	116
3.2	Transient Space Characterization	118
3.2.1	Voltage Phase Angle	118
3.2.2	Fault Conditions	121
3.2.3	Environmental Factors	123

3.2.4	Shipboard Load Variability	125
3.3	Feature Space Evaluation	128
3.3.1	Feature Space Evaluation	130
3.3.1.1	Separability Check	131
3.3.1.2	Test for Overlap	135
3.3.2	Separability Check Demonstration	136
3.4	Chapter Summary	142
4	Adaptive Classification and Drift Detection	143
4.1	Previously Reported Techniques	145
4.2	Classification and Drift Detection	147
4.2.1	Framework Overview	147
4.2.2	Preliminary Check	150
4.2.3	Coarse Classifier	152
4.2.4	Fine Classifier	154
4.2.5	Drift Clusters	156
4.2.6	Unclassifiable Loads	159
4.3	Experimental Results	160
4.3.1	Two-Dimensional Static Classifier	163
4.3.2	Framework Verification	165
4.4	Physical Interpretation of Results	168
4.4.1	Bilge and Ballast Pump	170
4.4.2	Controllable Pitch Propeller Pump	172
4.4.3	Main Propulsion Diesel Engine Prelube Pump	175
4.4.4	Fuel Oil Purifier	176
4.5	Chapter Summary	179
5	Energy Disaggregation of Stochastic Power Behavior	181
5.1	Statistical Feature Extraction	183
5.2	Energy Estimation	185
5.3	Machining Equipment	190

5.4	Shipboard Loads	195
5.5	Chapter Summary	200
6	Fault Detection and Diagnostics	201
6.1	Ventilation System Diagnostics	202
6.1.1	Fan Physics	203
6.1.1.1	Mechanical Characteristics	203
6.1.1.2	Electrical Tracking	206
6.1.2	Fault Detection Through Power Consumption	207
6.1.3	Fault Detection Using Slot Harmonic Tracking	208
6.1.3.1	Speed Estimation	209
6.1.3.2	Speed Estimation with Filter Blockages	212
6.1.3.3	Speed Estimation With Varying Supply Frequencies	213
6.1.3.4	Multi-Motor Frequency Space Speed Estimation	215
6.1.3.5	Rotor Bar Damage	217
6.1.4	Field Testing on USCGC Marlin	217
6.1.4.1	Primary Supply Fan	218
6.1.4.2	Marlin’s Multi-Motor Environment	221
6.1.4.3	USCGC Marlin At Sea	223
6.2	Three-phase Equipment Condition Monitoring	224
6.2.1	Load Imbalance Feature Space	226
6.2.2	Balanced Three-Phase Loads	229
6.2.2.1	Balanced Non-Linear Loads	230
6.2.2.2	Gradual Drift of Balanced Loads	232
6.2.3	Imbalances in Three-Phase Systems	233
6.2.3.1	Single-Phase Loads	234
6.2.3.2	Three-Phase Heater Faults	236
6.2.3.3	Arcing Fault	244
6.2.3.4	High Impedance Ground Fault	246
6.2.3.5	Unbalanced Nonlinear Loads	253

6.3	Chapter Summary	255
7	Dashboard User Interface	257
7.1	Diagnostic Indicators	257
7.2	Dashboard Overview	262
7.2.1	Timeline View	262
7.2.2	Metrics View	263
7.2.3	Historic View	264
7.3	USCGC Spencer Demonstration	264
7.4	USS Indianapolis Demonstration	266
7.5	Chapter Summary	272
8	Conclusion	273
A	Shipboard Installations	277
A.1	USCGCs Spencer and Escanaba	277
A.1.1	Main Diesel Propulsion Engine System	278
A.1.2	Ship Service Diesel Generator System	280
A.1.3	Fuel Oil Purification System	280
A.1.4	Controllable Pitch Propeller Pump	281
A.1.5	Graywater Pumps	282
A.1.6	In-port Auxiliary Saltwater Pump	282
A.1.7	Bilge and Ballast Pump	282
A.1.8	Auxiliary Room Loads	283
A.2	USS Indianapolis	283
A.2.1	Anchor Windlass VFD	284
A.2.2	Splitter Gear	285
A.2.3	Ship's Service Diesel Generator No. 3 Auxiliary Skid	286
A.2.4	Chill Water Pumps No. 2 and No. 3	286
A.2.5	Hydraulic Power Pack Cooling and Booster Pumps	286
A.3	USCGC Marlin	287

A.3.1	Engine Room Exhaust and Supply Fans	288
A.3.2	Heating Ventilation Air Conditioner	288
A.3.3	Steering Hydraulic Power Unit Cooling Fan	289
A.4	USCGC Thunder Bay	289
B	Jacket Water Heater Fault Analysis	293
B.1	Material Analysis	294
B.2	Natural Frequency Determination	299
B.3	Conclusion	304
C	NILM Software Documentation	305
C.1	Customizing Installable NILM Ubuntu ISO	305
C.2	NILM Computer Setup	310
C.2.1	Installing the ISO	310
C.2.1.1	Acer	310
C.2.1.2	GK41	310
C.2.2	Initial Setup	311
C.2.3	Setting up a Contact Meter	313
C.2.4	Setting up a Noncontact Meter	316
C.3	Data Backup	320
C.3.1	External Drive Setup	320
C.3.2	Manually Mounting /opt/secondary	322
C.3.3	Automatically Mounting /opt/secondary	323
C.3.4	Copying Data to External Drive	324
C.3.5	Change External Drive UUID	324
C.3.6	Check NilmDB Connection	325
C.4	Fix NilmDB on Startup	325
C.5	NILM Dashboard Demo	327
C.5.1	Server Configuration Files	328
C.5.2	Configuring library.yml	329
C.5.3	Running Dashboard Server	330

C.5.4	Example Dashboard Views	331
C.5.4.1	Home Page	331
C.5.4.2	Timeline Page	332
C.5.4.3	Historic Page	332
C.5.4.4	Metrics Page	333
C.6	Useful Joule and NILM Commands	334
C.7	Repairing NilmDB	336
C.8	Known Issues	337
C.9	Installation Files	337
C.10	Demo Event Detector	340
C.11	NerdJack Code Changes	347

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

1. Introduction	25
1-1 Schematic overview of a typical NILM installation.	27
1-2 Ships with NILM installations analyzed in this thesis.	30
1-3 Installed NILM hardware.	31
1-4 MEC engine room systems and monitored loads.	32
1-5 Graywater system.	34
1-6 Graywater pump power observed by a NILM on USCGC Spencer. . .	35
1-7 Scatter plot of graywater pump run duration versus time since last run.	36
1-8 Controllable pitch propeller (CPP) pump.	37
1-9 Power draw of an example run of the CPP pump.	38
1-10 Main propulsion diesel engine jacket water heater.	39
1-11 USCGC Spencer port-side MPDE JW heater turn-on events.	40
2. Power System Measurement and Preprocessing	43
2-1 Conceptual diagram of signal acquisition and preprocessing system. .	46
2-2 Example current stream and real power stream.	47
2-3 12- and 16-bit preprocessor transfer characteristics.	50
2-4 Simulation of preprocessor nonlinearity.	51
2-5 Preprocessor simulation showing correct differential measurements. .	52
2-6 Preprocessor output standard deviation and mean value.	53
2-7 Preprocessor outputs as histograms for 12-bit quantizer.	54
2-8 Histogram of preprocessor outputs.	55

2-9	Simulation of preprocessor linearity from noise for a small load. . . .	56
2-10	ENOB for 16- and 12-bit preprocessors for a sweep of SNR values. . .	61
2-11	12- and 16-bit preprocessor outputs in a shipboard microgrid.	61
2-12	Comparison of four different preprocessors.	64
2-13	Slot harmonics of the test motor at 1657.7 rpm.	72
2-14	Block diagram of the data acquisition, processing, and filtering board.	73
2-15	Data acquisition board.	73
2-16	Ten days of data collection using the described hardware.	77
2-17	Filter performance as a load energizes on USCGC Marlin.	77
2-18	Filter performance and power stream on USCGC Marlin.	78
2-19	Shipboard ring bus example marked with monitoring locations.	81
2-20	Readings from all meters at beginning of data capture and after 24 hours.	84
2-21	Transient distortion and artificial transients in misaligned power streams.	85
2-22	DAQ sample offset drift continuously tracked with voltage zero-crossings.	86
2-23	Voltage frequency and amplitude from USCGC Spencer while underway.	88
2-24	Patterns visible in utility voltage frequency used for time alignment. .	88
2-25	Reconstructed switchboard event stream.	91
2-26	Count of event stream samples outside threshold.	92
2-27	Shipboard ring bus demonstrator.	95
2-28	Generator streams and calculated switchboard power streams.	96
2-29	Calculated power of Switchboard 1 compared to measured.	97
2-30	Calculated power of Switchboards 2 and 3 compared to measured. . .	98
2-31	Generator streams and calculated switchboard power streams.	99
2-32	Power streams when Gen1 and Gen3 are brought offline.	100
2-33	Power streams when generators are put into split plant configuration.	100
3.	Physics-Informed Feature Extraction and Selection	103
3-1	Multi-scale framework.	105
3-2	Diagram for separating short-scale versus long-scale loads.	106
3-3	Geometric, statistical, and continuous events.	108

3-4	Features for an example turn-on transient.	109
3-5	Correlation metric for transient matching.	110
3-6	Power stream of a CNC router.	112
3-7	Real power streams and residual streams.	113
3-8	On-transients of MPDE system loads.	117
3-9	Inrush current transients at various voltage turn-on angles.	119
3-10	Inrush power peak at various voltage turn-on angles.	120
3-11	Variable speed fan with varying off-time between load activations. . .	121
3-12	Experimental setup for variable speed fan in ventilation system. . . .	122
3-13	Variable speed fan with varying filter blockage levels.	122
3-14	Axial fan at various blockage levels.	124
3-15	Axial fan steady state and peak inrush real power.	125
3-16	Normalized histograms of P_{ss}	126
3-17	Q_{ss} versus P_{ss} for the FOP centrifuge.	127
3-18	CPP pump characteristics over time.	128
3-19	Hyperellipsoids in the Q_{ss} versus P_{ss} feature space.	134
3-20	Hyperellipsoids in the Q_{ss} versus P_{ss} feature space.	137
3-21	Hyperellipsoids in the Q_{ss} versus P_{ss} feature space.	138
3-22	Q_{ss} versus P_{ss} boundaries for SVM and DNN classifiers.	139
3-23	Normalized confusion matrices for linear SVM and DNN classifiers. .	141
3-24	Hyperellipsoids after two and ten cruises.	141
4.	Adaptive Classification and Drift Detection	143
4-1	CPP pump preliminary and coarse boundaries and drift clusters. . . .	150
4-2	Graywater pump preliminary and coarse boundaries and drift clusters.	151
4-3	Coarse boundaries drawn in the P_{ss} , S_{peak} and Q_{ss} feature space. . . .	153
4-4	CPP pump drift clusters and representative exemplars.	159
4-5	Initial time-domain transients for each of the loads in the dataset. . .	162
4-6	Q_{ss} versus P_{ss} decision boundaries, trained on a limited dataset. . . .	164
4-7	Normalized confusion matrices with one month of training data. . . .	169

4-8	Example turn-on transients demonstrating drifting steady-state values.	170
4-9	Bilge and ballast pump drift clusters and representative exemplars.	171
4-10	Cumulative recall for the bilge and ballast pump.	172
4-11	Cumulative recall for the CPP pump.	173
4-12	CPP pump long-term and short-term distance metrics.	173
4-13	CPP pump cumulative recall for three GRU implementations.	174
4-14	MPDE prelube pump drift clusters and representative exemplars.	175
4-15	Cumulative recall for MPDE prelube pump.	176
4-16	MPDE prelube pump long-term and short-term distance metrics.	177
4-17	Fuel oil purifier centrifuge motor transients.	178
4-18	Cumulative recall and precision for FOP centrifuge motor.	178
5.	Energy Disaggregation of Stochastic Power Behavior	181
5-1	Example calculation of ECDF curves of CPP pump.	184
5-2	Example calculation of ECDF curves of two loads.	185
5-3	Power stream of bilge and ballast pump.	186
5-4	Energy estimation.	189
5-5	Power stream of industrial bench grinder.	191
5-6	ECDFs of the normalized CNC and grinder streams.	192
5-7	Machine shop energy estimation.	194
5-8	ECDFs of the normalized CPP and BP streams.	196
5-9	Normalized CPP and BP streams.	197
5-10	Shipboard load energy estimation.	198
5-11	Shipboard load energy estimation zoomed-in view.	199
6.	Fault Detection and Diagnostics	201
6-1	Characteristic fan power curves for various fans.	204
6-2	Axial fan power curve.	208
6-3	Centrifugal fan power curve.	208
6-4	Frequency space with three induction machines.	216

6-5	Slot harmonic location shift due to rotor bar damage.	218
6-6	Supply frequency at sea and in-port.	219
6-7	Supply fan on USCGC Marlin.	219
6-8	Measured power and estimated speed for the supply fan.	220
6-9	Marlin multi-motor environment.	221
6-10	FFT current spectrum for the motors for a 10 second window.	222
6-11	Current spectrum and SSEs for a 1 second window.	223
6-12	USCGC Marlin estimated speed with shifting supply frequency.	224
6-13	Conceptual diagrams load connections in a delta-configured system.	227
6-14	Line currents in a delta-configured system.	228
6-15	Conceptual diagram of steady-state calculation.	229
6-16	Three-phase uncontrolled diode bridge rectifier.	230
6-17	Line currents of a simulated 1 kW load controlled by a VFD.	231
6-18	Harmonics of a motor controlled by a VFD.	231
6-19	Three-phase fuel oil purifier (FOP) centrifuge over time.	232
6-20	Relative magnitude of SSDG LO heater over time.	234
6-21	Power of a single-phase inductive load.	235
6-22	Two degraded three-phase heaters.	237
6-23	Load connected across two delta phases.	238
6-24	Delta-connected heater under various fault conditions.	239
6-25	Real and reactive power for USCGC Thunder Bay MPDE JW heater.	240
6-26	Real and reactive power for USCGC Spencer MPDE JW heater.	241
6-27	Two delta-connected heaters operating in tandem.	242
6-28	Current and voltages during the first arcing event.	245
6-29	Currents during the second arcing event.	246
6-30	Zoomed-in view of the second arcing event.	247
6-31	Ungrounded power system with a phase- <i>a</i> fault to ground.	248
6-32	Ground fault power on the faulted phase for different fault resistances.	250
6-33	Measured phase <i>c</i> ground fault.	251
6-34	Measured phase <i>b</i> ground fault	252

6-35	Unbalanced line currents for a motor controlled by a VFD.	254
6-36	Power envelopes of a motor controlled by a VFD.	254
6-37	Unbalanced harmonics of a motor controlled by a VFD.	254
7.	Dashboard User Interface	257
7-1	PDF of a normal distribution showing progressive thresholds.	260
7-2	Timeline example.	263
7-3	Metrics for various conditions of graywater pump operation.	265
7-4	Dashboard timeline view of equipment for eight days of operation. . .	267
7-5	Historical view of the splitter gear LO pump.	268
7-6	Timeline view of in-port operation of the SSDG auxiliary skid.	269
7-7	Timeline view of underway operation of the SSDG auxiliary skid. . .	270
7-8	Daily duty cycle of the SSDG lube oil pump and jacket water pump.	271
A.	Shipboard Installations	277
A-1	USCGC Spencer installations.	278
A-2	Partial schematic of radial distribution onboard USCGC Spencer. . .	279
A-3	NILM installation locations on USCGC Spencer.	279
A-4	CPP system conceptual diagram.	281
A-5	NILM installs on USS Indianapolis (LCS 17).	284
A-6	USS Indianapolis propulsion schematic.	285
A-7	NILM installations on USCGCs Marlin and Thunder Bay	287
A-8	USCGC Marlin loads.	288
A-9	Conceptual diagram of USCGC Thunder Bay propulsion system. . . .	290
B.	Jacket Water Heater Fault Analysis	293
B-1	MPDE JW heater materials analysis.	295
B-2	ESCANABA JW chemistry from March 2017 to May 2021.	296
B-3	Finite element stress simulation and observed damage.	297
B-4	SEM/EDS of the interior surface of the sheathing.	298

B-5	Natural frequency modelling using finite element analysis.	300
B-6	DFTs to determine JW heater natural frequency.	301
B-7	DFTs to determine corroded JW heater natural frequency.	302
B-8	Configuration for monitoring vibration of installed JW heater.	303
B-9	DFT of monitored vibration of JW heater with MPDE idling.	304

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

2. Power System Measurement and Preprocessing	43
2.1 Line cycle sample offsets.	95
4. Adaptive Classification and Drift Detection	143
4.1 Loads from USCGC Spencer in dataset.	161
4.2 Accuracy of classifying on-events (one-shot).	167
4.3 Accuracy of classifying on-events (one month initial dataset).	168
5. Energy Disaggregation of Stochastic Power Behavior	181
5.1 Machining equipment statistical event classification.	193
5.2 Machining equipment energy estimation.	193
5.3 Shipboard loads statistical event classification.	196
5.4 Shipboard loads energy estimation.	199
6. Fault Detection and Diagnostics	201
6.1 Fault characteristics.	205
6.2 Centrifugal fan speed at 60 Hz.	213
6.3 Centrifugal fan speed with variable supply frequency.	214
6.4 Marlin supply fan speed estimates at various blockage levels	218
6.5 Phase currents based on relative line current magnitudes.	233
6.6 Line-to-line single-phase load relationships	236
6.7 Line-to-line open-circuit fault relationships	237

6.8	Delta-connected heating load fault conditions.	239
6.9	Two delta-connected heaters operating in tandem.	242
6.10	Relative line current magnitudes of MPDE JW heater system.	243
6.11	Relative line current magnitudes of unbalanced VFD	253
7.	Dashboard User Interface	257
7.1	Load duty cycles.	267
B.	Jacket Water Heater Fault Analysis	293
B.1	Induced stress magnitudes - CAD model	298
B.2	New heater natural frequency	302

Chapter 1

Introduction

Climate change mitigation and emissions reduction rely on three central aspects: decarbonizing power generation, reducing aggregate energy demand, and electrification of sectors currently dominated by the direct use of fossil fuels [1,2]. We must fundamentally redefine how we generate, control, and consume energy. At the same time, we must ensure the reliability and stability of the electric grid during this transformation. A key component in all aspects of this transformation is the monitoring and efficient operation of electromechanical systems. Electromechanical systems provide the world’s backbone for generating and using energy. They harness the wind to make electricity. They provide heating, cooling, and ventilation to buildings. They manufacture everything from vaccines to vans. Electromechanical systems can also experience an innumerable set of failures, causing induced wear and wasted energy, or eventually a complete failure of a critical piece of equipment or system.

Many electromechanical systems operate under some form of closed-loop or feedback control, which work to maintain an environmental or operating set point, such as temperature, pressure, or tank level [3]. These system operating points are maintained even as system health experiences a gradually evolving “soft fault” e.g., loss of refrigerant, slipping belts, clogged ventilation systems, and so on. As the performance of a critical component in a system degrades, an automatic controller will compensate to maintain commanded output levels by altering run times and net energy consumption. By design, feedback control works to mask the effect of soft faults. Soft faults

degrade operation and can persist for an expensively long time period without causing a full system failure; however, left undetected, a soft fault can eventually turn into a “hard fault,” or the complete failure of a piece of equipment or system. Soft faults are often associated with subtle but observable changes in electrical energy consumption [4]. If collated in a useful way, electrical data can make diagnostic information more easily attainable and improve the efficient operation of critical machines. It can enable operators and consumers to make more informed decisions for system maintenance and flexible energy management.

1.1 Nonintrusive Load Monitoring Background

A nonintrusive load monitor (NILM) is a convenient and economical tool for electrical monitoring since all loads connected downstream of an electrical subpanel are monitored with a single set of current and voltage sensors [5]. Transient behavior in the power stream, such as when a load energizes or changes state, is identified in order to keep a record of load activity. The process of accelerating a rotor, for example, is distinct from the process of booting a computer, and these differences appear as useful patterns or signatures in the power data. There exists an abundance of physical features that reveal information about load behavior and are valuable for load identification. These features can include, for example, the fundamental real power and reactive power changes in steady-state and inrush characteristics, such as peak, duration, and shape [5, 6]. The physical task of the load may also create higher-order harmonics. Power electronics create higher-order harmonics at multiples of the line frequency [7]. Induction motors can produce current harmonics related to the slip frequency that are not multiples of the line frequency [8].

A schematic overview of a typical NILM installation used in this research is shown in Fig. 1-1. The NILM meter contains the analog circuitry and the data acquisition (DAQ) hardware. An analog-to-digital converter (ADC) samples the current and voltage signals at 8 kHz. The NILM software runs on a Linux-based personal computer or laptop, where the data is stored in NilmDB [5]. The current and voltage

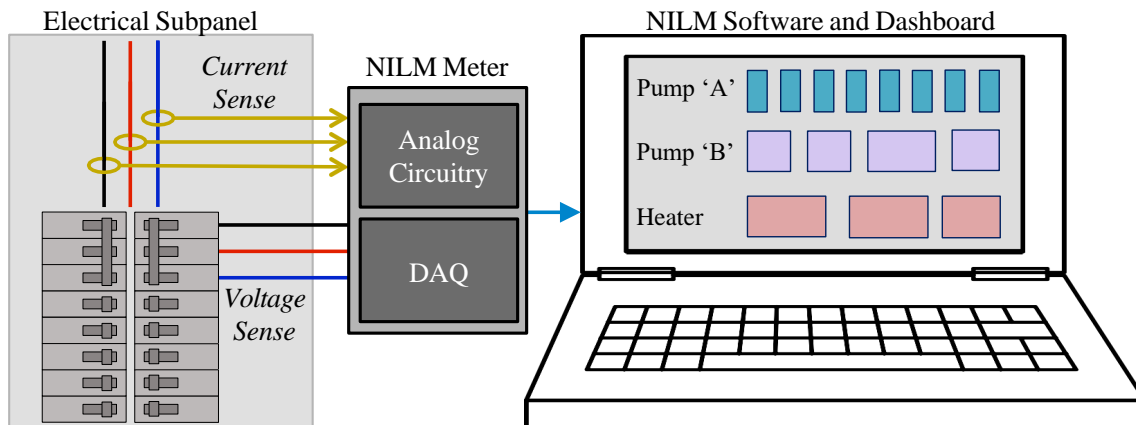


Figure 1-1: Schematic overview of a typical NILM installation.

data is preprocessed into real power, reactive power, and higher-order harmonics at an output frequency equal to the line frequency (i.e., 60 Hz) [9]. This processing compresses the high-rate raw current and voltage data while maintaining the richness of the original signal. After load identification, actionable information such as a timeline of individual load operation and diagnostic metrics is presented to the end-user with the NILM Dashboard [10]. The NILM Dashboard allows users to quickly select and view indicators that draw awareness to unusual or unacceptable operating parameters, including too frequent or infrequent operation, excessive power demand, excessive or inadequate duration of operation, and so forth.

Most nonintrusive load monitoring research is focused on incrementally improving various accuracy metrics for one or several residential (i.e., home electrical appliances) datasets, often a percentage of load operation or power consumption identified in an aggregate power stream [11–18]. However, nonintrusive load monitoring is fundamentally *ad hoc* and conjectural. The application matters [19]. Different approaches for nonintrusive disaggregation allow a trade-off between computational complexity in monitoring, accuracy in determining necessary information for a given application, and flexibility in dealing with changing load compositions. For example, energy scorekeeping in residential homes differs substantially from system diagnostics in an industrial facility. This thesis focuses on applying nonintrusive load monitoring for energy scorekeeping, condition-based maintenance and diagnostics of mission-critical

equipment onboard marine vessels and other industrial facilities. Ships present an application which requires reliability and minimized down-time of systems especially while the vessel is at sea.

The challenge in load identification and diagnostics is ensuring correct results even amidst changing operating conditions and fault scenarios. Most nonintrusive load monitoring research assumes that training data is forever representative of new data. However, real data is not static, referred to in many domains as concept drift [20–22]. That is, load transient events are prone to drift over time. A load’s power characteristics extracted at load transient events (e.g., steady-state and inrush transient peak) can vary; some variations are normal, while others signify impending faults. Many of the features useful for identification are also prone to variation and drift. For a healthy load, variability can arise from the power system itself, such as variations in turn-on angle and voltage amplitude. Mechanical loads have reasonably varying features during different operating conditions, such as cold-starting versus warm-starting a motor. Loads also experience feature variability over time due to changing environmental conditions, machinery aging and wear, or an underlying fault condition.

Evolving electrical characteristics can complicate load identification. However, if drifting features due to degradation can be tracked, it can serve as a useful diagnostic indicator. For many machine learning classifiers, the boundaries in a feature space are not readily explainable or based in physical understanding. Thus, machine learning applications typically rely on large, generalizable data sets. Open-access datasets for nonintrusive load monitoring are generally restricted to healthy residential appliances [23,24]. These datasets cannot generalize to the vast amount of loads and associated fault conditions in industrial and commercial sites. As a result, the training data for a practical nonintrusive load monitoring classifier needs to be collected from the specific system of interest. In active industrial facilities, different operating conditions and fault scenarios cannot be purposefully introduced. Thus, a practical pattern classifier will need to train with limited training data which is likely not representative of long-term operation. This is further complicated since ground truth labels are a scarce resource in industrial nonintrusive load monitoring scenarios. To nonintrusively track

the evolution of a soft fault from the aggregate power stream, this thesis uses an understanding of the physics and time-dependent nature of load behavior to inform feature space selection improvements and load identification.

Some useful information for condition monitoring and energy disaggregation occurs during load operation and cannot be extracted from the on and off transients. That is, not all loads have a constant power demand during their entire operation. Even if the turn-on transient of a load can be identified, it may not tell the full story of the load’s operation. Information remains “hidden” with conventional nonintrusive load monitoring techniques. For instance, energy estimation of individual loads typically relies on relatively well-defined transient signatures and discrete steady-state levels. Many loads exhibit non-constant power demand in response to dynamic loading conditions; for example, machines like routers and mills have power demand that depends on cutting conditions. This thesis introduces new statistical techniques for these loads. Soft faults can also result in slow changes in power during load operation. That is, for loads that operate for days or even weeks at a time, it is not feasible to rely on the load on and off events to track the degradation. In these cases, other features, such as induction machine slot harmonics, can be used in tandem with power stream data to track motor or system health.

1.2 Shipboard NILM Installations

To demonstrate the ideas presented in this thesis, power monitoring data has been collected from various marine microgrids. A typical installation monitors all or part of the main engineering space. Field installations of NILM systems have been conducted on several United States Coast Guard (USCG) and Navy vessels, including USCG Cutters (USCGCs) Spencer, Escanaba, Marlin, and Thunder Bay, and United States Ship (USS) Indianapolis. These ships are shown in Fig. 1-2. An example of the NILM hardware installation on the ships is shown in Fig. 1-3. Fig. 1-3a shows the NILM all-in-one (AIO) box mounted next to the monitored electrical subpanel on USCGC Thunder Bay. The AIO box contains the NILM meter, CPU, and touch-



(a) USCGC Spencer.



(b) USCGC Escanaba.



(c) USS Indianapolis.



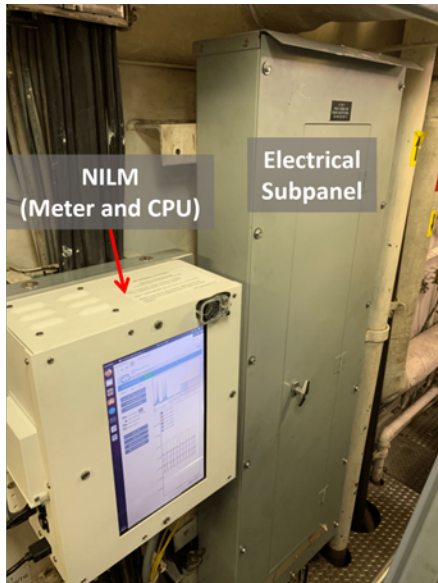
(d) USCGC Marlin.



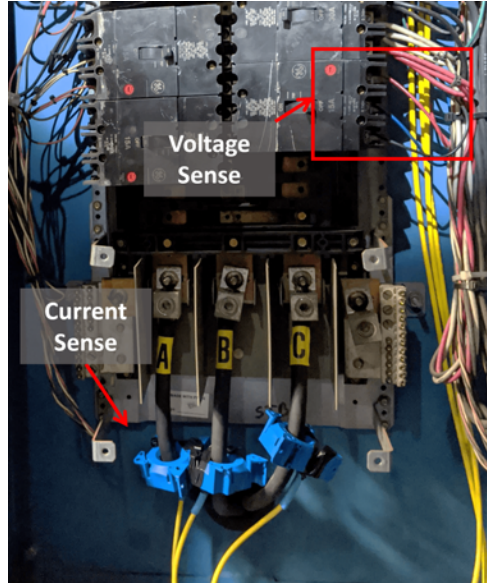
(e) USCGC Thunder Bay.

Figure 1-2: Ships with NILM installations analyzed in this thesis.

screen interface. Fig. 1-3b shows the electrical subpanel interior of USCGC Marlin with installed hardware for current and voltage sensing. Fig. 1-3c shows the NILM Dashboard on USCGC Spencer, which was installed overlooking the engine room. Data from the installations on these five vessels are analyzed in this thesis. This section introduces the shipboard NILM installations. The presented examples illustrate the value and challenges of nonintrusive load monitoring in industrial settings. For



(a) NILM AIO box installation on USCGC Thunder Bay.



(b) Panel interior of USCGC Marlin showing voltage and current sensing.



(c) NILM Dashboard display onboard USCGC Spencer, overlooking the engine room.

Figure 1-3: Installed NILM hardware.

reference, detailed descriptions of the installations are presented in Appendix A.

1.2.1 USCGCs Spencer and Escanaba

Our longest installations provided observations for over five years from USCGCs Spencer and Escanaba, two 270-ft. medium endurance cutters (MEC), homeported in Boston, Massachusetts. When these vessels are at sea, power is provided by two

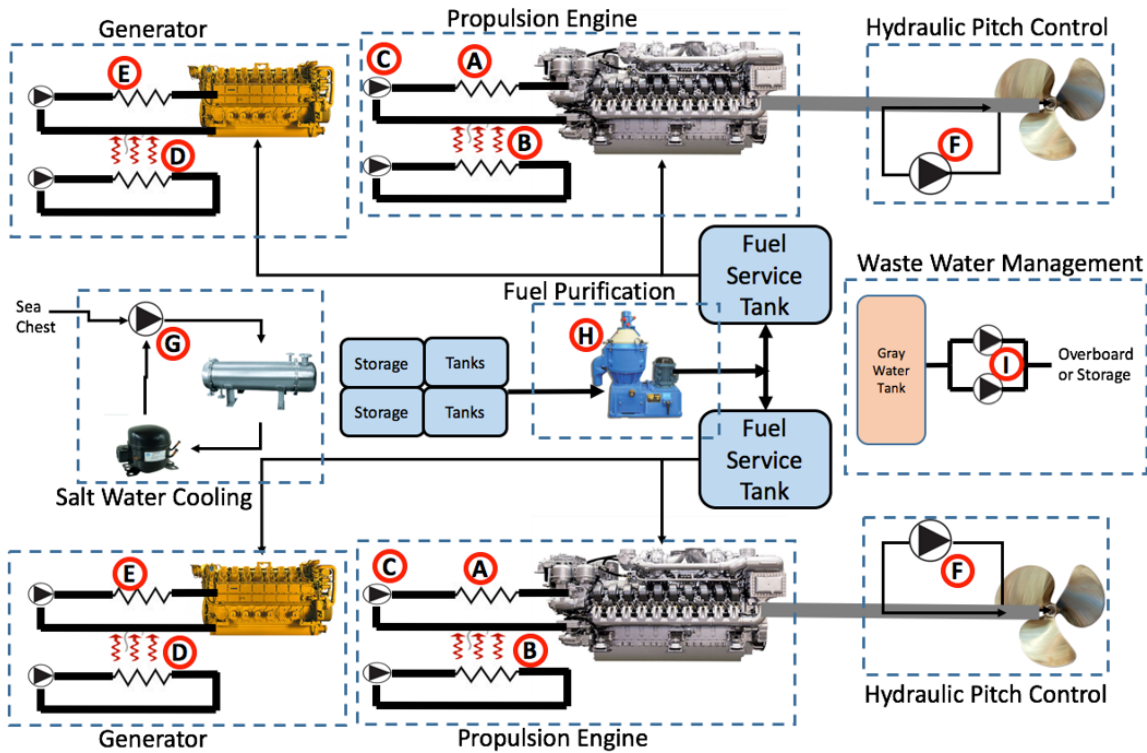


Figure 1-4: MEC engine room systems and monitored loads: A. MPDE lube oil heater, B. MPDE jacket water heater, C. MPDE prelube pump, D. SSDG jacket water heater, E. SSDG lube oil heater, F. controllable pitch propeller pump, G. inport auxiliary saltwater pump, H. fuel oil purifier centrifuge and feed pump, I. graywater pump [25].

ship service diesel generator (SSDG) sets. When in-port, power is provided by either an aft or forward connection to utility shore power. The generator sets, along with the main propulsion diesel engines (MPDEs) propelling the ship, are located in the ship's engine room. The SSDGs and MPDEs require auxiliary equipment, e.g., pumps and heaters, to maintain operational readiness when in standby mode. Two electrical subpanels, port and starboard, which power these loads along with several other engine room loads critical for ship operation, were monitored by NILM systems on both ships. These various NILM systems, installed between 2014 and 2017, consisted of two boxes, one for the NILM CPU and one for the NILM meter hardware. A conceptual diagram of the engine room loads on the monitored subpanels are shown in Fig. 1-4.

An interactive interface for the crew was added in July 2018 on USCGC Spencer,

as the first prototype of the NILM Dashboard graphical user interface (GUI). This allowed the crew to navigate through different loads on the port and starboard panels to plot on the Dashboard to verify operation. Additionally, at the request of the ship's crew, a NILM was installed in January 2019 to monitor the power panel in the main auxiliary room of USCGC Spencer. This panel supplies power to auxiliary equipment critical to ship operation.

1.2.2 USS Indianapolis

In February 2020, three all-in-one (AIO) NILMs were installed to monitor power panels on USS Indianapolis, a Freedom-class littoral combat ship (LCS) in Mayport, Florida. These NILMs were installed on subpanels in the auxiliary machine room, water jet machinery room, and machine shop. When USS Indianapolis is at sea, power is provided by four SSDG sets. For propulsion, the vessel uses a combined diesel and gas turbine plant coupled with four water jets. The vessel uses the diesel engines for cruising speeds and the gas turbines for high-speed transits. The SSDGs and propulsion system require electrical equipment both while operating and during standby.

1.2.3 USCGC Marlin

In August 2020, two AIO NILMs were installed on USCGC Marlin, a 87-ft. patrol boat homeported in Boston, Massachusetts. The two monitored panels, port and starboard, monitor the entire ship, with the exception of two steering pumps. Similar to the MECs, USCGC Marlin has two SSDG sets and two MPDEs. This installation included the first field installation of new data acquisition hardware, as will be described in this thesis.

1.2.4 USCGC Thunder Bay

In February 2021, two AIO NILMs were installed on USCGC Thunder Bay, a 140-ft. icebreaking tug homeported in Rockland, Maine. The two monitored electrical

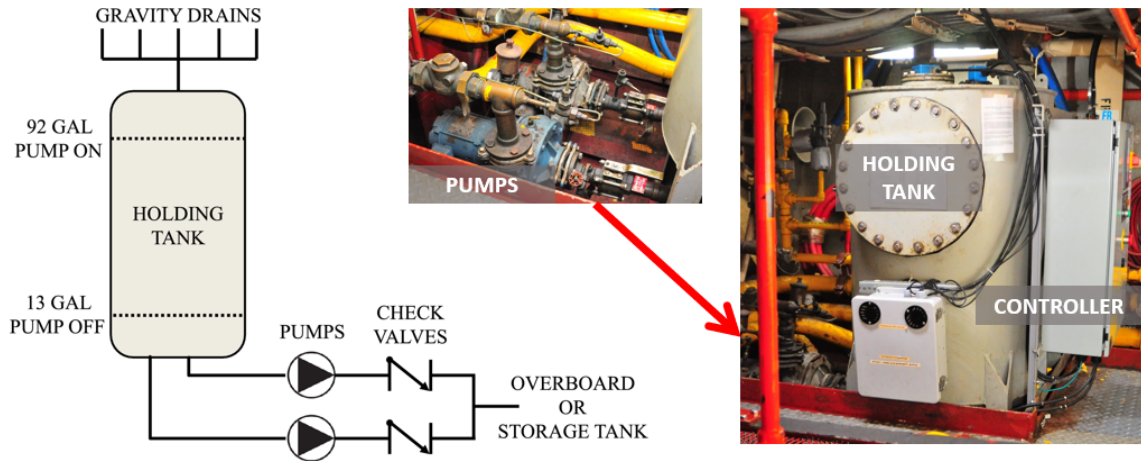


Figure 1-5: Graywater system.

panels include one in the engine control center (ECC) and one in the engine room. USCGC Thunder Bay is one of few Coast Guard assets powered by a diesel electric propulsion plant, with two MPDEs driving two main propulsion generators (MPG), which in turn provide power to the main propulsion motor that drives the ship's single propulsion shaft. The monitored ECC panel provides power to loads primarily related to the electric propulsion machinery, including the field exciters for the propulsion motor and propulsion generators. The monitored engine room panel supplies auxiliary equipment primarily related to the two MPDEs and two SSDGs.

1.2.5 Equipment Condition Monitoring

The nominally rated 3.7 kW graywater pumps on the USCG MECs are presented here to highlight the applicability of electrical monitoring for detecting load cycling faults. The graywater pump is part of the graywater disposal system. The graywater system and conceptual diagram are shown in Fig. 1-5. The system is designed to transfer, retain, process, and dispose of the relatively clean waste water from showers, sinks, and other appliances to a holding tank in the main engine room. When the holding tank is full, the graywater pump discharges the water from the holding tank either overboard or to a larger storage tank. There are two pumps (for redundancy) which alternate each cycle to empty the tank. The graywater pumps operate based on

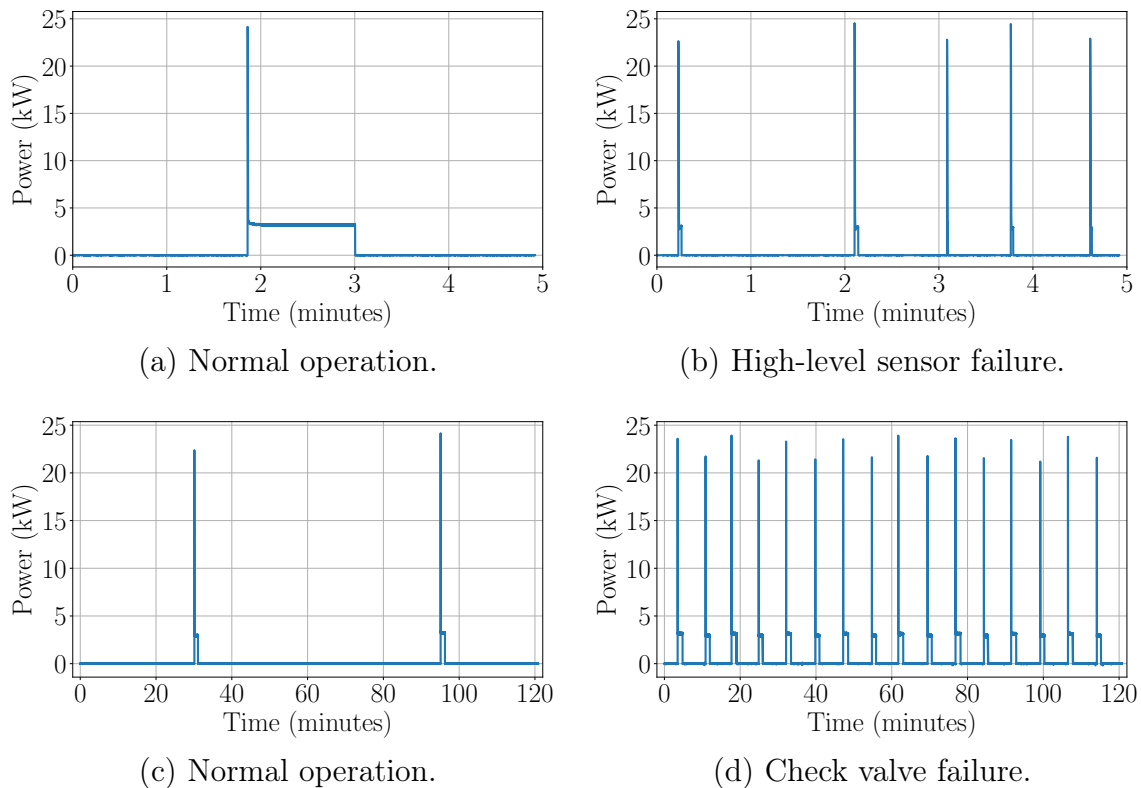


Figure 1-6: Graywater pump power observed by a NILM on USCGC Spencer.

water-level sensors. Normally, a pump turns on and begins discharging when water reaches the “high-level” sensor set point (92-gallon mark), and the pump turns off when water reaches the “low-level” sensor set point (13-gallon mark).

During monitoring, the graywater system on USCGC Spencer experienced two faults (not at the same time). The first fault was a failed high-level sensor and the second was a failed check valve. Both types of faults went unnoticed by the crew for extended periods as the graywater disposal system still performed its job of collecting and disposing graywater. However, both these faults caused significantly more pump runs, causing undue stress and mechanical wear on the system [26]. The high-level sensor failure, due to solid residue shorting the sensor, resulted in constant high-level readings. This caused the pumps to short-cycle. The graywater events were short, typically only a few seconds or less since the tank was essentially empty. The power streams for a five-minute window showing one example healthy pump run and several runs during the high-level sensor failure are shown in Fig. 1-6a and Fig. 1-6b,

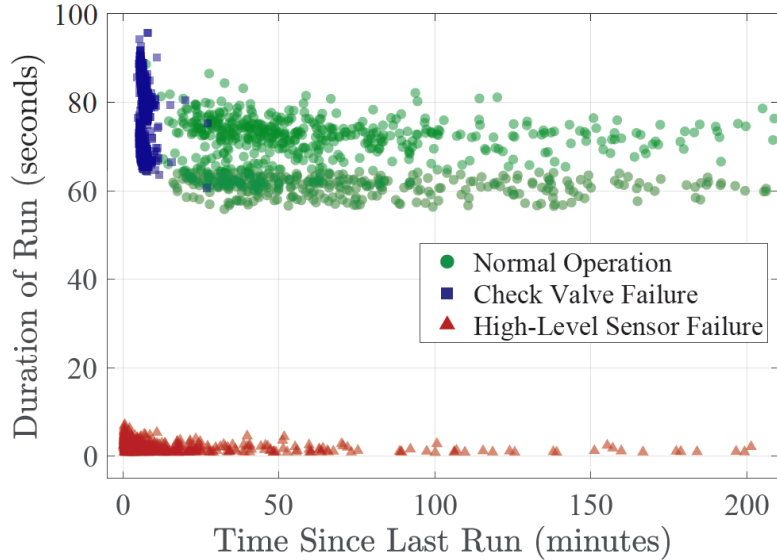


Figure 1-7: Scatter plot of graywater pump run duration versus time since last run.

respectively. The broken check valve fault allowed water to flow backwards from the storage tank to the holding tank. The pumps had to run almost continuously to keep the holding tank from overflowing. There is also a slightly longer pump run duration, because of the backflow. For example, power streams for two-hour windows of normal operation and a system with a broken check valve are shown in Fig. 1-6c and Fig. 1-6d, respectively. In this example, there are approximately 65 minutes between the two runs during normal operation, whereas with a broken check valve, the pump is cycling approximately every 7 to 8 minutes.

A NILM can keep a running scatter plot, for each detected observation of graywater pump operation, that locates an event on a plot of “duration of pump run” versus “time since last pump run.” Healthy pump runs might be expected, generally, to fall in a bubble of “normal operation.” Various faults are likely to create pump run events that consistently fall outside of this region. Fig. 1-7 shows a scatter plot of data from USCGC Spencer for the two described fault scenarios as well as during healthy operation. During the high-level sensor failure the runs were sometimes too short for the load to reach steady-state. As a result, the NILM ignored these transients and the actual “time since last run” for the labelled events were likely even shorter than the recorded values.

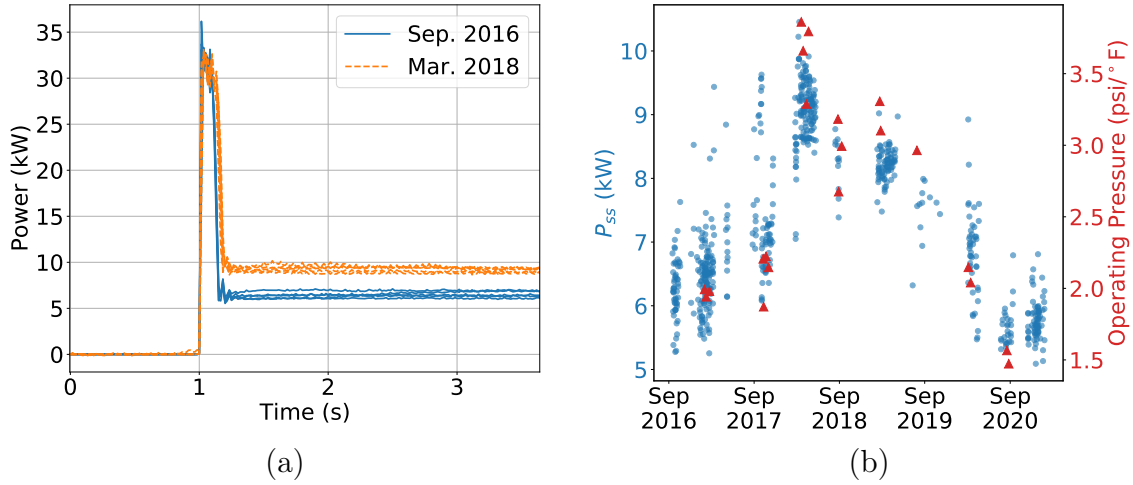


Figure 1-8: Controllable pitch propeller (CPP) pump. (a) Example of five turn-on transients from September 2016 and five turn-on transients from March 2018. (b) Steady-state power and operating pressure over time.

Next, the controllable pitch propeller (CPP) pumps from the MECs are used to highlight two challenges: 1) the drift of load power characteristics over time due to changes in operating condition and 2) non-constant power demand due to dynamic loading conditions. Controllable pitch propellers allow operators to adjust the amount of thrust generated by a propulsor while maintaining a fixed rotational speed [27]. The pitch of the propeller blades is altered via a hydraulic control system that runs from the engine room to the propellers through the center of the propulsion shaft. By changing pitch, the speed and direction of ship movement are controlled. On the MECs, the CPP system consists of three primary hydraulic pumps that provide pressurized oil to the system in order to maintain hydraulic control pressure at the propeller. The ‘A’ pump is a gear-driven pump that is powered by the propulsion system’s reduction gear. The pressure and flow provided by the ‘A’ pump is dependent on propeller shaft speed. The ‘B’ and ‘C’ pumps are electric hydraulic pumps that supplement the pressure and flow provided by the gear driven pump. NILMs on USCGCs Spencer and Escanaba monitor both the port and starboard CPP ‘C’ pumps, which are rated at 7.5 kW.

A NILM detected changes in the monitored CPP ‘C’ pump’s steady-state power consumption, as shown in Fig. 1-8 for the port-side pump on USCGC Spencer. Fig. 1-

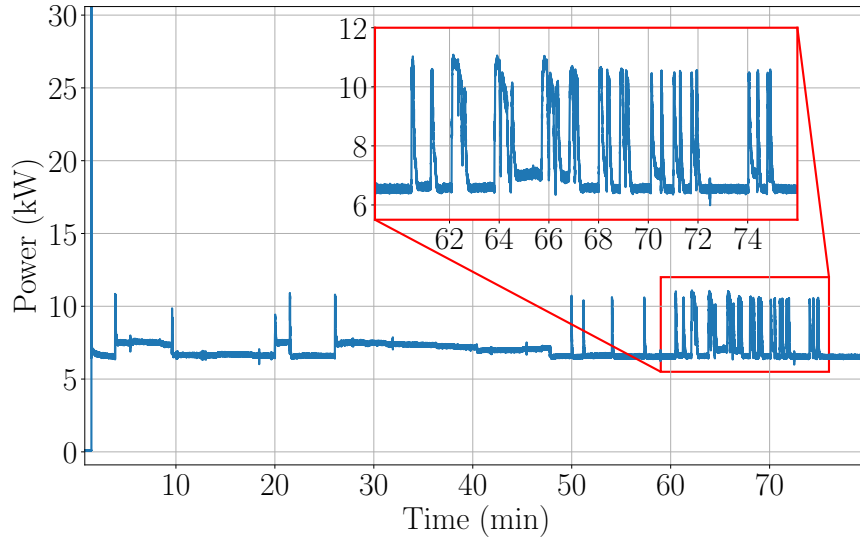
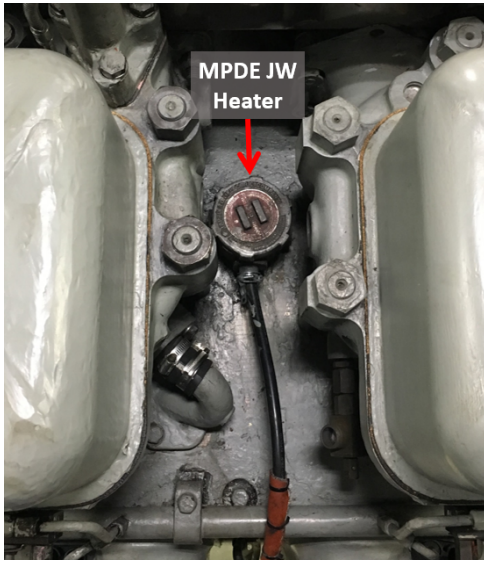


Figure 1-9: Power draw of an example run of the CPP pump with a zoom-in on power “surging.”

8a shows five time-domain on-transients from September 2016 and five on-transients from March 2018, showing significantly different steady-state levels. Fig. 1-8b shows the calculated real steady-state power (P_{ss}) for a four-year period, where each data point represents a single on-event. The steady-state values were calculated as the difference between the median values over 0.5 second length windows before and after each identified event. This change in steady-state power correlates with the operating fluid temperature normalized by temperature (as obtained from the ship logs), as shown on the right axis in Fig. 1-8b. In February 2018, after replacement of the hydraulic control valves, there was a large increase in both the normalized operating pressure and the power draw. Then, as the normalized operating pressure slowly decreased over time, the power draw also decreased. This demonstrates how a load’s power characteristics can drift over time due to operating condition. Furthermore, it is likely a pattern classifier would only be trained on a limited data set collected shortly after installation. That is, much of the long-term operation will not be represented in the training dataset.

The steady-state values in Fig. 1-8b represent the steady-state power shortly after the load energizes. However, the power consumption of the CPP pump can also change during operation due to varying pressure demand. Between an on-event and



(a) Heater mounted between two pistons on the engine block.



(b) New heater and degraded heaters with holes 1 cm and 5 cm from base, respectively.

Figure 1-10: Main propulsion diesel engine jacket water heater.

subsequent off-event, the load does not always consume a constant amount of power. For instance, Fig. 1-9 shows the real power of one example run of a monitored CPP pump. There are “surges” in power, as highlighted by the zoomed-in window. These surges are a result of the electric CPP pump compensating for the extra pressure required during the changes in command of the pitch of the propeller blades as the operator is maneuvering the ship. Detection of these surges in power can provide insights into ship operation, since throttle commands that alter propeller blade pitch require greater system pressure and places a greater demand on the system.

As a final example, the MPDE jacket water (JW) heaters of the MECs are presented here to illustrate changes in steady-state power consumption that are due to load degradation. Both the port and starboard subpanels of USCGCs Spencer and Escanaba supply power to a MPDE JW heater system, consisting of two symmetrical delta-connected 4.5 kW heaters on either side of the engine block. The JW heaters are part of the MPDE keep-warm system, used when the engine is not running to minimize time to reach engine standard operating temperature upon start up. Fig. 1-10a shows a heater mounted between two pistons on the engine block, where the

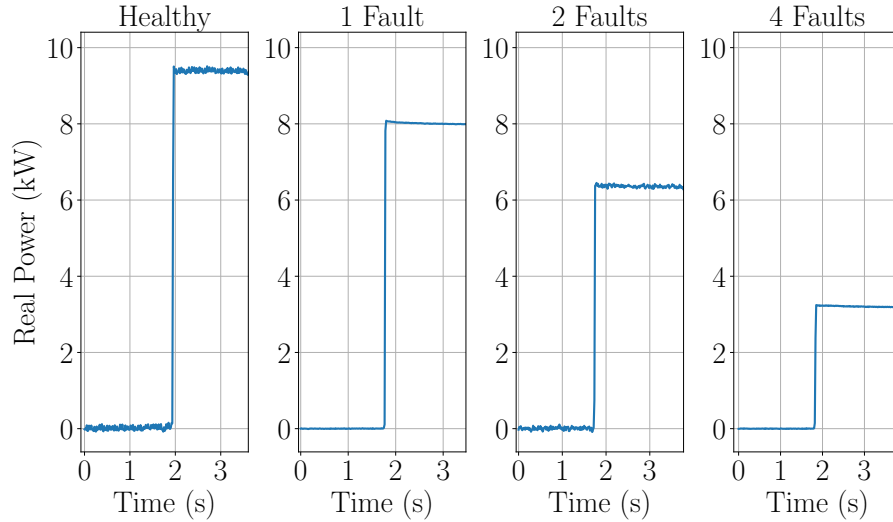


Figure 1-11: USC GC Spencer port-side MPDE JW heater turn-on events, starting healthy and after various stages of degradation.

heater experiences considerable stress due to heat and vibration while the engine is running. Inspection of the various degraded JW heaters revealed significant corrosion and damage to heating elements. Fig. 1-10b shows a comparison of a new heater and two degraded heaters.

The two heaters are activated in tandem, so a NILM sees the heater pair as drawing 9 kW under normal operation. An open-circuit failure can occur on any of the six heating elements. That is, only after the sixth consecutive failed heating element will the JW heating system have a complete system failure. These events are insidious, as the JW heating system can still use its operational heating element or elements to warm the jacket water. The heaters may also operate longer, but this is invisible to the ship's watchstander as it is a fully automated system. As a heating element fails, it leads to changes in the associated power waveform. For instance, Fig. 1-11 shows the real power turn-on transient for four different measured scenarios of the port-side MPDE JW heater system on USC GC Spencer. The leftmost plot is of two healthy heaters. The subsequent plots have one, two, and four failed heating elements, respectively. Early detection of these failures can prevent service delays and electrical safety hazards.

1.3 Contributions and Organization

This thesis makes contributions in six areas for fault detection, diagnostics, and advanced power monitoring. Each of these contributions relate to a main step of non-intrusive load monitoring and will be presented in the thesis as a chapter:

- **Power system measurement and preprocessing:** This chapter examines the advantages and trade-offs of spectral envelope preprocessing for use in power monitoring, fault detection, and diagnostic applications. Then, details are presented of new data acquisition hardware that filters out the fundamental utility component, and enables inspection of subtle higher-order harmonics. Signal processing techniques for the application of nonintrusive load monitoring in non-radial power distribution systems, such as ring distribution systems, are provided.
- **Physics-informed feature extraction and selection:** Next the relevant features for load identification and diagnostics need to be extracted and selected. Various examples of load variability are characterized using laboratory demonstrations and field data. A method is presented for assessing a possible feature space to find a set of electrical characteristics that provide adequate separability between classes.
- **Adaptation for classification and drift detection:** Once a feature space is selected, a classifier needs to be trained. This chapter presents a multi-level framework of classification for problems facing concept drift, limited training data, and extreme verification latency, as is often the case for a practical nonintrusive load monitoring scenario.
- **Energy estimation for stochastic power behavior:** The next step after load identification is energy disaggregation. For loads with approximately discrete steady state levels, this is a reasonably straightforward step. However, energy disaggregation is more difficult for loads that do not have unique power consumption levels or a fixed number of states. This chapter presents an energy estimation method for loads that have dynamic power consumption due to changes in physical loading.

- **Fault detection and diagnostics:** Once individual loads and their power consumption have been identified from the aggregate stream, patterns in load behavior can be analyzed for faulty behavior. This chapter presents methods for detecting fault conditions that manifest as changing power characteristics by incorporating additional physics-based features.
- **Dashboard user interface:** The final step is for the data to be presented as an intuitive decision aid for users. This chapter presents the NILM Dashboard and its easy-to-understand visual displays with demonstrations from various ship installations.

These six areas encompass a complete NILM system from data collection to data presentation. Laboratory experiments and case-studies are presented from NILM installations onboard the described marine microgrids. The applicability of nonintrusive load monitoring is demonstrated for making diagnostic information more easily attainable and improving the efficient operation of critical machines. Hardware, software, and signal processing techniques are presented.

Chapter 2

Power System Measurement and Preprocessing

This chapter makes contributions in several areas of power system measurement and preprocessing. First, background is presented for data preprocessing of power system measurements using spectral envelopes. Then the advantages and limitations of spectral envelope preprocessing for power measurements are examined. Data acquisition hardware is presented that can automatically track and reject carrier (utility) frequency content in a nonintrusive current measurement, permitting the resolution of subtle higher harmonic content with the full range of an available analog-to-digital converter (ADC). This data acquisition hardware was made compatible with the NILM software suite, enabling measurement of three-phase current and voltage data and filtered current data with the fundamental component removed. The hardware and software was demonstrated with a field installation on USCGC Marlin. Finally, this chapter provides signal processing techniques for the application of nonintrusive load monitoring in non-radial power distribution systems, such as ring distribution systems.

2.1 Data Preprocessing

The shipboard field installations presented in this thesis are monitoring three-phase distribution systems. For a three-phase distribution system with harmonic-free voltages (i.e., containing only the fundamental component), the line-to-neutral voltages can be represented as:

$$\begin{aligned} v_a(t) &= V_a \cos(\omega t + \theta_a) \\ v_b(t) &= V_b \cos\left(\omega t - \frac{2\pi}{3} + \theta_b\right) \\ v_c(t) &= V_c \cos\left(\omega t + \frac{2\pi}{3} + \theta_c\right) \end{aligned} \quad (2.1)$$

where V_a , V_b , and V_c are the peak magnitudes and θ_a , θ_b and θ_c are the phase angles. For a balanced system, $V_a = V_b = V_c = V_{pk}$ and $\theta_a = \theta_b = \theta_c = 0$. For a single frequency component, the voltages can be represented in phasor notation. The line-to-neutral and line-to-line voltages are:

$$\begin{aligned} v_a &= V & v_{ab} &= \sqrt{3}V e^{j\frac{\pi}{6}} \\ v_b &= V e^{-j\frac{2\pi}{3}} & v_{bc} &= \sqrt{3}V e^{-j\frac{\pi}{2}} \\ v_c &= V e^{j\frac{2\pi}{3}} & v_{ca} &= \sqrt{3}V e^{j\frac{5\pi}{6}} \end{aligned} \quad (2.2)$$

where V is the rms line-to-neutral voltage (i.e., $V = V_{pk}/\sqrt{2}$). The line currents at line-frequency harmonic k can be represented as:

$$\begin{aligned} i_{ak}(t) &= A_{ak} \cos(k\omega t + \delta_{ak}) \\ i_{bk}(t) &= A_{bk} \cos\left(k\left(\omega t - \frac{2\pi}{3}\right) + \delta_{bk}\right) \\ i_{ck}(t) &= A_{ck} \cos\left(k\left(\omega t + \frac{2\pi}{3}\right) + \delta_{ck}\right) \end{aligned} \quad (2.3)$$

where A_{ak} , A_{bk} , and A_{ck} are the peak magnitudes, δ_{ak} , δ_{bk} and δ_{ck} are the phase angles, and ω is the angular frequency.

The discrete current signal for each phase is $i_\psi[n]$, where $\psi \in \{a, b, c\}$. Given a sampling frequency f_s and line frequency f_0 , one period is length $N = f_s/f_0$ samples. To detect transient behavior, the currents are first converted to spectral envelopes,

i.e., a measure of harmonic content averaged over every ac line cycle [9,28]. Spectral envelopes are good representations of signals assuming they are locally periodic over one ac line cycle. Spectral envelopes of the currents are computed over sliding windows of the input signal phase-aligned with a reference voltage, $v[n] = V_{pk} \sin(2\pi n/N)$. By assuming the voltages are relatively “stiff” and harmonic-free, a single measured voltage is used as the phase reference. For a general current $i[n]$, the in-phase and quadrature components of the rms current envelope at line frequency harmonic k , assuming the current is phase-aligned with the reference voltage, is:

$$I_k^I = \frac{\sqrt{2}}{N} \sum_{n=0}^{N-1} i[n] \sin\left(k \frac{2\pi n}{N}\right) \quad (2.4)$$

$$I_k^Q = -\frac{\sqrt{2}}{N} \sum_{n=0}^{N-1} i[n] \cos\left(k \frac{2\pi n}{N}\right). \quad (2.5)$$

These are equivalent to the imaginary and real components, respectively, of the discrete Fourier transform (DFT) of $i[n]$, scaled by $-\sqrt{2}/N$. The DFT transforms some signal $x[n]$ into X_k :

$$\begin{aligned} X_k &= \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-jk2\pi n/N} \\ &= \sum_{n=0}^{N-1} x[n] \cdot \left(\cos\left(k \frac{2\pi n}{N}\right) - j \cdot \sin\left(k \frac{2\pi n}{N}\right) \right). \end{aligned} \quad (2.6)$$

For three-phase systems, the measured voltage used as the phase reference can either be the line-to-neutral voltage or line-to-line voltage, depending on availability and access of measurements, as long as the correct phase rotation corrections are applied. A correcting rotation of $k\phi_0$ can be applied to the complex DFT coefficient X_k as $X_k \cdot e^{\phi_0 \cdot jk}$. When the voltages are measured line-to-neutral, phase rotations of 0° , 120° , and 240° should be applied when computing spectral envelopes corresponding to $i_a[n]$, $i_b[n]$, and $i_c[n]$, respectively. When the voltages are measured line-to-line, phase rotations of 30° , 150° , and 270° should be applied. In the line-to-line case, it is also possible for there to be a phase shift of 180° of the voltages with respect to

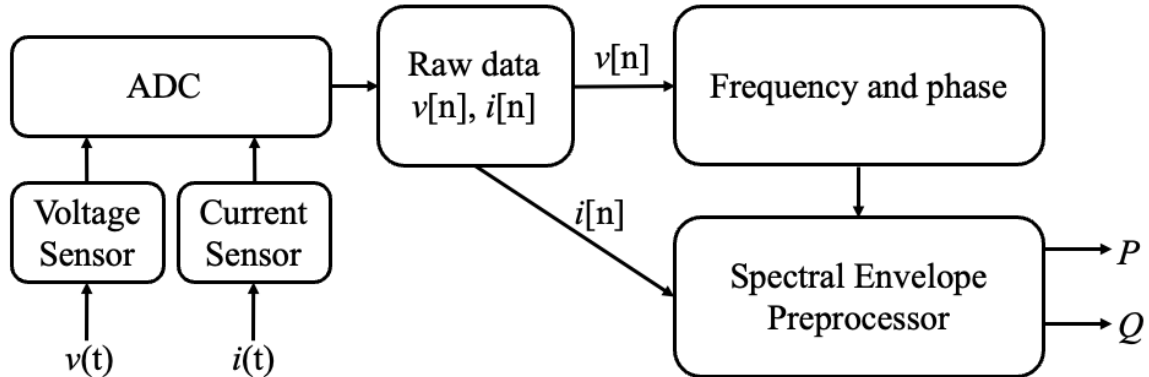


Figure 2-1: Conceptual diagram of signal acquisition and preprocessing system.

the currents. In which case, phase rotations of 210° , 330° , and 90° should be applied instead. The in-phase and quadrature spectral envelopes for each phase $\psi \in \{a, b, c\}$ are represented as $I_{\psi k}^I$ and $I_{\psi k}^Q$, respectively. The fundamental current component is used to calculate the real power (P) and reactive power (Q) streams for each phase:

$$P_\psi = VI_{\psi 1}^I, \quad Q_\psi = VI_{\psi 1}^Q. \quad (2.7)$$

A conceptual diagram of the signal acquisition and preprocessing system is shown in Fig. 2-1. Fig. 2-2 shows an example current stream and computed real power using Eq. (2.7) with rms voltage $V = 120\text{V}$.

2.2 Resolution Analysis

This section provides in-depth analysis of the benefits and limitations of spectral envelope preprocessing for use in power monitoring. The work in this section was in collaboration with Aaron Langham and is presented in [29].

A wealth of information about the operation of electromechanical devices lies hidden in measurable signals. Many of these signals can be characterized as being locally periodic. That is, at some resolution or level of accuracy, a signal exhibits periodicity, but contains nonperiodic characteristics when observed on a longer time scale. Averaging techniques can increase the accuracy and fidelity of acquisition of these types of signals. Oversampling, both with direct quantization and with noise shaping, can im-

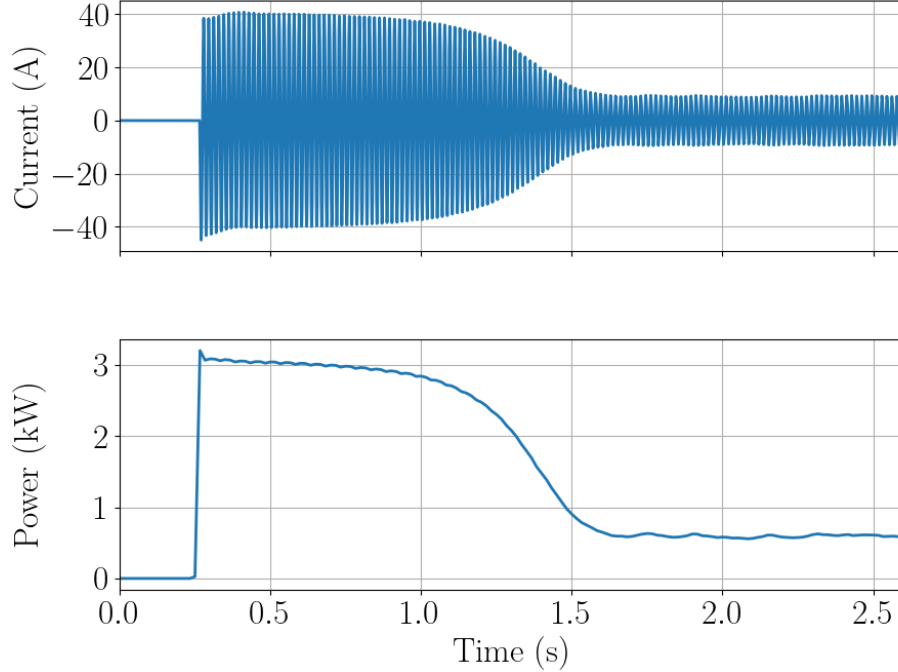


Figure 2-2: Example current stream and real power stream.

prove resolution at the expense of sample rate, in a form of averaging [30]. Ensemble averaging can make use of analog or “prequantization” noise to reduce quantization error and improve resolution across ensembles of periods [31]. An averaging technique that is particularly useful for the monitoring of electromechanical equipment is spectral envelope preprocessing, as was briefly described in Section 2.1, which allows for compression of signals sampled at a high sampling rate while retaining salient information about the signal’s harmonic content [9]. Spectral envelopes are particularly useful for power monitoring, where extraction of in-phase and quadrature components is desirable to approximate real and reactive power, respectively.

The analysis presented in this section uses the fundamental ($k = 1$) in-phase component (P) as the preprocessor output without loss of generality due to the orthogonality of sine and cosine. P is given by:

$$P = \frac{V_{pk}}{N} \sum_{n=0}^{N-1} i[n] \cdot \sin(2\pi n/N). \quad (2.8)$$

For loads with a non-zero phase angle or with harmonic content, similar analysis can

be done for Q and higher-order harmonics that will yield the same insights. Since spectral envelope preprocessing averages or sums over many time points, the number of possible power values is larger than the number of current sample levels that are resolvable with an input quantizer of B bits [9]. However, similar to the quantization artifacts that a quantizer introduces, this averaging is associated with its own set of artifacts and distortions present in the resulting downsampled output signal. The extra preprocessor output values are not evenly spaced across the range of input amplitudes. Thus, the preprocessor’s ability to resolve loads (i.e., to both detect and accurately measure step changes), varies based on the input signal. When using a preprocessor output as an input to a feature extraction system for training a classifier, as in the case of nonintrusive load monitoring, it is highly undesirable to have training data that contains biases or inaccuracies, even if, or perhaps especially if, they only predominantly affect certain ranges of input signal values.

This section examines advantages of spectral envelope preprocessing for power measurements [29]. Design techniques are presented for optimizing the resolution and linearity of spectral envelope data for power monitoring, fault detection, and diagnostic applications. These design techniques are illustrated by contrasting the performance of two different hardware preprocessors, the one that will be described in Section 2.3 and the LabJack UE9. The effects of noise on the power system and introduced by data acquisition hardware are included. A new measure of the effective number of bits or power resolution is developed for the output of a preprocessing system in the presence of prequantization noise, which is illustrated with data collected on USCGC Marlin. A design guide is developed for the construction or modification of a data acquisition system using a spectral envelope preprocessor, with relevant trade-offs and considerations for parameter choices.

2.2.1 Noise-free Preprocessor Performance

Spectral envelope preprocessing increases output resolution by averaging over N sampled data points over one or more periods of a quasi-periodic input signal [9]. We begin by considering a noise-free system with a stiff, sinusoidal voltage. A sinusoidal,

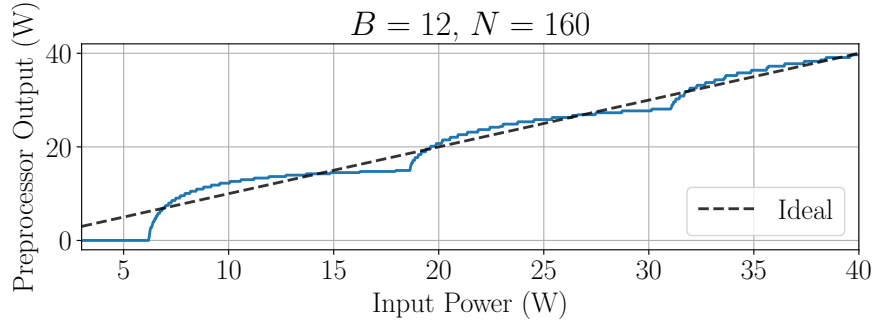
in-phase current with amplitude A produces a fundamental frequency spectral envelope associated with real power proportional to A . Quantized sampling results in a discrete number of possible values, 2^B , for each current sample $i[n]$, where B is the number of input quantizer bits. Therefore, over an observation window, there is a finite number of unique sampled waveforms of the current signal, denoted as U [9]. In calculating spectral envelopes, the DFT, shown in Eq. (2.6), maps each unique input current signal to a corresponding sequence of unique frequency-domain outputs, creating the same number U of unique preprocessor outputs. Effectively, the preprocessor can discern U different waveforms corresponding to different values of A . As derived in [9], the total number of unique preprocessor outputs is:

$$U = 2^{B-1} + \sum_{n=1}^{\frac{N}{4}-1} \left[(2^{B-1} - 1) \cdot \sin\left(\frac{2\pi n}{N}\right) + \frac{1}{2} \right]. \quad (2.9)$$

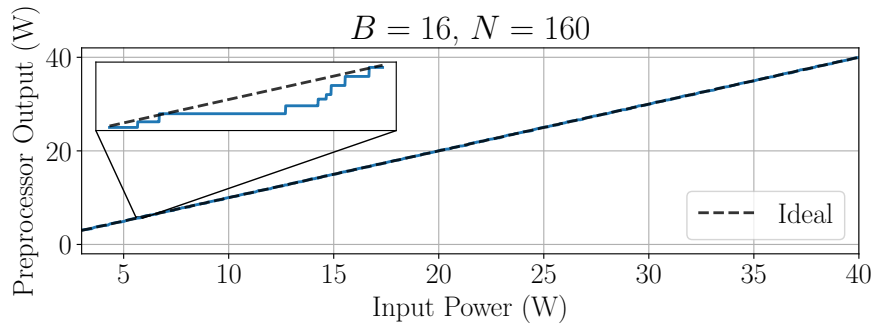
This number of outputs can be characterized in terms of bits by taking $\log_2 U$. For example, for preprocessors with input bits $B = 12$ and $B = 16$ averaged over $N = 133$ data points, the number of output bits are approximately 15 and 19, respectively.

Clearly, there are practical resolution benefits from averaging. Current measurements are quantized in a stair-step fashion, with equal spacing between each bit level. The averaging inherent to the preprocessor smooths this quantization. However, the preprocessor “transfer characteristic,” the mapping between actual input power and the preprocessor output level, is nonlinear.

Consider a range of resistive loads powered by a 120 V rms sinusoidal source. Observed current amplitude is swept from approximately 35 mA to 471 mA, corresponding to a sweep in power consumption from 3 W to 40 W. To illustrate the nonlinearity of the spectral envelope calculation, contrast the outputs of two different preprocessors, one operating with 12-bit ($B = 12$) input sampling and the other operating with 16-bit ($B = 16$) sampling, both with $N = 160$ and a maximum quantizable current $I_{max} = 300$ A (corresponding to approximately 25.5 kW for this system). All examples use a current sampling rate of 8 kHz; this preprocessor with $N = 160$ thus



(a) 12-bit preprocessor.



(b) 16-bit preprocessor.

Figure 2-3: Preprocessor transfer characteristics for a sinusoidal input current.

outputs at 50 Hz. The resulting computations for P from Eq. (2.8) are shown in Fig. 2-3. The 12-bit preprocessor in Fig. 2-3a shows visible “bumps” and sharp edges forming nonlinearities in the transfer characteristic. The width of the first bump in the plot approximately corresponds to the difference in input power for one least significant bit (LSB) of quantized input current. Fig. 2-3b shows a 16-bit preprocessor that appears more linear at the same scaling. However, the nonlinearities are still present, as shown in the inset of Fig. 2-3b. The deviation of each “bump” from an ideal linear transfer characteristic is most noticeable at the lower end of the input power range. The transfer characteristic becomes more linear as the input amplitude is increased. Thus, for measurement of small loads or for preprocessors with relatively few input quantization bits, these nonlinearities or distortions can cause an appreciable bias of the preprocessor output away from the correct value, both for single readings and measured differences in readings.

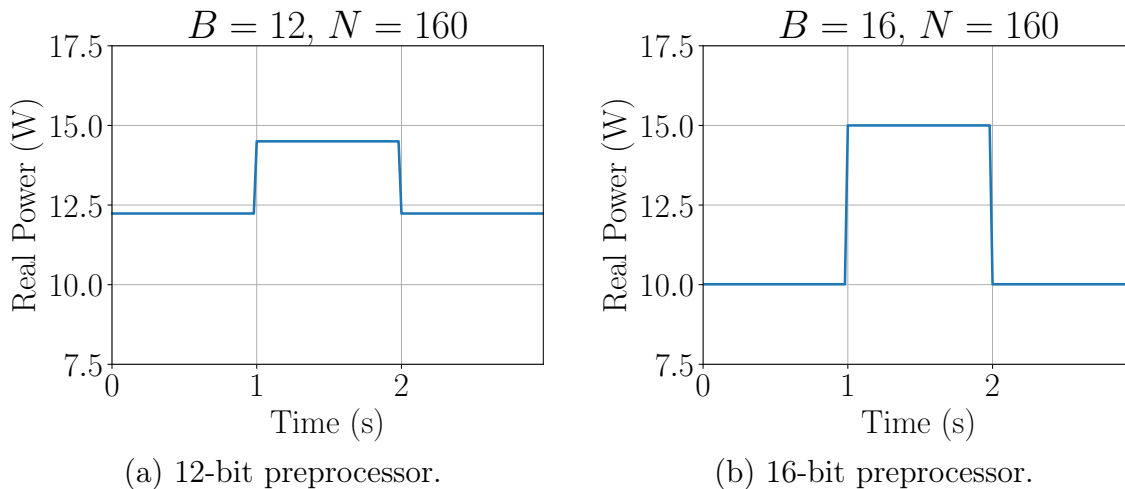


Figure 2-4: Simulation of preprocessor nonlinearity for a 5 W load cycling on top of a 10 W base load.

Deviations from the ideal linear transfer characteristic can introduce significant error in estimating the actual power associated with a load. For example, Fig. 2-4 shows the outputs of the 12- and 16-bit preprocessors for a 5 W resistive load cycling on top of a 10 W base load. The 12-bit preprocessor output is inaccurate at every point in Fig. 2-4a. The step change shows a difference of 2.27 W, significantly different than the actual 5 W load demand. That is, even a simple event detector looking at changes in steady-state power demand would not find the expected 5 W change. The 16-bit preprocessor’s output, shown in Fig. 2-4b, is much more accurate. The output value for the base load is 10.01 W, and the output power of the base load and cycling load together is 15 W.

The 12-bit preprocessor is unable to adequately resolve a 5 W difference with the 10 W base load. However, in regions where the slope of the preprocessor output to preprocessor input is approximately linear, the differential change would be closer to the actual 5 W change, even if the measurement at any single point is incorrect. This can be seen in Fig. 2-5 for a base load of 7.45 W. The values before and after the first step change in Fig. 2-5a are incorrect, at 8.53 W and 13.65 W respectively, but the difference is close to actual, at 5.12 W. For the 16-bit preprocessor, as shown in Fig. 2-5b, both the single and differential output measurements are close to the

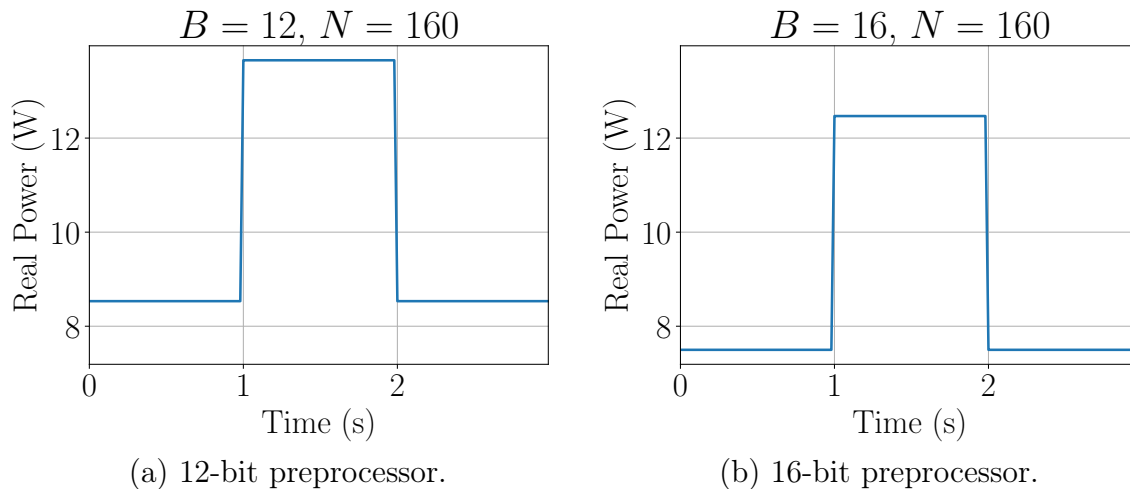


Figure 2-5: Preprocessor simulation showing correct differential measurements.

actual values. The values before and after the first step change are 7.5 W and 12.47 W respectively, and the difference is 4.97 W. In a noise-free, idealized environment, the preprocessor’s ability to make accurate measurements, both single and differential, depends on the value of the signal being measured.

2.2.2 Practical Preprocessor Performance

Linearity can be improved by increasing the input bit resolution. However, in a practical system, there are also other factors, such as noise at different stages in the signal processing chain, that can impact the preprocessor performance. Any practical system will contain some non-zero amount of prequantization noise. Due to the uncertainty in each measurement, noise reduces the number of meaningful preprocessor output values. However, prequantization noise can be either a detriment or a benefit to system performance, depending on the amount of noise and the preprocessor parameters.

2.2.2.1 Prequantization Noise

Analog distortions in the current signal prior to the quantization stage produce prequantization noise. Prequantization noise arises from many sources, including the

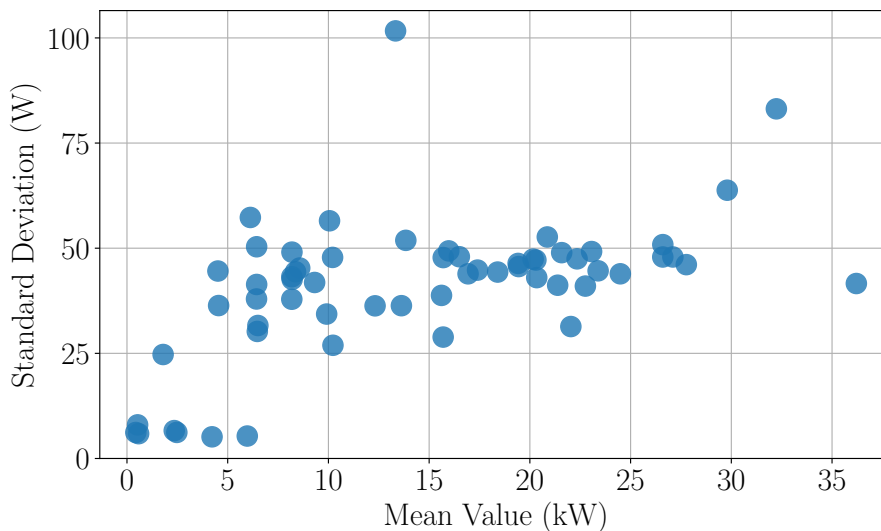


Figure 2-6: Preprocessor output standard deviation and mean value for several intervals of shipboard data.

thermal noise in the amplifiers of the signal processing chain of the data acquisition (DAQ) hardware [32]. Distortions can also result from electromagnetic coupling in sensors and connections external to the DAQ. Another source of distortion is high-frequency current components from the physical operation of other loads on the line, such as nonlinear loads and inductive loads [33, 34]. While not noise in the strict sense, these fluctuations in power drawn by other loads can affect the preprocessor calculations and resolution.

The microgrid of USCGC Marlin serves as a demonstration. Preprocessor hardware was used to monitor the aggregate current of a subpanel which powers approximately half of the ship. The variance in the preprocessor output for real power (computed as the sum of three phases of real power envelopes) tracks with the mean value of the preprocessor output in steady state. Fig. 2-6 shows a scatter plot of standard deviation versus mean of real power spectral envelopes for the aggregate shipboard power in steady-state operation. Low-variance values in the bottom left of the scatter plot include resistive loads like heaters. Steady-state windows with higher variance include the operation of loads with significant harmonics or large fluctuations in power draw. An important component of the preprocessor output variance in this industrial environment arises from these fluctuations.

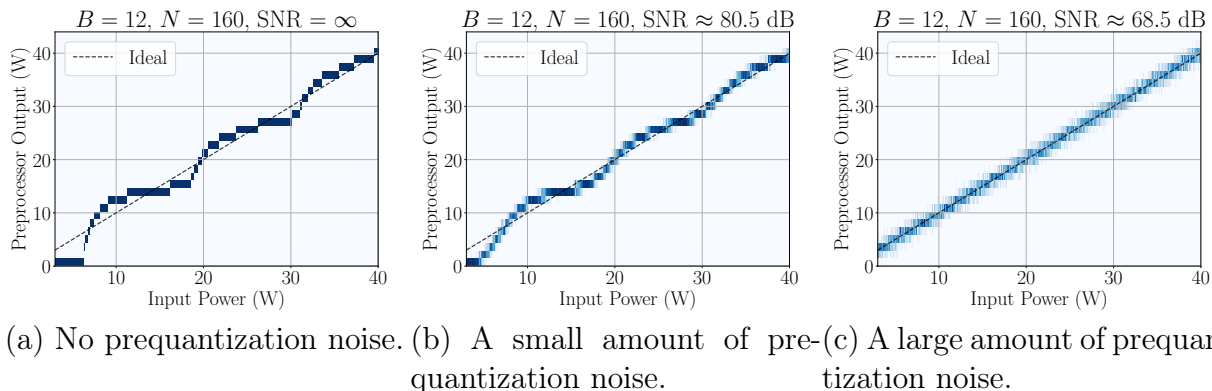


Figure 2-7: Preprocessor outputs as histograms for calculated input power values plus Gaussian noise, for a 12-bit quantizer.

These sources of prequantization noise are modeled as uncorrelated additive Gaussian white noise (AGWN). The amount of noise introduced into a system is characterized using a signal-to-noise ratio (SNR). Consider a system with a maximum quantizable current of I_{max} and an rms current noise of σ_G . The maximum quantizable sinusoidal current has rms value $I_{rms} = I_{max}/\sqrt{2}$, and the SNR is defined as:

$$\text{SNR} = 20 \log_{10} \frac{I_{rms}}{\sigma_G} = 20 \log_{10} \frac{I_{max}}{\sigma_G \sqrt{2}}. \quad (2.10)$$

2.2.2.2 Transfer Characteristic Linearization

Adding or exploiting a small amount of Gaussian white noise, that is, dither, into an analog-to-digital converter or quantization system linearizes the mean output value of the quantizer [35]. The beneficial dithering effect can improve the accuracy of an acquired current signal, and can decrease error in the preprocessor output [9]. The dithering benefit of prequantization noise “propagates” through the preprocessor system, and as a result, the relationship of the mean output versus input gradually becomes more linear as noise is added to the system. This effect is shown in Fig. 2-7, where prequantization noise is gradually added to a system with $B = 12$ and $N = 160$. Here, the preprocessor transfer characteristic is shown as a series of vertical histograms for a range of input values, with the average of each histogram forming an “average transfer characteristic.” The highly nonlinear average transfer characteristic

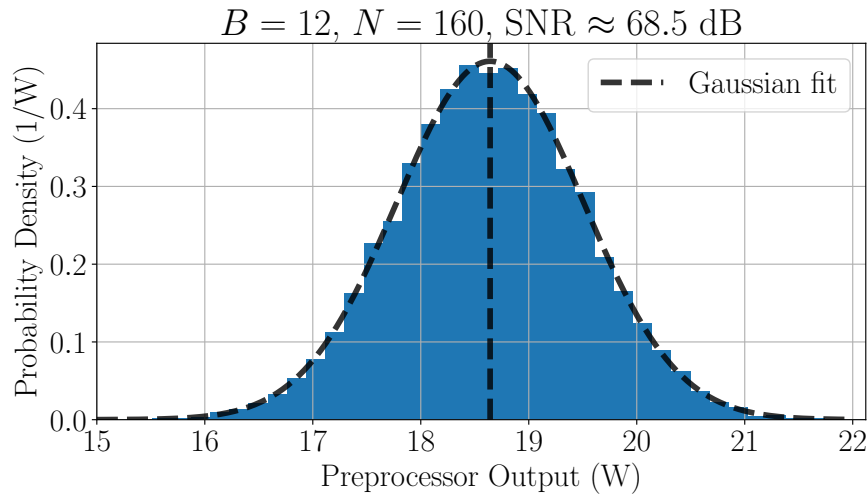


Figure 2-8: Histogram of preprocessor outputs for an input power of 18.6 W with the same parameters as Fig. 2-7c.

of Fig. 2-7a becomes more linear when a small amount of prequantization noise is added in Fig. 2-7b, and becomes even more linear when sufficient noise is added in Fig. 2-7c. The preprocessor average transfer characteristic becomes linear at an rms noise value of approximately one half of the least significant bit of the input quantizer. In Fig. 2-8, a vertical cross-section of Fig. 2-7c is shown for an input power of 18.6 W, corresponding to a point where the transfer characteristic is highly nonlinear in the noise-free preprocessor of Fig. 2-3a. A Gaussian distribution is fit with the calculated mean and standard deviation. As can be seen in the plot, the mean output from this preprocessor is approximately equal to the input power of 18.6 W. However, the correct mean value does not come without drawbacks. The variance from the noise increases the uncertainty in measurement, as evident by the higher vertical spread of the histograms of Fig. 2-7b and Fig. 2-7c. Measurements of the preprocessor output will therefore appear to be “noisy.”

The same load cycle as in Fig. 2-4 is simulated again, but this time 80 mA rms of Gaussian current noise is added, for an SNR of 68.5 dB (the same amount of noise as in Fig. 2-7c). The resulting P streams are shown in Fig. 2-9. For the 12-bit preprocessor in Fig. 2-9a, the average value before the step change is around 10.14 W and the average value after the first step change, between 1 and 2 seconds,

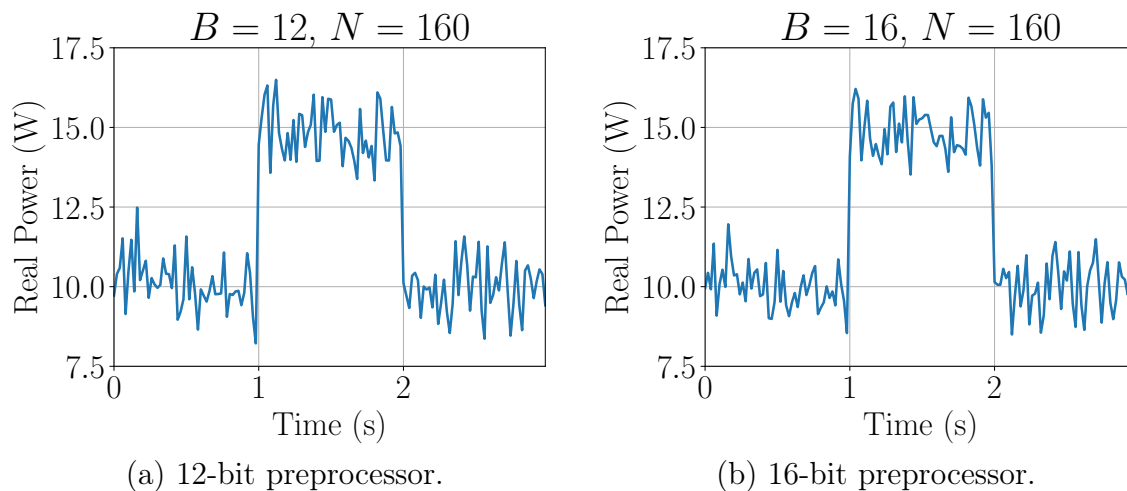


Figure 2-9: Simulation of preprocessor linearity from noise for a small load while a base load is running.

is around 14.76 W. The differential measurement is much closer to 5 W, the true power consumption of the load, than the result in Fig. 2-4a. The noise linearizes the transfer characteristic and leads to more accurate average single and differential measurements. For the 16-bit preprocessor in Fig. 2-9b these values are around 10.06 W and 15.24 W respectively. Here, the average single and differential measurements are similar to those observed in Fig. 2-4b. The noise-free 16-bit preprocessor transfer characteristic was already close to linear. Thus, adding noise did not significantly change the average, but it did create more uncertainty in any single output power measurement.

2.2.2.3 Effective Number of Bits

A reasonable level of input prequantization noise helps linearize the preprocessor performance. More than this level degrades the output estimates with undesirable noise. To quantify the effect of prequantization noise on resolution ability, the system’s “effective number of bits” is examined. Per IEEE standard 1057 [36], effective number of bits (ENOB) is defined as the number of bits of a theoretical quantizer whose rms quantization noise equals the rms noise of the system in question. For some system

yielding output x , the ENOB is:

$$\text{ENOB} = \log_2 \left(\frac{x_{pk-pk}}{\sigma_x \sqrt{12}} \right), \quad (2.11)$$

where x_{pk-pk} is the full range of the system output and σ_x is the rms noise and distortion of x . A liberal use of this definition will allow a derivation of the ENOB of the spectral envelope preprocessor output, given some amount of prequantization noise. With the results of this derivation, the effect of noise on the performance of the preprocessor in terms of its number of meaningful “bits” can be examined.

The goal is to find both P_{pk-pk} , the full-scale swing possible in preprocessor output, and σ_P , the standard deviation in the preprocessor output due to input noise. For this derivation, σ_P is assumed to be constant across all input power levels. The input current signal is now treated as a random process, $I[n]$. The equation for the fundamental real power spectral envelope (P), for a window size of N , is given in Eq. (2.8). It is further assumed that the input noise, represented here as $I_g[n]$, is white and Gaussian, and has zero mean and some variance σ_G^2 . In order to show the effect of current signal quantization on the preprocessor input, another assumption is made that the distortion from quantization, represented as $I_q[n]$, can be approximated as white uniform noise with zero mean and variance σ_Q^2 [30]. For a quantizer that operates between $-I_{max}$ and $+I_{max}$:

$$\sigma_Q^2 = \frac{\Delta^2}{12}, \quad \Delta = \frac{I_{max}}{2^{B-1}}, \quad (2.12)$$

where Δ is the size of the least significant bit. These approximations are suitable when the average preprocessor transfer characteristic is well-approximated as linear, such as in Fig. 2-7c. For convenience in derivation, the sum of $I_g[n]$ and $I_q[n]$ will be represented as $I_o[n]$, which will also be white noise with zero mean and variance $\sigma_o^2 = \sigma_G^2 + \sigma_Q^2$ (since $I_g[n]$ and $I_q[n]$ are assumed to be independent). Thus, the full representation of $I[n]$ is as follows:

$$I[n] = i_o[n] + I_g[n] + I_q[n] = i_o[n] + I_o[n], \quad (2.13)$$

where $i_o[n]$ is the deterministic input current signal. Since P is now a random variable, its variance, σ_P^2 can be computed as $E\{P^2\} - E\{P\}^2$, where $E\{\cdot\}$ represents the expected value of a random variable [37]. Due to the linearity of the expected value operator and the fact that $I_o[n]$ is zero-mean, $E\{P\}^2$ can be computed as:

$$E\{P\}^2 = \frac{V_{pk}^2}{N^2} \left(\sum_{n=0}^{N-1} \sin\left(\frac{2\pi n}{N}\right) i_o[n] \right)^2. \quad (2.14)$$

Next, $E\{P^2\}$ can be found by taking the expected value of the expansion of P^2 and recognizing that the expected value of the random cross term will be zero, since it is zero-mean. The expected value of the deterministic square term will be the value of $E\{P^2\}$ found in Eq. (2.14). Thus, σ_P^2 will simply be the expected value of the random square term. By rewriting it as a product of sums with different indices and combining the sums, the following expression for σ_P^2 can be found:

$$\sigma_P^2 = \frac{V_{pk}^2}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \sin\left(\frac{2\pi m}{N}\right) \sin\left(\frac{2\pi n}{N}\right) E\{I_o[m]I_o[n]\}. \quad (2.15)$$

Because $I_o[n]$ was assumed to be white noise, the autocorrelation of $I_o[n]$ is zero except for the zero-shift case, and $E\{I_o[m]I_o[n]\}$ becomes $\sigma_o^2\delta[m-n]$. Through the filtering property of the delta function, the m summation and the delta function can be discarded and m can be set equal to n :

$$\sigma_P^2 = \frac{V_{pk}^2 \cdot \sigma_o^2}{N^2} \sum_{n=0}^{N-1} \sin^2(2\pi n/N). \quad (2.16)$$

Using the power reduction identity and Lagrange's trigonometric identity [38], it can be shown that for any integer $N > 2$, the sum above will simplify to $N/2$. Thus, the expression for σ_P becomes the following, since $V_{rms} = V_{pk}/\sqrt{2}$:

$$\sigma_P = \frac{V_{rms} \cdot \sigma_o}{\sqrt{N}} = \frac{V_{rms} \sqrt{\sigma_G^2 + \sigma_Q^2}}{\sqrt{N}}, \quad N > 2. \quad (2.17)$$

This implies that for some input current noise, the corresponding preprocessor output

noise will be the product of the rms voltage and the rms current noise, but reduced as the number of averaging points increases. This makes intuitive sense, as in many statistical contexts variance tends to decrease in proportion to the number of samples (N) and the standard deviation tends to decrease proportionally with \sqrt{N} . Next, it is assumed that the full-scale preprocessor swing, P_{pk-pk} , is approximately equal to $V_{pk} \cdot I_{max}$, i.e. twice the maximum real power observable, occurring when there is zero phase angle between the voltage and current.

With σ_P and P_{pk-pk} obtained, these quantities can be used in the equation for effective number of bits given in Eq. (2.11). In order to characterize noise as the third parameter in the preprocessing system (alongside B and N), a new parameter G is defined as the linear form of the quantizer's signal-to-noise ratio, as defined in Eq. (2.10),

$$G = \left(\frac{I_{rms}}{\sigma_G} \right)^2 = \frac{1}{2} \left(\frac{I_{max}}{\sigma_G} \right)^2 = 10^{\text{SNR}/10}. \quad (2.18)$$

G is then substituted into Eq. (2.17). σ_Q can be substituted out of this equation and replaced with $\Delta/\sqrt{12}$ due to the assumption made previously about the character of the quantization noise. Since $\Delta = I_{max}/2^{B-1}$ from Eq. (2.12), it can be substituted out of the equation for ENOB. The product and quotient rules for logarithms can also be used, and terms can be cancelled and rearranged to obtain the final representation of the effective number of bits of the preprocessor for a given B , N , and G :

$$\text{ENOB} = B + \frac{1}{2} \log_2 \frac{N}{2} - \frac{1}{2} \log_2 \left(\frac{3 \cdot 4^B}{2G} + 1 \right), \quad N > 2. \quad (2.19)$$

This equation shows that for the effective number of bits of the preprocessor, the number of bits from the quantizer (B) is the baseline, and then some benefit is obtained from increasing the number of averaging points (N), but only logarithmically. If there is no noise, that is, if $G = \infty$ (infinite SNR), then the third term goes to zero, meaning that there is no penalty in the effective number of bits from prequantization noise. However, if there is prequantization noise, it will lower the effective number of bits of the preprocessor approximately logarithmically.

2.2.3 Demonstration with Hardware

Spectral envelope preprocessors using two different data acquisition boards are examined in this section. The first is the LabJack UE9 data acquisition hardware, which obtains data at a 12-bit input resolution. The second is the data acquisition hardware which will be presented in Section 2.3, which obtains data at a 16-bit input resolution. Both boards sample voltage and current signals at a 8 kHz sampling rate, for each phase of the power system. One period is used for the spectral envelope window with no window overlap, so the output stream has a sampling rate equal to the line frequency (60 Hz). Thus, the average value of N is $133.\bar{3}$, or approximately 133. Both DAQs' signal processing chains introduce approximately half an LSB rms of prequantization noise into their preprocessors, resulting in linear average transfer characteristics. Due to the difference in number of bits, this means that the 16-bit DAQ contains less noise than the 12-bit DAQ. The current sensor hardware is the LEM LF-305 which can measure a current of up to 300 A. The quantizer is chosen such that its maximum quantizable value, referred to here as I_{max} , is aligned with the maximum value of the sensor (i.e., $I_{max} = 300$ A). If I_{max} is larger than the maximum sensor value, there will be a degradation in resolution since the LSB will be unnecessarily large. If I_{max} is smaller than the maximum sensor value, there will be clipping in the input signal if a value over I_{max} is acquired.

In Fig. 2-10, the simulated ENOB for the preprocessor output versus signal-to-noise ratio, as derived in Eq. (2.19), has been plotted for the case of 12- and 16-bit preprocessors with $N = 133$. Three operating points are marked. The leftmost operating point is for both the 12- and 16-bit DAQs monitoring the starboard-side aggregate shipboard power of USCGC Marlin (in this context, *aggregate* refers to metering the entire panel and its collection of loads, as opposed to submetering a smaller branch of the power delivery system). For this point, the ENOB of the preprocessor output was obtained by approximating the standard deviation of the steady-state preprocessor output. Since there is so much noise, the calculated ENOB of both of these DAQs is approximately 10.4, meaning the 16-bit DAQ holds no

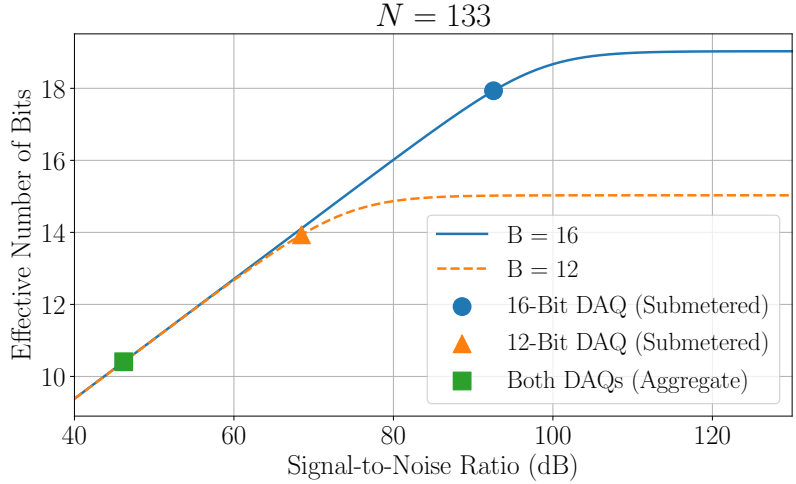
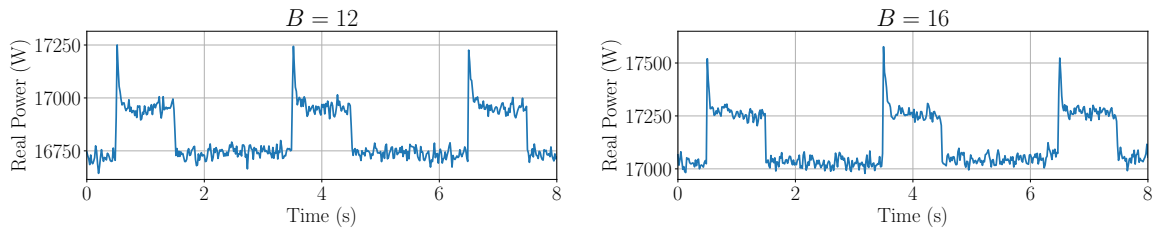
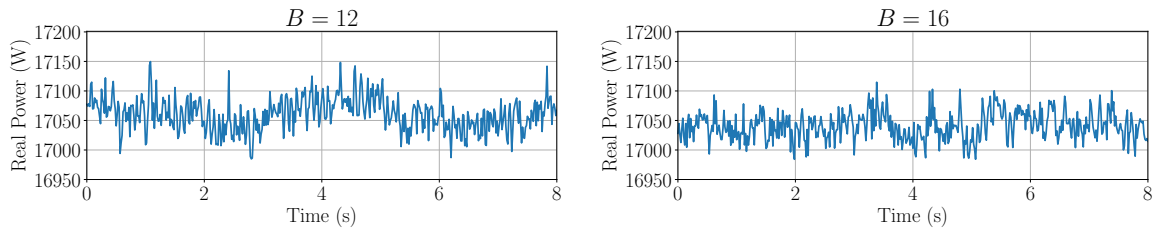


Figure 2-10: Simulated ENOB for 16- and 12-bit preprocessors for a sweep of SNR values. Also plotted are the ENOB from measurements of two DAQs, each in an aggregate (i.e., metering the entire subpanel) and submetered (i.e., metering a smaller subset of the panel) environment.



(a) A 200 W load on the aggregate power stream (12-bit DAQ). (b) A 200 W load on the aggregate power stream (16-bit DAQ).



(c) A 15 W load on the aggregate power stream (12-bit DAQ). (d) A 15 W load on the aggregate power stream (16-bit DAQ).

Figure 2-11: 12- and 16-bit preprocessor outputs in an aggregate shipboard microgrid for 200 W and 15 W loads.

significant advantage here. A comparison of the two DAQs' power streams can be seen in Fig. 2-11 for the starboard-side subpanel of USCGC Marlin. In Fig. 2-11a and Fig. 2-11b, for the 12- and 16-bit DAQs, respectively, an approximately 200 W

resistive load is energized on top of the aggregate shipboard power (at different times, hence at different base loads), turning on at around $t = 0.5$ s and cycling three times for one second on and two seconds off. The load is equally resolvable for both the 12-bit and 16-bit DAQs. In Fig. 2-11c and Fig. 2-11d, an approximately 15 W resistive load is energized on top of the aggregate shipboard power, for the 12- and 16-bit DAQs, respectively. The 15 W resistive load cycles three times, as in the previous scenario, however it is not visible for either DAQ, because of the large fluctuations in power drawn by other loads.

In Fig. 2-10, moving rightward from the leftmost operating point, as the amount of noise decreases (SNR increases), the ENOB of each preprocessor increases essentially linearly until it reaches a point of saturation. For the 12-bit DAQ, the ENOB saturates at around 15 bits, and for the 16-bit DAQ, the ENOB saturates at around 19 bits. Note that these saturated values approximately match the values calculated in Section 2.2.1 using the formula provided in [9]. A second shipboard setup was configured to monitor an individually submetered outlet with the same 12-bit and 16-bit DAQs used previously. In this case, the dominant prequantization noise is thermal noise in the analog processing chain, since no other load currents are being metered. The middle operating point in Fig. 2-10 is for the 12-bit DAQ in a submetered environment with internal noise of $\sigma_G \approx 80$ mA (SNR ≈ 68.5 dB). The rightmost point is for the 16-bit DAQ in a submetered environment, with internal prequantization noise of $\sigma_G \approx 5$ mA (SNR ≈ 92.6 dB). The prequantization noise values can be used to obtain the ENOB using Eq. (2.19). The middle and rightmost operating points yield an ENOB of approximately 14 and 18, respectively. These are only slightly lower than the maximum saturated value, due to the low amount of noise.

2.2.4 Design Guide

Design of a spectral envelope preprocessing system requires consideration of three parameters: number of quantization bits, B ; window length, N ; and signal-to-noise ratio, SNR. The minimum resolvable load of the target application is another parameter that is either tunable or fixed for a given system. The minimum resolvable load

can be expressed either with ENOB or with a ratio of minimum to maximum resolvable powers. This section guides the design of a spectral envelope preprocessing system based on parameters that are either tunable or fixed for a particular system.

2.2.4.1 Measuring System Noise

The preprocessor quantizer's SNR is the ratio of the square of the rms maximum quantizable input current (I_{rms}^2) to the variance of the prequantization noise in the current signal (σ_G^2), as defined in Eq. (2.10). To determine the SNR, the distribution of the noise is approximated as a zero-mean Gaussian, whose variance must be either acquired or designed. If the variance is not known *a priori*, it can be approximated in multiple ways. If the current signal can be measured with no loads drawing power, the standard deviation of this signal can be computed over a window. If current signal data is only available for an energized system, a window of the current signal for steady-state power consumption can be obtained, and then a three-parameter sine-wave fitting algorithm can be applied [39]. The generated sinusoid can then be subtracted from the current signal and the standard deviation obtained from this resulting difference signal. This method has the advantage that if the monitored system contains variance that is not independent of the loads energized, such as that in Fig. 2-6, it can find an approximation of the standard deviation for any load configuration. Finally, if the preprocessor transfer characteristic can be assumed to be linear due to sufficient prequantization noise, the standard deviation of the preprocessor output (σ_P) over a window can be calculated. The standard deviation of the quantized current signal (σ_G) can be approximated by rearranging Eq. (2.17):

$$\sigma_G = \sqrt{\frac{\sigma_P^2 N}{V_{rms}^2} - \sigma_Q^2}. \quad (2.20)$$

2.2.4.2 Choosing the Number of Bits

As shown previously in Fig. 2-11, increasing the number of quantizer bits in the presence of excessive noise may be fruitless. Thus, when designing a system to operate

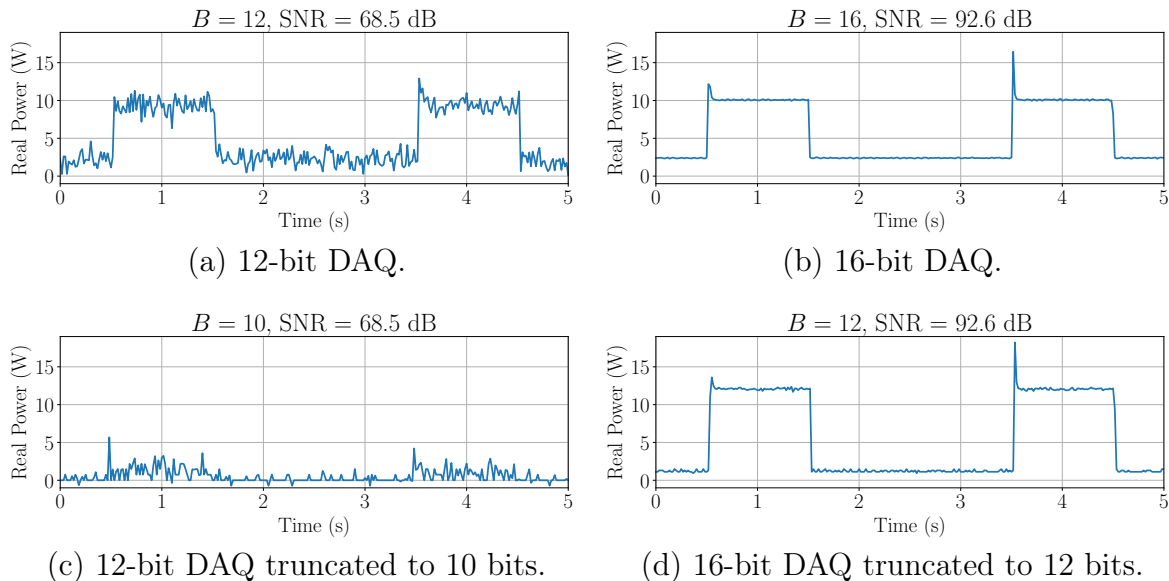


Figure 2-12: Comparison of four different preprocessors for a nominal 7.5 W load turn-on on an approximately 2.5 W base load.

in noisy conditions, the knowledge of the SNR and window length N can be used with Eq. (2.19) to compare various candidate values for B , and assess whether the marginal change in effective number of bits of the preprocessor is worth the extra expense of the increase in B . Due to the approximations made in the derivation of this equation, it should only be used when the preprocessor average transfer characteristic can be assumed to be approximately linear due to sufficient prequantization noise. For the case when the preprocessor transfer characteristic is nonlinear, increasing B will result in an appreciable benefit in the preprocessor’s ability to resolve small loads by reducing the size of the nonlinear regions.

For example, Fig. 2-12 shows a comparison of four different preprocessors for a nominal 7.5 W load energized on top of an approximately 2.5 W base load (corresponding to the smart plug used for cycling the load) in the submetered environment on USCGC Marlin. Fig. 2-12a shows the resulting power stream for the 12-bit DAQ. The prequantization noise linearizes the average preprocessor output, leading to a difference in the average steady state values of 7.06 W, close to the correct value. Using the 16-bit DAQ, as shown in Fig. 2-12b, the steady-state and difference in steady-state power values are correct. The difference in steady-state when the load

turns on is approximately 7.5 W. Then, the number of bits is artificially reduced through truncation for both DAQs. Fig. 2-12c shows the power stream for the 12-bit DAQ downquantized to $B = 10$. In this case, the load becomes barely visible. Reducing the 16-bit DAQ to $B = 12$ results in a deterioration of accuracy, as shown in Fig. 2-12d. Due to low noise, the transfer characteristic of this case is nonlinear and the difference in power values when the load is turned on is approximately 10.8 W, which is clearly incorrect.

2.2.4.3 Choosing the Sampling Rate

The parameter N governs the number of data points in the spectral envelope averaging window. For the case of a window length of one period, N is the ratio of sampling frequency to operating frequency. Broadly, N controls the shape and smoothness of the transfer characteristic nonlinearities compared to the “stair-step” transfer characteristic of a simple quantizer. However, the marginal benefit of increasing N drops off logarithmically. If the sampling frequency is not an integer multiple of the operating frequency, then N will alternate between different values for each spectral envelope calculation window. For example, for a sampling frequency of 8 kHz and an operating frequency of 60 Hz, every third window will have an N of 134, and two out of every three windows will have an N of 133, resulting in an average N of $133.\bar{3}$. As a result of inconsistent window sizes, the preprocessor output for this situation will appear to have “spikes” every third sample, even for what should be a constant value. This can be removed by using a rolling average of length three, at the expense of smoothing the transients and potentially reducing the ability to identify load transients. These fluctuations may be inconsequential if there is a high amount of variance in the signal. Another consideration is that the line frequency is not always at a consistent 60 Hz, especially for systems with a limited generation capacity. By utilizing the tracked line frequency [9], the actual fundamental component can be better determined. The line frequency output can be used as input to the generalized Goertzel algorithm [40], which computes the k^{th} DFT coefficient, where k can be a non-integer value.

B and N are typically not independent for a given ADC hardware. Thus, it

is also important to consider the relationship between B and N given the specific hardware constraints. As shown, for a given hardware with a fixed B , oversampling and averaging via spectral envelopes can increase the ENOB. Eq. (2.19) shows that there is benefit to increasing N . On the other hand, increasing the sampling frequency, and consequently N , may involve a loss of precision depending on the ADC hardware.

2.2.4.4 Ability to Resolve Small Loads

For a given system installation, there exists some minimum load power consumption that can be resolved. Spectral envelope preprocessing introduces two issues that impact the ability of a system to resolve small loads. First, the nonlinearity of the noise-free preprocessor transfer characteristic results in a reduction of resolution at certain input values, as shown in Fig. 2-4. Second, significant prequantization noise reduces the number of meaningful output values even with an increase in number of quantizer bits. For both cases, it is useful to examine the preprocessor's ability to differentiate a small load event given values for B , N , and some base load.

If the prequantization noise is sufficient to linearize the average preprocessor transfer characteristic, the minimum resolvable power will be independent of any base load value. The minimum power that can be resolved by the preprocessor, P_{min} , can be estimated as σ_P . The linear signal-to-noise ratio G from Eq. (2.18) can be related to σ_P from Eq. (2.17). $I_{max}/\sqrt{2G}$ can be substituted for σ_G and $I_{max}/(2^{B-1}\sqrt{12})$ for σ_Q . Since the maximum power detectable by the system can be approximated as $P_{max} = V_{rms} \cdot I_{rms}$, where $I_{rms} = I_{max}/\sqrt{2}$, the following equation for P_{min}/P_{max} can be derived:

$$\frac{P_{min}}{P_{max}} = \sqrt{\frac{1}{N} \left(\frac{1}{G} + \frac{2}{3 \cdot 4^B} \right)}. \quad (2.21)$$

For example, this can be calculated for the USCGC Marlin example in Fig. 2-11b over a window between $t = 1.75$ s and $t = 3.25$ s. The standard deviation of the real power preprocessor output is computed to be approximately 22.7 W, which will be used as P_{min} . Since the rms voltage is 254 V and the maximum quantizable

current signal is 300 A, P_{max} is found to be 53881.5 W. Thus, the ratio of P_{min} to P_{max} is 1 : 2374. This implies that all other parameters being constant, increasing the maximum observable power (likely by changing I_{max}) will increase the minimum observable load power by the same factor, providing a heuristic for the range of observable values. P_{min}/P_{max} can be related to ENOB through the following equation:

$$\text{ENOB} = -\log_2 \left(\sqrt{3} \cdot \frac{P_{min}}{P_{max}} \right). \quad (2.22)$$

Either P_{min}/P_{max} or ENOB can be used to quantify the ability of a system to resolve small loads.

If prequantization noise does not linearize the average preprocessor transfer characteristic, the system's resolution will be lower at the nonlinear regions of the transfer characteristic. For example, in Fig. 2-3a, between an input power of approximately 12 W to 18 W, there will be very little difference in the output value assigned. What counts as "resolvable" is application-specific. For example, consider a correlation matching algorithm with a set tolerance [5]. If a match tolerance of 20% is chosen, a load event is considered to be resolvable if the difference in preprocessor outputs is within 20% of the actual load value. Considering the base and small load scenario, different base load values can be iterated, and for each, the minimum small load value is found that yields a preprocessor output difference within the desired tolerance. Finding a closed-form solution for this value for any given match tolerance is not tractable, but in general, the worst case is at the first inflection point, when the preprocessor output becomes non-zero (e.g., at approximately 6.2 W in the Fig. 2-3a). For power monitoring applications, it is unlikely that the preprocessor would operate in this nonlinear situation, as only approximately half of an LSB of rms noise current signal is required to linearize the average transfer characteristic.

2.2.4.5 Design Scenarios

The presented techniques can inform system design. Given three fixed or desired parameters, Eq. (2.21) can be used to solve for the variable of interest. In the first

scenario, N and SNR are fixed and there is some requirement on the minimum resolvable load, which fixes P_{min}/P_{max} . This scenario could be brought about by design of data acquisition hardware at some required sampling frequency in an environment with a known amount of noise. Here, solving for B will yield the number of input quantization bits necessary to meet the resolution requirement. However, it is not always possible to meet the desired requirements by only tuning B , as there is no realizable value of B that will yield a P_{min}/P_{max} of less than $\sqrt{1/(NG)}$. In the second scenario, N and B are fixed, perhaps due to use of specified data acquisition hardware, and P_{min}/P_{max} is set to some desired value. Solving for the minimum SNR for such a resolution requirement allows a system designer to evaluate potential sensor placements. That is, can a sensor be placed “upstream” to monitor a larger and most likely noisier system, or are multiple “downstream” sensors required? Similar to the first scenario, P_{min}/P_{max} is lower bounded. In this scenario, the lower bound is $\sqrt{2/(3 \cdot 4^B \cdot N)}$, so even an infinite SNR will not meet every requirement. Finally, in the third scenario, N and B are again fixed due to prespecified hardware. SNR is also fixed in this scenario, which may be due to unavoidable noise in the installation environment. By solving for P_{min}/P_{max} , the designer can understand which loads can and cannot be resolved in this installation. That is, the ratio of minimum resolvable power to maximum resolvable power can be derived for the preprocessor configuration to describe the system’s ability to resolve small loads.

2.3 Measurement Hardware

As was shown, the ac utility effectively creates a substantial “carrier frequency” component at the utility frequency that indicates real and reactive power demand. Valuable information for diagnostics, control, and energy scorekeeping may be found over a range of frequencies and amplitudes in utility waveforms. Other frequency content, for example, in the current waveforms demanded by a load, can serve as important tell-tale signatures for identifying loads and recognizing operating state and fault conditions. The additional frequency content may be at harmonic multiples of the utility

frequency, or distributed in other ways, for example, as a multiple or function of both utility frequency and slip frequency in a motor. The relative amplitude of signals like current at the utility frequency versus motor slot harmonic frequencies may vary over several orders of magnitude. This creates a significant practical problem in achieving an adequate dynamic range to observe all signals of interest. The problem is compounded by normal variations in the utility frequency, which complicates the design of data acquisition instrumentation for making complete measurements of interest.

Knowledge of shaft speed assists with many control and monitoring applications for electric machines [41–46]. Using intrusive approaches for speed estimation adds sensors to machines, adding complexity and cost. Installation of the sensors, cables for power and communication, and interface circuits requires additional work and space. Nonintrusive, sensorless speed monitoring can reliably estimate speed with varying approaches and effort. For induction machines, motor speed can be determined from observed electrical waveforms by tracking slot harmonics [47]. However, three problems challenge efforts to implement practical nonintrusive speed sensors for electric machines. First, slot harmonic signals are much smaller in size and located in higher frequency bands when compared with the fundamental component of current. Notch filters have been applied to attenuate the line frequency component and amplify the high-frequency harmonics [47]. Subtle variations in the utility frequency can defeat the efficacy of this approach as utility waveforms slide out of the stopband of a tightly tuned filter. Second, as shaft speed changes with mechanical load, the slot harmonic frequencies change with changing machine slip. Complex circuit solutions can phase-lock to these changes, but digital signal processing methods permit more flexible implementations assuming a reliable measurement of slot harmonics can be made. Third, because an electric machine typically generates a family of harmonics, tracking applications can become confused by coincident changes in utility frequency and higher harmonics, for example, slip-related frequency [47, 48].

This section presents a custom design for data acquisition hardware that can automatically track and reject carrier (utility) frequency content in a nonintrusive current measurement, permitting the resolution of subtle higher harmonic content

with the full range of an available analog-to-digital converter (ADC). This new instrumentation design automatically tracks variations in utility frequency and adapts to ensure reliable measurements of relatively small signals which are present in the waveform of interest. The hardware was designed by Kahyun Lee and Łukasz Huchel and presented in [49]. This hardware was made compatible with the NILM software suite, as detailed in this section. Demonstrations of the hardware are presented from a field installation on USCGC Marlin. The hardware is further demonstrated in Section 6.1 with a proposed nonintrusive speed estimation method. The proposed instrumentation and harmonic tracking algorithm permit enhanced nonintrusive load monitoring. A shipboard system provides challenges such as a multi-motor environment and changing supply frequencies while at sea. The data acquisition board with an automatically tunable notch filter tracks utility frequency and provides high resolution measurements of higher harmonics as the utility frequency experiences inevitable variations.

2.3.1 Frequency Signatures

Most impedance loads draw current signatures with frequency content beyond the utility frequency. Induction motors, an industry workhorse, for example, have rotor slots that create harmonics that affect phase currents, voltages, and machine fluxes. These can be observed at frequencies:

$$f_h = f_s \left[(kR \pm n_d) \frac{1-s}{p} \pm v \right], \quad (2.23)$$

where f_s is the supply frequency, the non-zero integer k is the order of rotor slot harmonics, R is number of rotor slots, p is the number of pole pairs in the motor (the number of poles divided by two), the non-negative integer n_d is the order of rotor eccentricity or decentering as it rotates with respect to the stator, s is the rotor slip, and the odd integer v is the order of stator magneto-motive force (MMF)

harmonics [49, 50]. Rotor slip, s , is calculated as:

$$s = \frac{n_s - n}{n_s}, \quad (2.24)$$

where n_s is the synchronous speed in rpm and n is the measured rotor speed. For a mechanically ideal motor ($n_d = 0$) that is fed with a pure sinusoidal supply ($v = 1$) the harmonic content in the line current demanded by the machine is expected to be distributed around the principle slot harmonic (PSH) at $k = 1$:

$$f_{psh} = f_s \left[\frac{R}{p}(1 - s) \pm 1 \right]. \quad (2.25)$$

However, if the motor is not mechanically ideal as is generally the practical situation, there will also be harmonic content at slot harmonic frequencies for different values of n_d . As shown in Eq. (2.23), three factors are involved in the creating of slot harmonics: rotor slots, rotor eccentricity, and stator MMF harmonics. Harmonics at different frequencies are generated by different combinations of k , n_d , and v . Fig. 2-13 shows a measured example of the current spectrum of a three-phase induction motor with a 60 Hz supply. The frequency spectrum was obtained by the DFT of the motor current data sampled for 10 seconds. The test motor with 18 rotor slots and 2 pole pairs was running at 1657.7 rpm. There are several peaks at different frequencies, with labels indicating where a current harmonic is present.

Often, as seen for the test motor, the PSH has the largest amplitude. With R and p fixed by the motor structure, the frequency of the PSH depends only on the rotor slip and the supply frequency. Thus, tracking PSH is an effective way to estimate the rotor slip:

$$\hat{s} = 1 - \frac{p}{R} \left(\frac{\hat{f}_{psh} - f_s}{f_s} \right), \quad (2.26)$$

which typically ranges for practical induction machines between 0% and 5%. The PSH is therefore an example of a valuable harmonic that can be measured, in principle, by a nonintrusive power monitor to track machine operation and to differentiate the activity and operation of several machines on the same power service. Unfortunately,

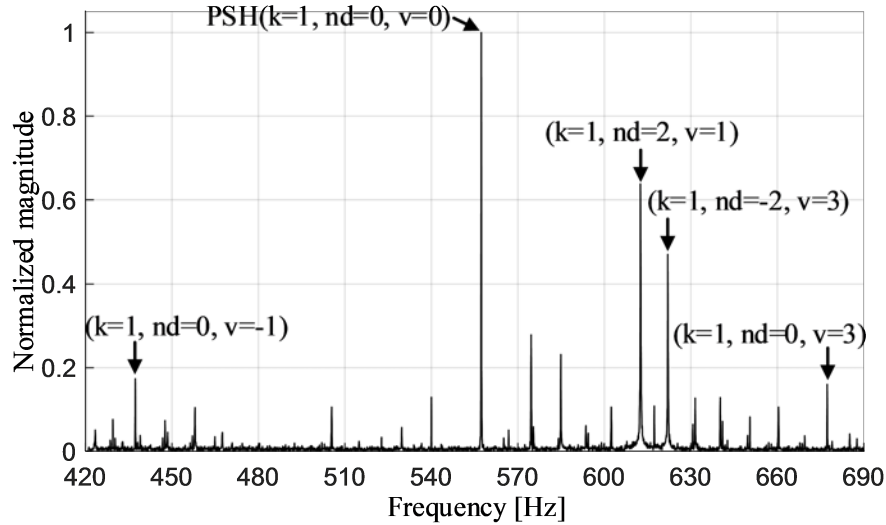


Figure 2-13: Slot harmonics of the test motor at 1657.7 rpm.

the PSH may be several orders of magnitude smaller in amplitude than the utility frequency fundamental current. It is therefore difficult to locate. This problem is exacerbated by any variations in the utility frequency, which happen frequently, and which complicates the design of a filter to reject fundamental frequency current.

2.3.2 Data Acquisition, Processing, and Filtering

To improve the detectability of rotor slot harmonics, this section gives an overview of the architecture of a data acquisition, processing, and filtering board that can selectively and adaptively reject larger signal components in favor of much smaller harmonics. Fig. 2-14 shows a block diagram of the design, which provides eight-channels of simultaneously sampled data at sampling rates as high as 24 kHz. Two key features offered by the design include a zero-crossing detector for line frequency estimation and a switched capacitor (SC) notch filter with tunable center frequency. A microcontroller coordinates the operation of the SC filter and data sampling. One of the eight input channel signals is selected for the slot harmonic detection by the microcontroller using an 8:1 multiplexer. Fig. 2-15 shows the acquisition system, with different functional regions of the board identified.

The zero-crossing detector allows the microcontroller to detect the frequency of

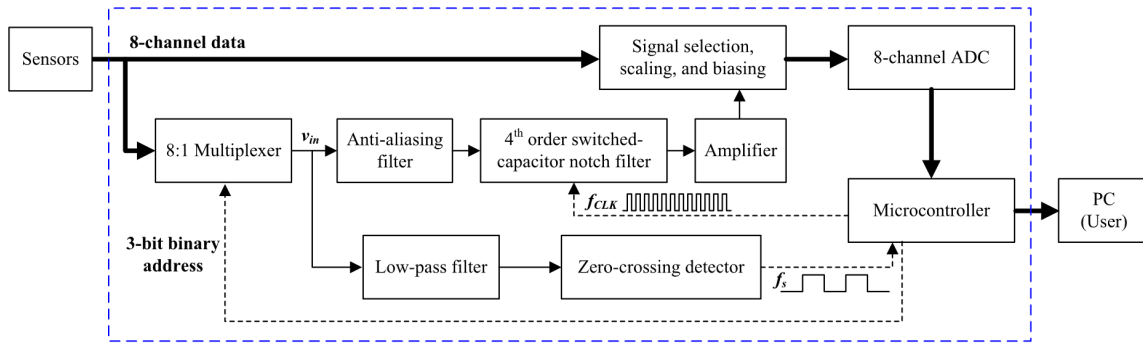
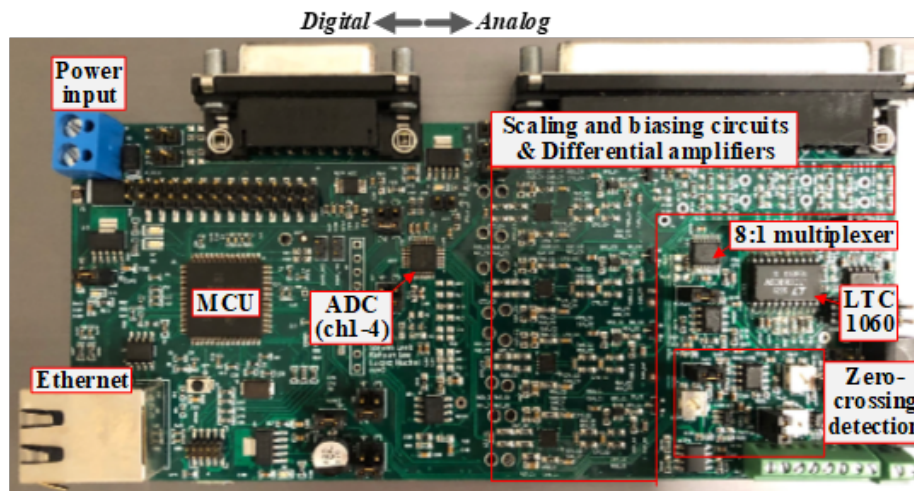
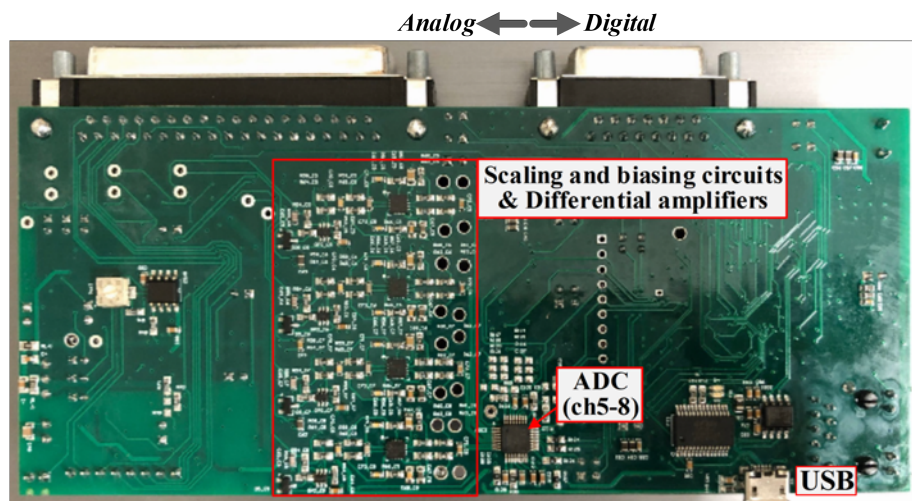


Figure 2-14: Block diagram of the data acquisition, processing, and filtering board.



(a) Top view.



(b) Bottom view.

Figure 2-15: Data acquisition board.

the dominant (e.g., utility frequency) fundamental component in the chosen signal. It consists of op-amp circuits for scaling and biasing, and a hysteresis comparator. The front-end lowpass filter eliminates input noise that can cause undesirable transitions in the comparator output and also attenuates harmonics other than the fundamental. Then, the comparator generates a pulse signal from which the microcontroller estimates the fundamental frequency of the signal. A hysteresis comparator with adjustable hysteresis band helps ensure accurate zero-crossing detection with a potential phase delay that does not affect fundamental frequency estimation.

The real-time estimates of frequency are used to center or tune a switched-capacitor (SC) notch filter to eliminate utility frequency current. Considering that the SC notch filter is a discrete-time filter, a second-order low-pass filter is applied before the SC filter as an antialiasing filter to restrict the bandwidth of the input signal [30]. The cutoff frequency of the antialiasing filter is set to 3.1 kHz, much higher than the input frequency band of interest, allowing for a tradeoff between aliasing and bandwidth. Then, the SC notch filter is implemented with a cascade of two second-order notch filters constructed using the LTC 1060 from Linear Technology. The center frequencies are adjustable with an external clock provided by the microcontroller. The microcontroller adapts the clock frequency to follow the fundamental signal, and therefore allows the notch filter to effectively attenuate the line frequency component even if the line frequency changes.

A total of eight signals are selected for analog-to-digital conversion, based on user configuration. They are scaled and biased to fit the input range of the ADC, an ADS131A04 from Texas Instruments, which allows simultaneous sampling of four channels at high data rates. Two ADCs are daisy chained to perform simultaneous sampling of all eight channels of data. Serial digital data are passed to the microcontroller using the Serial Peripheral Interface (SPI) connection at 16-bit resolution and 8-kHz sampling rate for our experiments. The prototype board allows the user to receive data via Ethernet, USB, UART, or I2C. In our experiments, six of the available eight channels sample the three-phase voltages and currents and one channel is for a filtered current waveform with the fundamental removed. This filtered waveform can

be inspected for subtle harmonics.

2.3.3 Integration with NILM

The data acquisition board was designed to have the same form factor as the LabJack UE9 [51], which is the data acquisition board that has typically been used for NILM installations. Thus, the new data acquisition board can be installed in the existing NILM meter box. The data acquisition board was integrated with the NILM software suite, which automates the data acquisition and processing. In a configuration file, users can specify which data acquisition hardware to use (either the new described hardware, referred to as “NerdJack,” or the LabJack UE9, referred to as “LabJack”). If using the new data acquisition board, users can easily specify the channel (one of the three current streams or three voltage streams) to use for notch filtering. An example snippet of the relevant settings of the configuration file is shown below:

```
meter1:
  type: contact
  enabled: true # set to true or false
  daq_type: nerdjack # labjack or nerdjack
  ip_address: 192.168.1.209 # default LJ or NJ address
  phases: 3 # 1 - 3
  sensors:
    voltage:
      sensor_indices: [3,4,5] # maps to phase A,B,C
      sensor_scales: 0.0919 # built-in constant
      sinefit_phase: A # [A,B,C] voltage
      nominal_rms_voltage: 120 # used to scale prep to W
    current:
      sensor_indices: [0,1,2] # maps to phase A,B,C
      sensor_scales: 0.00156402587 # set by resistors and LEM
      sinefit_rotations: [240,0,120]
  filter:
    enabled: true
    filter_index: 0 # filter index 0 - 5
```

Here, the additional “filter” setting was added. After the user runs the `nilm configure` command, the relevant configuration files are created to capture and process data with the Joule data processing pipeline [52]. The Joule data processing framework, developed by John Donnal, is used to streamline the capturing and preprocessing stages. This tool models the data pipeline as a series of processing “modules,” with formally defined “streams” of information passing between them. Data flows between modules through efficient memory pipes, without needing to access the database as an intermediary. The data capture and processing has two main modules, `meter capture` and `meter process`. The `meter capture` module calls either `EthstreamCapture` or `NerdjackCapture`, for the LabJack and NerdJack, respectively. `NerdjackCapture` was added for the NerdJack as the analogous code of `EthstreamCapture` for the LabJack. The output of the `meter capture` module is the stream `/meter/sensor`. This contains the raw 8 kHz sensor data (three currents, three voltages, and the notch filter output, if configured). The `meter process` module takes as input `/meter/sensor`, and outputs three streams, `/meter/IV`, `/meter/sinefit`, and `/meter/prep`. The `meter process` module is a composite module, which runs the full filter pipeline, instead of running each process individually. This composite module runs `nilm-reconstructor`, `nilm-sinefit`, and `nilm-prep`. The `nilm-reconstructor` process takes as input `/meter/sensor` and applies the specified scale factors to output the 8 kHz currents and voltages to the IV stream. The `nilm-sinefit` process takes as input the IV stream and outputs the `sinefit` stream containing the calculated frequency, amplitude, and offset of the reference voltage stream, timestamped at the zero-crossings. The `nilm-prep` process takes as input both the IV and `sinefit` streams and outputs the `prep` stream. The `prep` stream contains spectral envelopes for the fundamental, third, fifth, and seventh order harmonics for each phase.

The data acquisition board was installed in the field as part of a NILM installation on USCGC Marlin, and collected data from August 2020 to January 2022. The notch filtered data was saved for a portion of this period. Fig. 2-16 shows ten days of data collection on USCGC Marlin, showing the real and reactive power on three phases on the left axis and the notch filtered output of one of the current streams on the

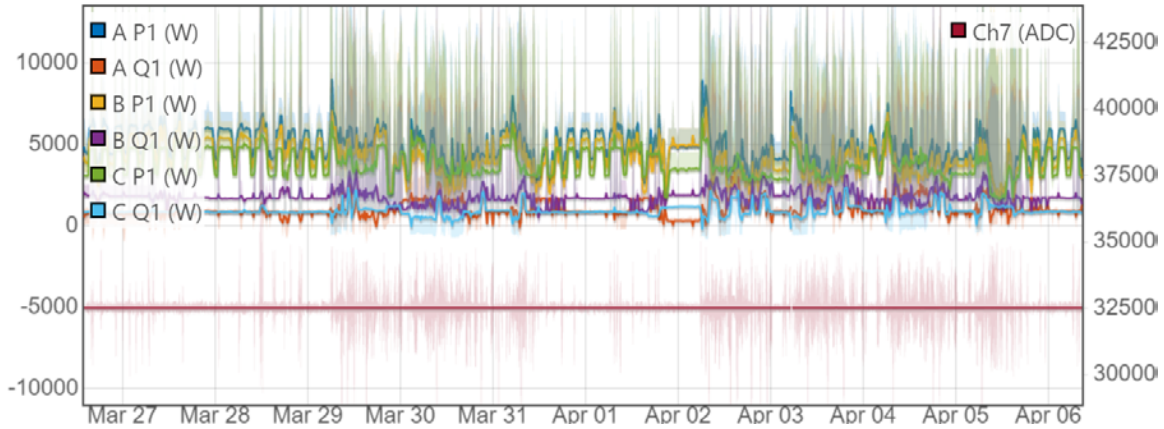


Figure 2-16: Ten days of data collection on USCGC Marlin using the described hardware. The left axis is the power streams in watts and the right axis is the filter stream in ADC counts.

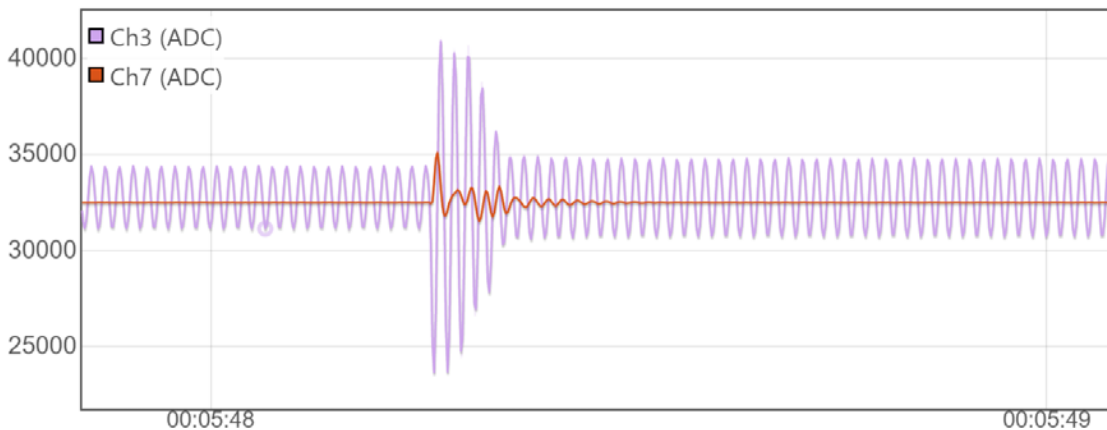


Figure 2-17: Filter performance as a load energizes on USCGC Marlin. Ch3 is the current stream and Ch7 is the filter stream.

right axis. The data is plotted with the Lumen visualization interface [53], in which the data is decimated, such that the mean value is plotted with the minima and maxima used to plot signal envelopes as a lighter shade. Fig. 2-17 shows the raw sensor data for a current stream (Ch3) and the filter response (Ch7) when a load on USCGC Marlin energizes. Fig. 2-18 shows the power stream and filter response as a different load turns on. The filter response in both these examples show that the fundamental frequency signal is quickly attenuated. The applicability of this hardware is demonstrated in Section 6.1 for diagnostics of ventilation systems.

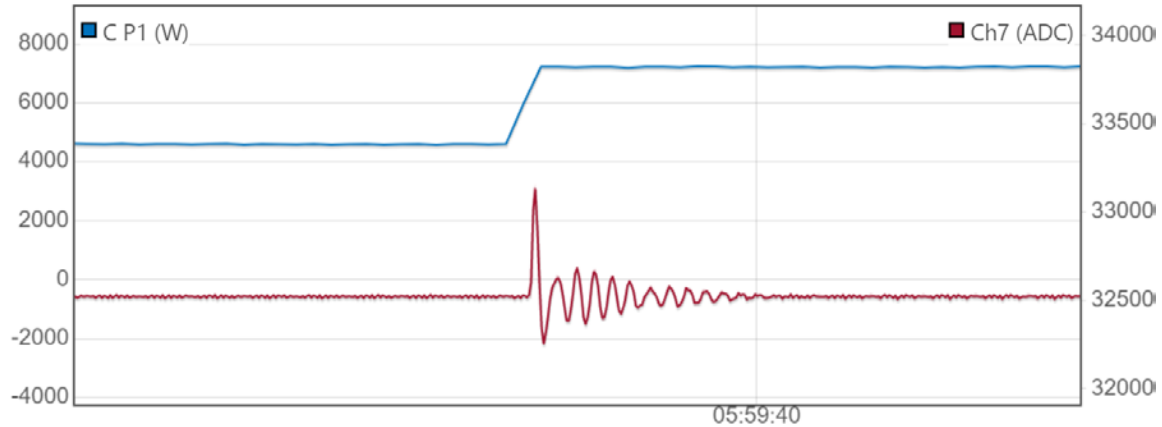


Figure 2-18: Filter performance as a load energizes on USCGC Marlin. The left axis is the power stream in watts and the right axis is the filter stream in ADC counts.

2.4 Beyond Radial Distribution Networks

Another important piece for power system measurement and preprocessing is the placement of NILM sensors in a power distribution network. All of the NILM installations described in Chapter 1 are monitoring radial distribution networks. In a radial power-distribution network, a source delivers power from the “root” of a collection of loads organized in a tree structure. Protection is relatively straightforward, typically based on calculation of available short circuit current calculated using line and system impedances. Load monitoring is also relatively easy: since power flows “out” from the source to a network of loads, one sensor on the “trunk” of the tree nonintrusively captures all power events.

Alternative distribution architectures are desirable in situations with diverse sources and a requirement for high-reliability, fault-tolerant operation. For example, the increase in distributed energy resources has increased the interest in dc ring microgrids, which provide increased reliability and efficiency by reducing power conversion stages [54–56]. Many naval vessels employ ring bus distributions to provide redundancy and protection in the event of a failure [57]. Ring distribution loops the service from a source (or several different sources) through a collection of loads and back to the source. The ring can support ac or dc distribution, and can combine sources with differing types and capacities. Interlocking rings can be separated or joined to form a

larger network or microgrid with enhanced flexibility for interconnecting power. Since a ring bus can provide power in any direction on the ring or rings, a fault can be isolated while preserving services for vital loads. The variety of power transfer paths offers significant flexibility for maintenance, fault protection, and failure response.

Path multiplicity complicates automatic relaying or protection on a multi-ring system [58–61]. Therefore, complex monitoring systems may be necessary for mission-critical ring microgrids to measure ring voltages and currents in order to identify and isolate faults [60]. Unfortunately, this instrumentation does not naturally provide power consumption information for loads or systems, as the power provided to a load may arrive over multiple paths. Additional load monitoring hardware (such as a NILM) is typically installed on service panels at the root of radial networks fed by the ring [62].

These extra sensors for load monitoring should be unnecessary. With data from a fully instrumented ring bus, it is simple in principle to apply Kirchoff’s Current Law (KCL) on the known grid architecture to determine the current flowing into radial distribution service panels further down the grid. However, the time-alignment of each meter’s data must be ensured to accurately reconstruct load power consumption. This is especially important for capturing load transients over hundredths or tenths of a second. Transient signatures serve as fingerprints that permit the recognition and disaggregation of load behavior. Power to a load delivered from multiple paths must be reconstructed from multiple measurements. Synchronization errors in data collection will distort transient shapes.

Clock synchronization is a challenging and heavily researched task for applications requiring meaningful sensor fusion, such as phasor measurement units (PMUs) [63–65]. Device-level time synchronization of PMUs typically relies on a reliable synchronizing signal like a clock derived from the Global Positioning System (GPS) [66]. In many smart meter applications it is assumed that the measurements are synchronized. However, unlike PMUs, many smart meter measurements are not actually perfectly synchronized [67]. In practice there may be significant time differences which reduces the quality of the desired output [68]. Weak signal reception makes

GPS and other wireless signals difficult to apply indoors [64]. For example, wireless synchronization solutions may be effectively impossible on small microgrids serving metal-hulled marine vessels [69, 70]. Synchronization is further complicated by drift due to temperature, EMI (electromagnetic interference), and vibration, all of which can affect the accuracy of digital clocks [65]. A wired and shared clock signal can synchronize different measurement systems, but this approach may be expensive to retrofit.

Reliable time alignment signals can be derived from the power system itself. This section explores nonintrusive load monitoring on a ring-bus microgrid similar to that found on many marine vessels. Techniques are developed for time-aligning and resolving the KCL sums in any part of the network. These techniques make it possible to nonintrusively monitor collections of loads strictly from measurements made along the ring bus. That is, signals that are likely already available for protection and relaying on the ring can be used to monitor the grid loads in detail. Demonstrations of the proposed algorithms are presented on hardware. The work in this section was in collaboration with Brian Mills and is presented in [71].

2.4.1 Ring Bus: Shipboard Examples

The monitoring methods discussed in this chapter apply to any electrical distribution system with path multiplicity. Marine microgrids serve to illustrate our hardware and algorithms. Ships use a variety of ring distribution systems to power vital loads such as fire pumps, communications, and emergency lighting. For most commercial ships, simply having multiple generators can satisfy redundancy requirements and system monitoring can be performed by meters on each generator. However, for many vessels, interconnected switchboards are required to prevent loss of power from a failure.

On a naval vessel, ring bus distribution provides protection in the event of failures of entire switchboards or machinery rooms [57]. Vital loads are fed from two different switchboards via Automatic Bus Transfers (ABTs). Because a switchboard or vital load can receive power through more than one path, power continuity and reliability is increased [72]. An example ring bus similar to that found on USCG Na-

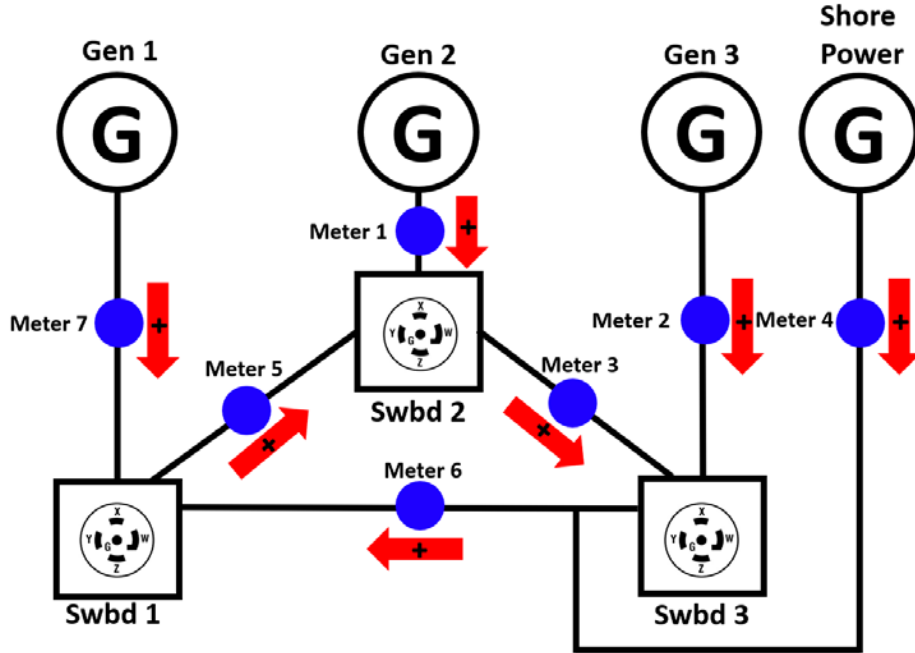


Figure 2-19: Shipboard ring bus example marked with monitoring locations [71].

tional Security Cutters is shown in Fig. 2-19. Larger ring buses can be formed. For example, the Italian aircraft carrier Cavour uses eight circularly-connected generators and switchboard installations [73]. The US Navy uses an architecture called a Zonal Electric Distribution System (ZEDS) on newer large ships. The ZEDS, a series of interconnected rings, provides additional ability to isolate different elements during a fault [74, 75]. Zonal systems, typically use both a port and starboard bus running the length of the ship, with cross-ties at multiple points that form multiple rings. Each zone of the ship has at least one load center (LC) from each bus. The load center powers a radial subnetwork of loads.

Monitoring capabilities often already exist onboard ships for oversight of the ring distribution system, but not necessarily for monitoring the status and health of individual pieces of equipment. Dual-use could be made of the monitoring equipment already in place. For example, the ZEDS onboard US Navy DDG-51s uses Multi-Function Monitors (MFMs) for automated failure response. The MFMs observe the current and voltage at each switchboard bus tie and generator breaker, and communicate via a dedicated information network with each other to execute a coordinated

shunt response to system faults [76,77]. The voltage and current data collected by one MFM is translated to events or flags and is not used directly in calculations by any other MFM, thus avoiding time-alignment challenges [78]. Adding a time-alignment capability would allow MFM hardware, or other sensors already in place, to function as a load monitor.

For demonstration, a ring bus distribution system was constructed with a delta-configured bus. Each “meter,” shown as blue circles in Fig. 2-19, consists of a data acquisition (DAQ) device fed by three LA-55 current sensors and three voltage connections, one for each phase. All seven DAQs communicate with a single computer that collates and time aligns this data to calculate power consumption for each switchboard. With meters positioned to monitor all sources of power and all paths between switchboards, the power demand of each switchboard can be uniquely determined. Treating each switchboard as a “node,” we construct the KCL matrix for the power system and sensor arrangement. Accounting for the polarity of the sensors shown in Fig. 2-19, a preprocessing matrix transforms measured ring bus currents into switchboard currents and also total current provided by all of the generators to the ship. The monitoring system performs these calculations for each phase $\psi \in \{a, b, c\}$:

$$\begin{bmatrix} i_{\text{swbd1},\psi} \\ i_{\text{swbd2},\psi} \\ i_{\text{swbd3},\psi} \\ i_{\text{ship},\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_{\text{Meter1},\psi} \\ i_{\text{Meter2},\psi} \\ i_{\text{Meter3},\psi} \\ i_{\text{Meter4},\psi} \\ i_{\text{Meter5},\psi} \\ i_{\text{Meter6},\psi} \\ i_{\text{Meter7},\psi} \end{bmatrix} \quad (2.27)$$

where $i_{\text{Meter1},\psi} \dots i_{\text{Meter7},\psi}$ are the individual line currents of Meter 1 through Meter 7 in this example. Since the power entering a node equals the power leaving that node, Eq. (2.27) can be applied for the P and Q for each phase. The matrix reflects the orientation and placement of the sensors and distribution system. If the system includes ring transformers that change voltage, appropriate entries of the matrix

can be scaled by the transformer turns ratio prior to applying KCL. A matrix can be built for both delta and wye configurations, provided all sensors are measuring entirely either delta or wye wires. In the case of a system using a delta-wye or wye-delta transformer, each measured wye line current is related to a delta phase current, which is associated with two measured delta line currents, and all measurements must be converted to a uniform wye or delta set prior to applying KCL. The same wye-delta conversion will also be applied to each meter's P and Q measurements prior to summing. For four-wire wye systems, a neutral current sensor is unnecessary, as the neutral current in a wye system is only a result of phase current imbalances and does not change the power calculations. The neutral current can be calculated by summing the individual line currents.

2.4.2 Synchronization

DAQs collecting sensor measurements operate with an internal clock for sampling data. However these clocks have limited accuracy and the actual frequency of the clock may not precisely match a specified frequency. A time offset can appear between different DAQs. This causes drift in the KCL calculations that grows over time if left uncorrected. That is, if the meter sampling rates differ slightly, there also exists a time offset error that integrates over time. This offset must be continually corrected. This complicates applying Eq. (2.27) to recreate the ship total or switchboard powers. The DAQ used for testing, a LabJack UE9, has a rated maximum clock error of ± 30 ppm [51]. Thus, the total error in one day can be up to 2.6 seconds. A test was run on the ring bus setup in Fig. 2-19 with all power sources paralleled and the ring intact (and thus all seven meters reading a portion of the power stream), in which a resistive load on Switchboard 3 was cycled multiple times at the start of the test and again 24 hours later. At the start of the test (Fig. 2-20a) all the meters are aligned within a few samples of the 60 Hz sample points; however after one day (Fig. 2-20b) the meters have drifted, the furthest two being 2.3 seconds apart.

These time offsets between meters cause errors when computing KCL with Eq. (2.27), in particular creating “artificial” transients (i.e., numerical artifacts due to

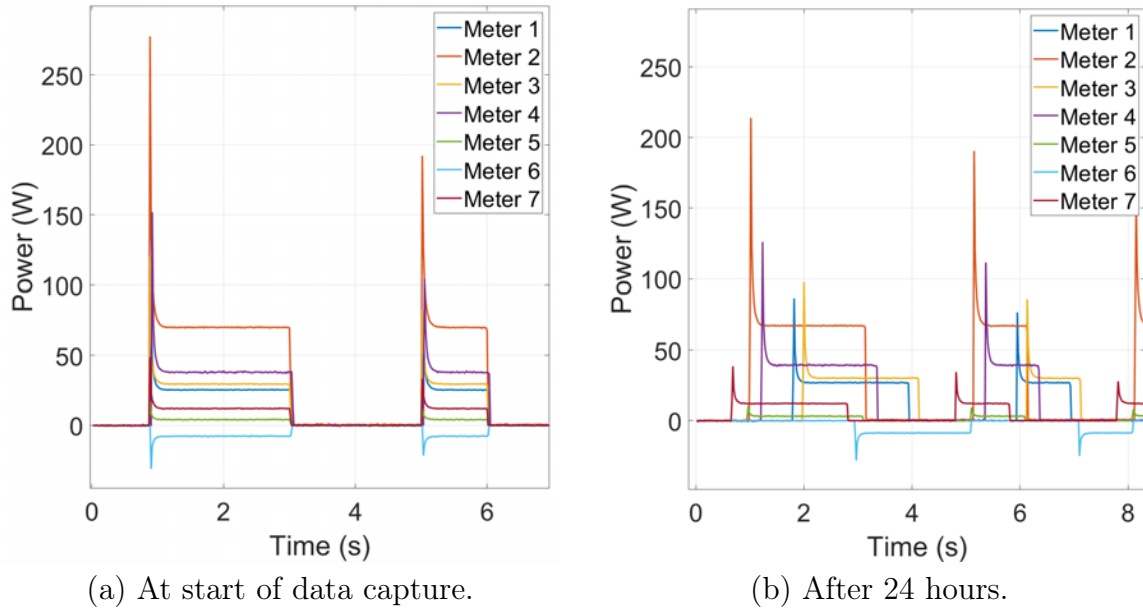


Figure 2-20: Readings from all meters at beginning of data capture and after 24 hours.

time-alignment errors that could be misinterpreted as transient events) or distorted transients in the calculated switchboard power waveforms. Artificial and distorted transients are especially pronounced during load events, e.g., when a load turns on or off, and during changes in the generator configuration. If misaligned, a load event on a switchboard will be distorted and possibly unrecognizable in the KCL recreated power stream. The same load event also results in artificial transients in the other switchboards in the system with common meters. In one demonstration, for example, power is provided to the ring from two paralleled sources, Gen 1 and Shore, while a 250 W resistive load cycles on Switchboard 1 and a constant 90 W heater operates on Switchboard 2. Fig. 2-21a and Fig. 2-21b show the reconstructed switchboard aggregate power without and with correction for the time shift, respectively. Without time-alignment, both the on and off transient on Switchboard 1 is distorted. At the same time, the time misalignment causes power to be incorrectly attributed to Switchboard 2 and Switchboard 3 when KCL is performed. In contrast, when correctly aligned, Switchboard 2 and Switchboard 3 show correct, constant steady-state values of 90 W and 0 W, respectively, and Switchboard 1 shows a clean on and off transient.

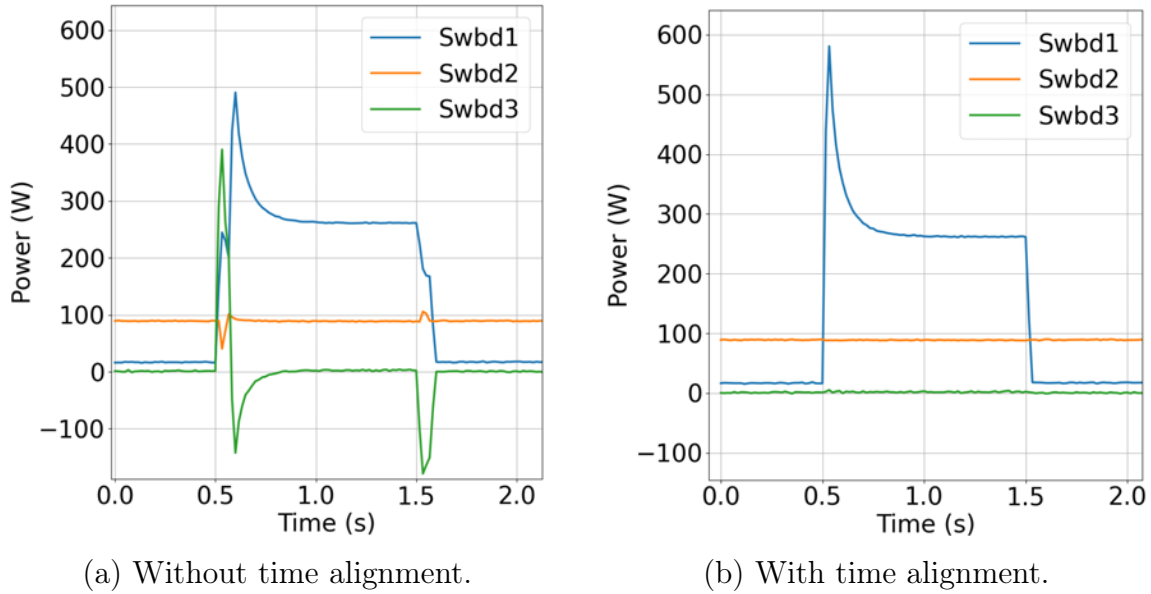


Figure 2-21: Transient distortion and artificial transients in misaligned power streams.

2.4.3 Alignment Methods

An obvious solution is to use an external signal to precisely align the DAQ meter data. Wireless alignment signals, e.g., from GPS or a local substitute, may be difficult to transmit reliably on a ship, and a wired alignment signal may not be retrofittable to existing sensors. When an external time-alignment signal is not practical, the power system itself can be used to align the data. Several different approaches can be used to “zero in” the alignment of the utility frequency cycles across the DAQs. Four methods are explored here. The first uses the voltage waveform zero-crossings to continuously track the time drifts. This method requires the initial time delay to be known. The other three methods can determine this delay, and can also be used independently, rerun periodically to realign DAQs as they drift. Methods are first explained individually, but can be used together as detailed in Section 2.4.3.5.

2.4.3.1 Zero-Crossing Delay Tracking

The voltage waveform on an ac utility is a classic source for a time reference; motorized wall clocks, for example, have been constructed for decades to take advantage of a reasonably controlled utility frequency for timekeeping. The zero-crossings of the

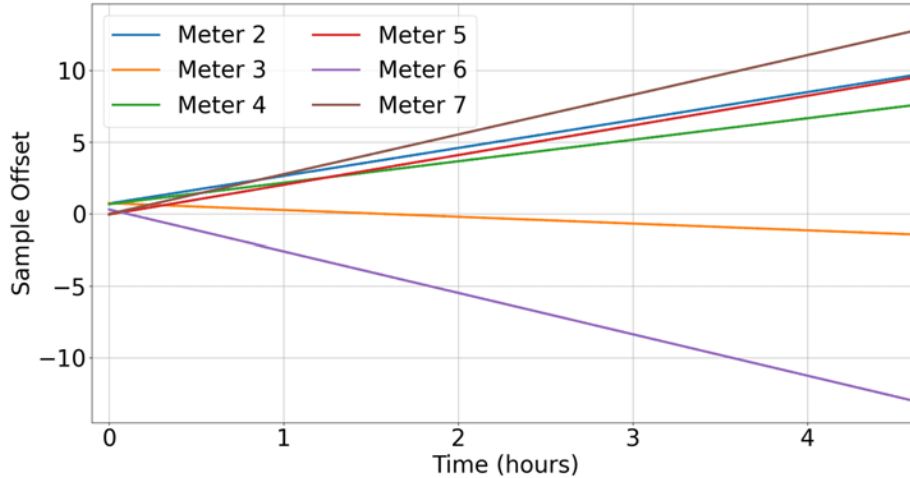


Figure 2-22: DAQ sample offset drift continuously tracked with voltage zero-crossings.

utility voltage waveform can be used to continuously track and correct the DAQ time offset. Each positive zero-crossing is marked and timestamped using the Sinefit algorithm [9], but because of individual DAQ inaccuracies, the timestamps themselves drift. Although each meter may have a slightly different sample rate, the zero-crossing event occurs consistently across all meters, making this an ideal “mile marker” for aligning the data. By simply maintaining a count of the zero-crossings, the time offsets between meters can be determined and the power streams combined perfectly. Fig. 2-22 shows the calculated DAQ sample offset drift of each of the meters in the testbed power system relative to Meter 1. These drifts can be corrected by updating the timestamps.

The primary obstacle to using voltage zero-crossings for DAQ alignment arises from the ambiguity in the precise 60 Hz cycle being observed by a DAQ. We require an approach, analogous to a “home pulse” on a rotary encoder, that establishes the initial time delays between DAQs for correct relative counting of the different observed DAQ line cycles. There are many possible approaches for establishing the initial alignment for zero-crossings.

2.4.3.2 Voltage Correlation Alignment

Regardless of power flow through each meter, the voltage waveform is universally visible by all of the DAQs. Even a healthy electrical grid will not maintain a perfectly “stiff” voltage at all times, resulting in variations in both voltage amplitude and frequency. Microgrids are particularly susceptible to disturbances. These variations can be used as serendipitous events and patterns that provide alignment markers. The voltage frequency and amplitude are automatically calculated by the Sinefit algorithm [9], which uses successive 4-parameter sine wave fitting to fit an observed voltage to the functional form:

$$v[n] = A \cdot \sin \left(2\pi n \frac{f_0}{f_s} + \phi_0 \right) + C, \quad (2.28)$$

where A is the amplitude, f_0 is the line frequency, f_s is the sampling frequency, ϕ_0 is the phase angle, and C is the offset. For example, Fig. 2-23 shows the voltage frequency and amplitude data collected by two independent meters installed onboard USCGC Spencer, monitoring two separate subpanels (port and starboard). This data is taken while the ship is underway and the ship’s grid is being powered by a ship service diesel generator. Because of the limited generation capacity and finite inertia of a shipboard microgrid, there are frequently small fluctuations in supply frequency. A readily apparent time offset exists between the sample streams of these two meters.

The ring bus demonstrator used for our laboratory testing draws power from a terrestrial grid, operating with a relatively stiff voltage waveform, but even still, there are small voltage amplitude and frequency fluctuations as a result of changing loads on the grid. These “events” in voltage frequency and amplitude can be used to time-align the DAQ data. Fig. 2-24 shows the voltage frequency for two meters on the ring bus demonstrator; the pattern can be matched by cross-correlation. The time window chosen for cross-correlation needs to be large enough for these patterns to become visible. For a “stiffer” grid, the noise (shown in Fig. 2-24b) in the frequency stream may dominate over smaller time windows and cause a loss of precision. Alignment of the ring bus demonstrator using the grid was able to identify all meter delays to

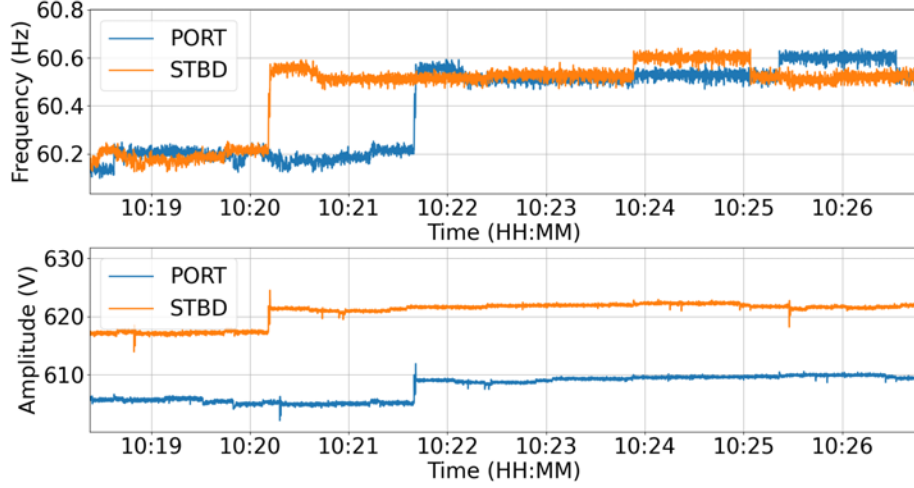


Figure 2-23: Voltage frequency and amplitude from USCGC Spencer while underway.

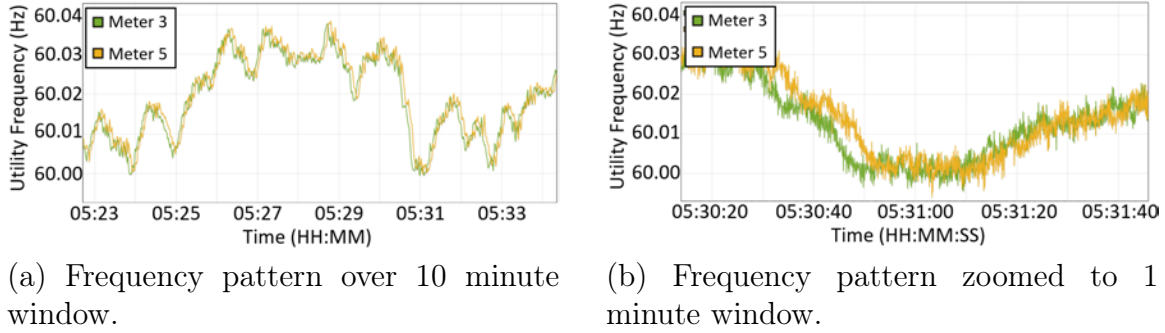


Figure 2-24: Patterns visible in utility voltage frequency used for time alignment.

within 1 sample of the correct delay value. This precision may vary for different grids, thus this method should be tested on the specific utility voltage waveform prior to use.

For a signal $s[n]$ observed by at least two DAQs, e.g., a utility voltage waveform, the measurement from the two meters $x[n]$ and $y[n]$ can be described as,

$$\begin{aligned}
 x[n] &= s[n] + n_1[n] \\
 y[n] &= s[n - D] + n_2[n] \quad 1 \leq n \leq N
 \end{aligned}
 \tag{2.29}$$

where D is the unknown delay, $n_1[n]$ and $n_2[n]$ are the noise of the two meters (assumed to be uncorrelated), and N is the number of samples. To time-align these

meters, or a larger collection of DAQ meters, one stream is chosen as the base stream $x[n]$ and the delay D must be found for the remaining six meters with respect to $x[n]$. The estimated delay \hat{D} is where the cross correlation of $x[n]$ and $y[n]$ is maximized [79]:

$$\hat{D} = \max_{\tau} R_{xy}(\tau), \quad R_{xy}(\tau) = \sum_{n=1}^N x[n]y[n + \tau], \quad (2.30)$$

where τ is the time shift applied to the shifting meter $y[n]$ during the correlation process and ranges across the correlation window ($\tau = 1 \dots N$).

2.4.3.3 Adjacent Power Event Alignment

Observed load on and off events in the power stream can be used for time-alignment. Transient events in the power stream are generally much larger and easier to detect than voltage variations. This approach requires identifying events in different meter power streams that are a result of the same physical event. Power flow is not uniformly or consistently distributed across all meters because different bus configurations shift the direction and amounts of power flowing around the ring bus. Thus, because it is not guaranteed that each meter is seeing all of the same events and because the power seen by each meter is not the same, the correlation-based method for voltage correlation alignment cannot be used here. Instead, an edge detector or change-of-mean detector is used to find the locations of on and off events in each power stream, and corresponding events between each meter are used to calculate the needed time-alignment shift. To determine corresponding events between meters, an assumption needs to be made about the maximum time shift that can occur between two meters. For example, if alignment is conducted immediately after the DAQs are activated, the maximum drift is bounded, e.g., to 10 samples, or $1/6^{\text{th}}$ of a second, based on the known DAQ hardware drift.

Using an edge detector to find the locations of all transient peaks in a given time window, a meter has a set of detected events $E = \{E_1 \dots E_M\}$ with corresponding sample indices $\{m_1 \dots m_M\}$ and a second meter has a set of detected events $F = \{F_1 \dots F_N\}$ with corresponding sample indices $\{n_1 \dots n_N\}$. The goal is to find the

intersection of events, $E \cap F$. For each event in E with index i the distance to the closest event in F is calculated:

$$d_i = \min(m_i - n_j), \quad j = 0 \dots N. \quad (2.31)$$

The delay is estimated as the mean distance \bar{d} using all $d_i \leq B$, where B is the bounded maximum drift (10 samples in our testing):

$$\hat{D} = \text{mean}(d_i), \quad \forall i \text{ where } |d_i| \leq B. \quad (2.32)$$

It is assumed that when $d > B$ the event in E has no corresponding event in F , and that event is discarded. To minimize this discarded data, the meter with the maximum number of events provides the reference set of events E .

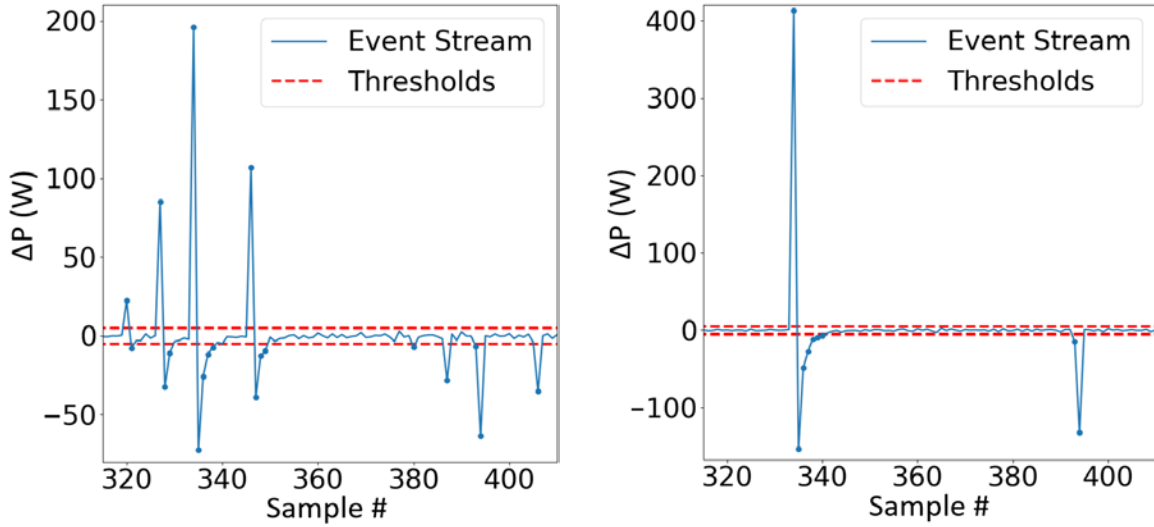
2.4.3.4 System Power Reconstruction

Alignment can also be performed by focusing on the end goal: no “artificial transients” in the calculated switchboard power stream. One meter is chosen as the reference meter, and data from the other meters surrounding that switchboard are shifted in time across a range of delay values τ . For each combination of delays the estimated switchboard power stream is calculated by combining the meter power streams, using the relevant part of Eq. (2.27). Taking each recreated switchboard power stream P , a first order difference filter is applied to generate an event stream:

$$\Delta P[n] = P[n] - P[n - 1]. \quad (2.33)$$

If the meters are aligned, individual physical events seen by each meter will overlap in the reconstructed switchboard event stream, showing as a “single” event. However, if the meters are not aligned, individual physical events from each meter will be distanced from each other, appearing as “multiple” events in the switchboard event stream.

The four meters surrounding Switchboard 3 are used as an example here. A



(a) Incorrect delay values (τ); all four meter events misaligned.

(b) Correct delay values (τ); all four meter events aligned.

Figure 2-25: Reconstructed switchboard event stream used for alignment by minimizing data points outside threshold.

resistive load is cycled on and off, which results in a large positive peak at turn on followed by a large negative peak as inrush current fades, and a negative peak at the turn-off event. Fig. 2-25a shows the event stream when the four meters of Switchboard 3 are not aligned. The four meters all see the same event, but summing the misaligned streams creates four separate peaks for the on-event and four separate peaks for the off event in the event stream. In contrast, Fig. 2-25b shows that the time-aligned meters results in a single on-event and off-event. The event stream with the correct alignment will have the fewest event peaks. Choosing an appropriate threshold to filter out the noise (5 W in Fig. 2-25), the number of samples that fall outside this threshold is counted. The delay values yielding the minimum number of samples outside the threshold yields the correct alignment.

For example, for a three-meter system, the reference stream is given as $w[n]$ and streams $x[n]$ and $y[n]$ have unknown delays D_x and D_y . Then, C is calculated as the number of event stream samples outside the threshold γ for a range of delay values.

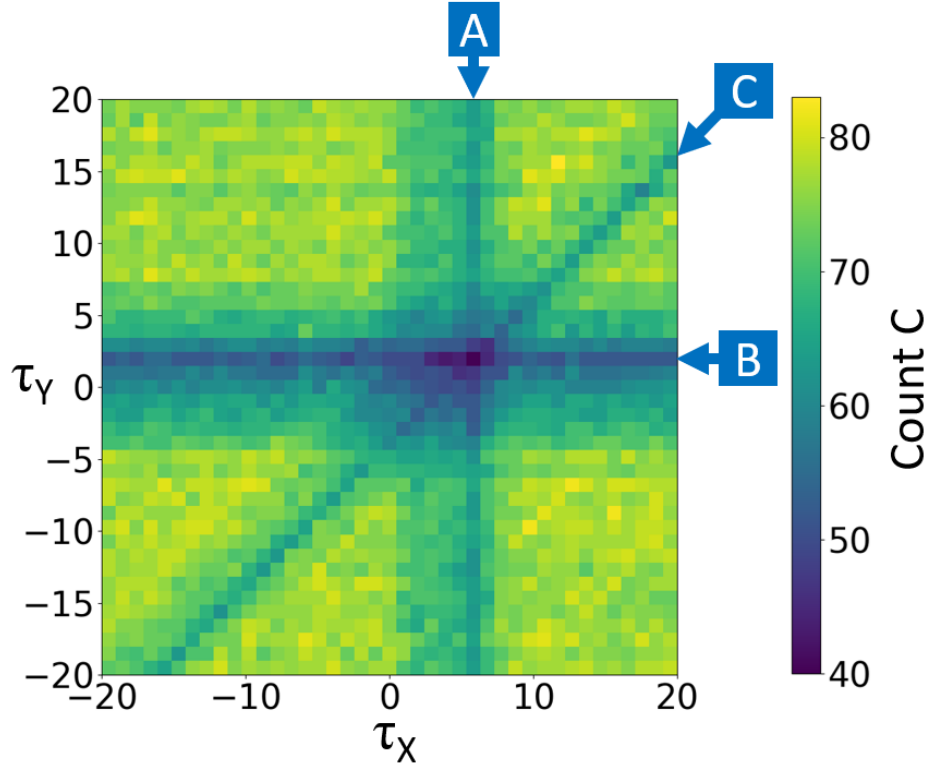


Figure 2-26: Count of event stream samples outside threshold for two meters shifted by time delays (τ_x, τ_y) . Minimum lines appear in the color plot where each meter aligns with the base meter (A and B), as well as with each other (C).

The estimated delays \hat{D}_x and \hat{D}_y are the delay values (τ_x, τ_y) that minimize C :

$$\hat{D}_x, \hat{D}_y = \min_{(\tau_x, \tau_y)} C, \quad C = \sum_n \begin{cases} 1, & |\Delta P_{(\tau_x, \tau_y)}[n]| > \gamma \\ 0, & |\Delta P_{(\tau_x, \tau_y)}[n]| \leq \gamma \end{cases}. \quad (2.34)$$

The values of C for a three-meter alignment over a 40 sample delay window is shown as a color plot in Fig. 2-26. Each axis represents the delay applied to each meter, shown as the number of samples one meter is shifted. The color plot makes clearly identifiable lines appear where two of the three meters align. In Fig. 2-26, A shows the minimum line where the x meter aligns with the w meter, B shows where the y meter aligns with the w meter, and C shows where the x meter aligns with the y meter. These three lines all cross at the absolute minimum point where all three meters align, which in this example is the point $(\tau_x, \tau_y) = (6, 2)$. This tells us that

the x-axis meter is offset by 6 samples and the y-axis meter is offset by 2 samples.

This method is not limited to aligning three meters; additional meters simply add more dimensions. The largest number of meters aligned in our lab testing was four, as both Switchboard 3 current/power and total ship current/power calculations require up to four meters to reconstruct. The computation increases as larger windows and more meters are aligned as $O(m^{n-1})$, where m is the range of time delays tested and n is the number of meters being aligned. Distribution systems with more meters involved in reconstruction calculations will require drastically more computation. While this method will work for large offsets, it is preferable to use this method in smaller windows where meters are already known to be in close alignment to keep m low. Further research could optimize computations by searching for local minimums along several rows and columns, finding these partial alignment lines to point towards the complete alignment at the global minimum.

Once the meters surrounding each switchboard have been aligned with their neighbors, each switchboard can be aligned with the other switchboards by using common meters. The meters on the bus ties between switchboards are used in both switchboard alignment calculations, and serve as intermediaries to calculating the total alignment of the system. Each bus tie meter reveals the delay of one switchboard relative to another switchboard, allowing all meters to be adjusted to align with one base meter. Alternatively, inter-switchboard alignment can be found by performing this method using the meters of all online generators, reconstructing the total ship power instead of switchboard power. This will align all three switchboards only if all three generators are paralleled.

2.4.3.5 Method Choice

Each of these time-alignment methods have strengths and weaknesses when aligning several meters, and can be combined to achieve greater certainty. The zero-crossing method can perfectly track the time-drift, but relies on knowing the initial offset, which must be found through the other methods. The voltage correlation alignment method can be used to determine the offsets even in the absence of load transients.

Voltage correlation alignment can also be used to quickly recover from large offsets, and can be run periodically to maintain alignment throughout changing generator and bus configurations.

When load transient observations are available, the transient or power-based methods can be used. Adjacent power event alignment relies on having a known small offset, otherwise physical events can no longer be correlated between different meters. When the offset is bounded, from either zero-crossing tracking or voltage correlation alignment, the power stream events can be used for finding the exact cycle-count integer offset. Another way to keep the offset small is to periodically restart the data capture. The system power reconstruction can recover from large offsets, but computation time increases as the alignment window grows.

Particularly when used in combination, these methods provide strong data integrity for alignment. Prolonged periods without power transients reduce opportunities for employing some of the methods, but precise alignment of meters is only necessary when transients occur, as performing KCL using steady-state levels will not produce “artificial” transients.

2.4.4 Experimental Verification

To demonstrate the monitoring capability of switchboards fed by a ring bus, a ring bus demonstrator was built and tested (see Fig. 2-27) following the USCG National Security Cutter schematic of Fig. 2-19. The test system emulates three ship service generators and a shore power connection, splitting incoming three-phase power into four different paths [80]. Relays serve as generator ties and breakers on the ring bus. A collection of L22-30 receptacles represent the switchboards, capable of providing power to load centers.

The time-correction methods were tested on the shipboard ring bus demonstrator. A four minute test was run with a three-phase variable-speed motor operating on Switchboard 1, a 250 W resistive load and an axial fan on Switchboard 2, and a 250 W resistive load and a centrifugal fan on Switchboard 3. Various generator configurations were used to simulate different ship configurations. The various generator

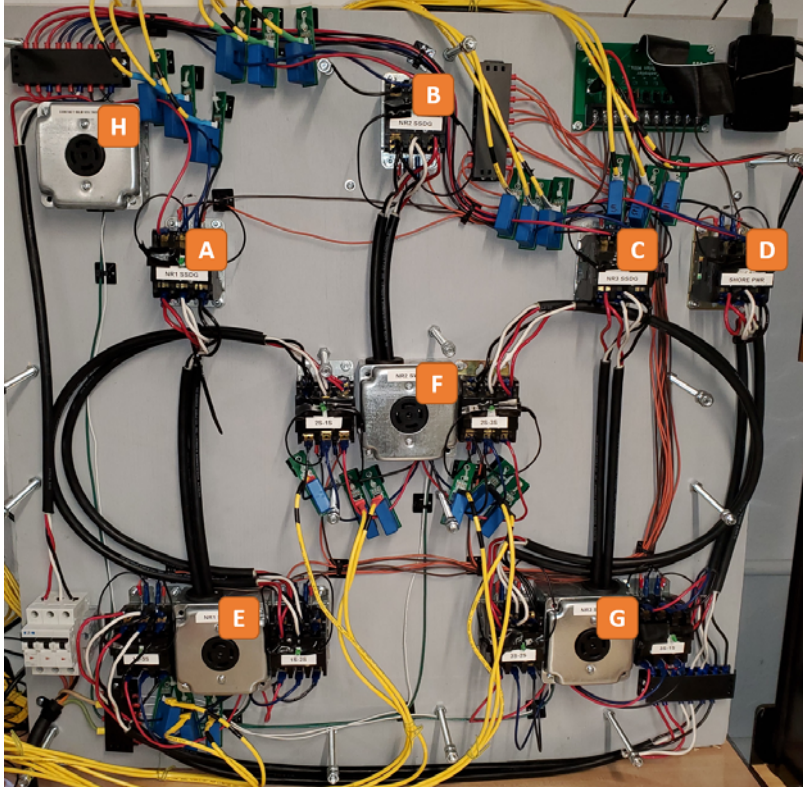
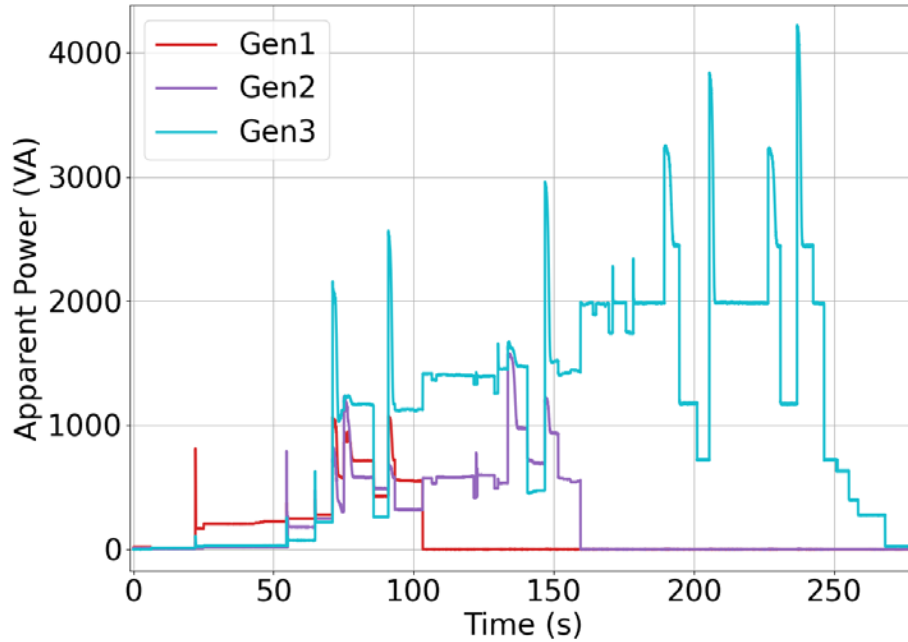


Figure 2-27: Shipboard ring bus demonstrator. Incoming power (H) splits into four power sources, Gen 1 (A), Gen 2 (B), Gen 3 (C), and Shore Power (D), which feed three switchboards, Swbd 1 (E), Swbd 2 (F), and Swbd 3 (G). Current measurements are from the blue LA-55 current sensors.

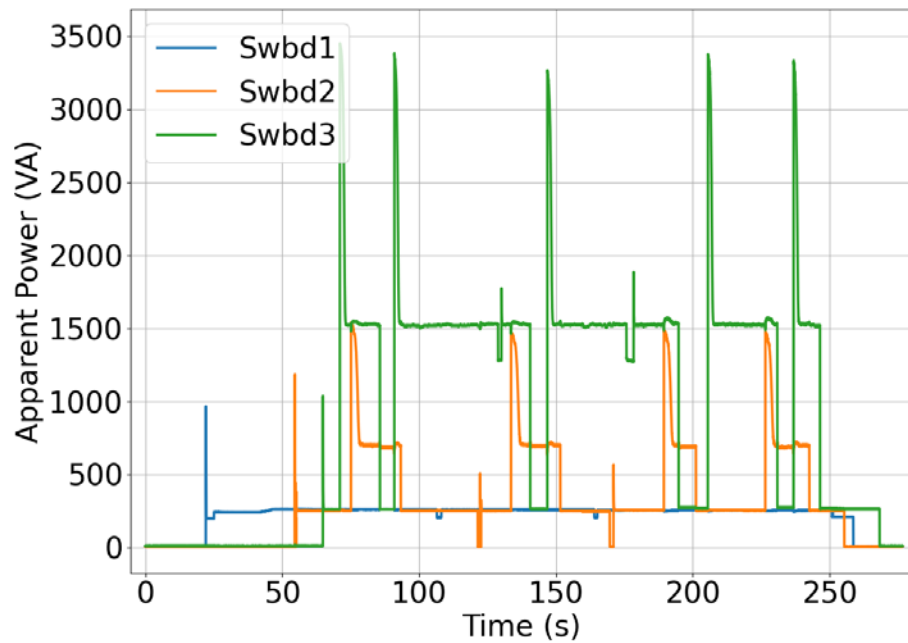
Table 2.1: Line cycle sample offsets.

	Meter2	Meter3	Meter4	Meter5	Meter6	Meter7
APE	2	0	N/A	4	-3	4
SPR	2	0	N/A	4	-3	4
VCF	2	0	2	4	-3	4
VCA	2	0	3	4	-3	4

configurations are shown in Fig. 2-28a. For convenient visual display, the power is plotted as the total apparent power provided by each generator tie. DAQ time delays were calculated using the adjacent power event (APE) method, system power reconstruction (SPR) method, and voltage correlation amplitude (VCA) and frequency (VCF) methods. Using Meter 1 as the reference stream, the relative line cycle offsets for Meter 2 through Meter 7 are shown in Table 2.1. All of these methods calculated the same offsets for all active meters. Shore power (Meter 4) was not paralleled dur-



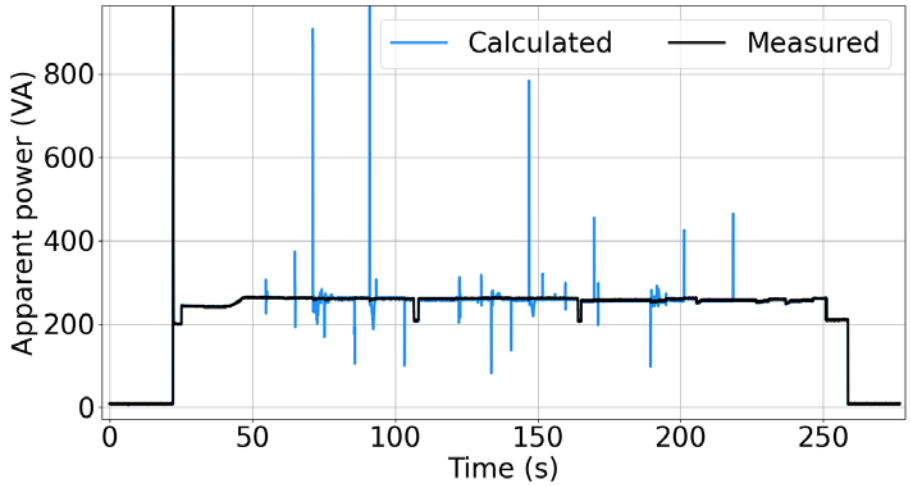
(a) Generator power streams.



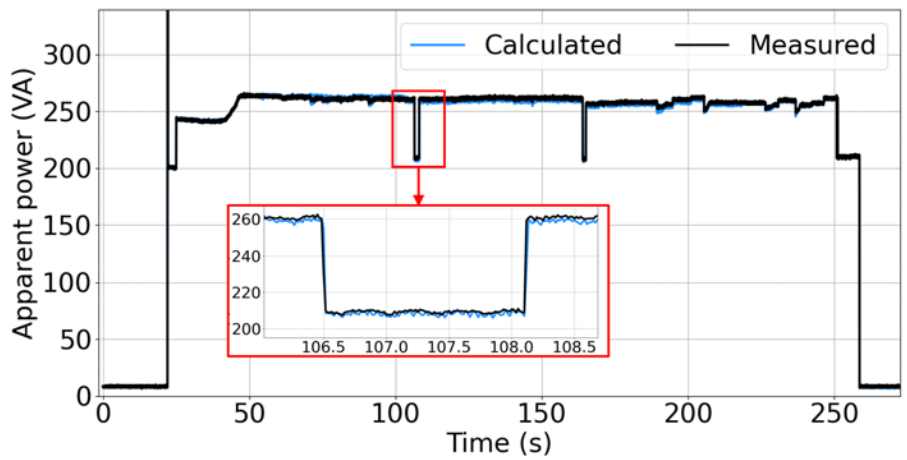
(b) Calculated switchboard streams.

Figure 2-28: Generator streams and calculated switchboard power streams.

ing this test, thus the methods that rely on load transients in the power stream (i.e., adjacent power event alignment and system power reconstruction) did not identify an offset value for Meter 4. However, the meter still recorded system voltage, allowing



(a) Without time alignment.

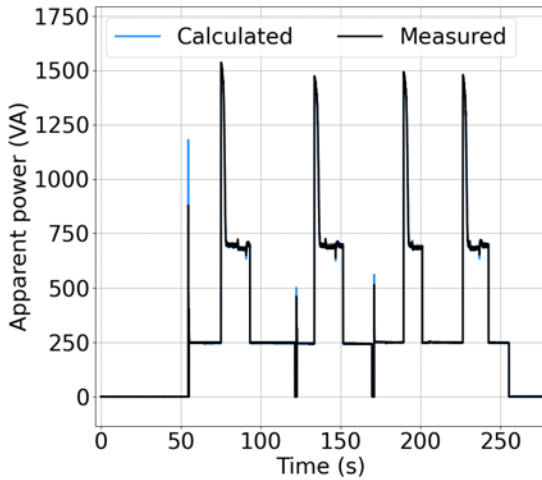


(b) With time alignment, with a zoomed-in view of a region.

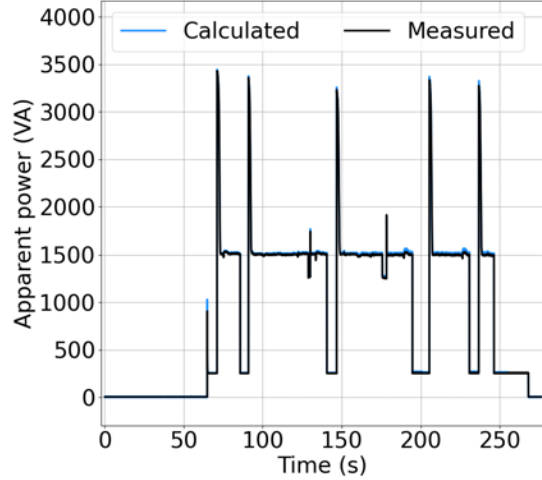
Figure 2-29: Calculated power of Switchboard 1 compared to measured.

the offsets to be determined using voltage correlation alignment. Voltage amplitude and frequency alignment methods calculated a 2 sample and 3 sample offset, respectively, for Meter 4. Since Meter 4 observed no power flow and no load transients, the offset for Meter 4 does not affect switchboard reconstruction. The Meter 4 offset would be correctly computed when load events are present.

Fig. 2-28b shows the calculated power for each of the three switchboards using the calculated sample offsets. For this test, an additional NILM meter was installed directly upstream of each switchboard so that the KCL-calculated power can be compared and cross-validated to directly measured switchboard power. Fig. 2-29a and



(a) Switchboard 2.

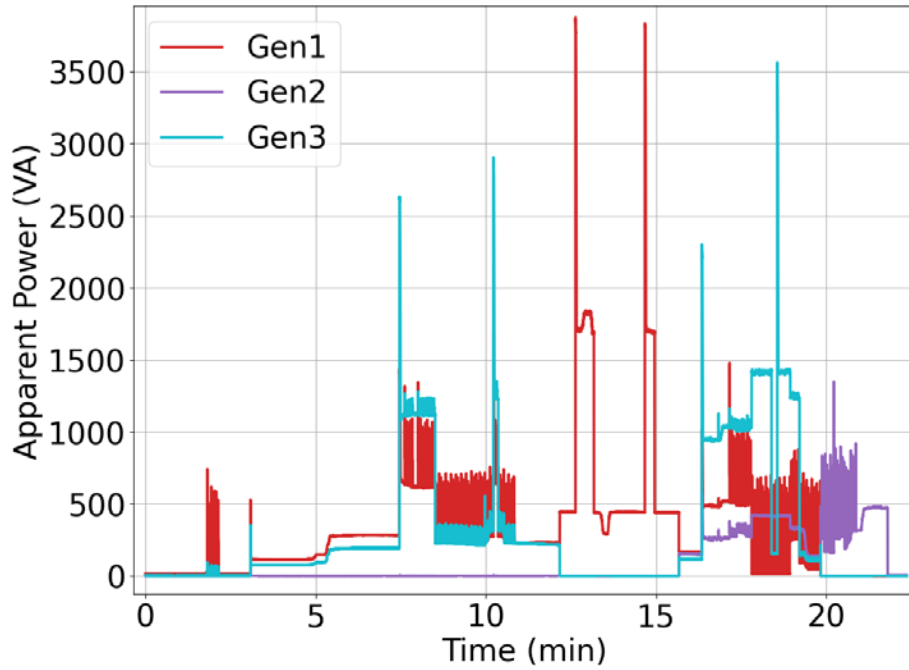


(b) Switchboard 3.

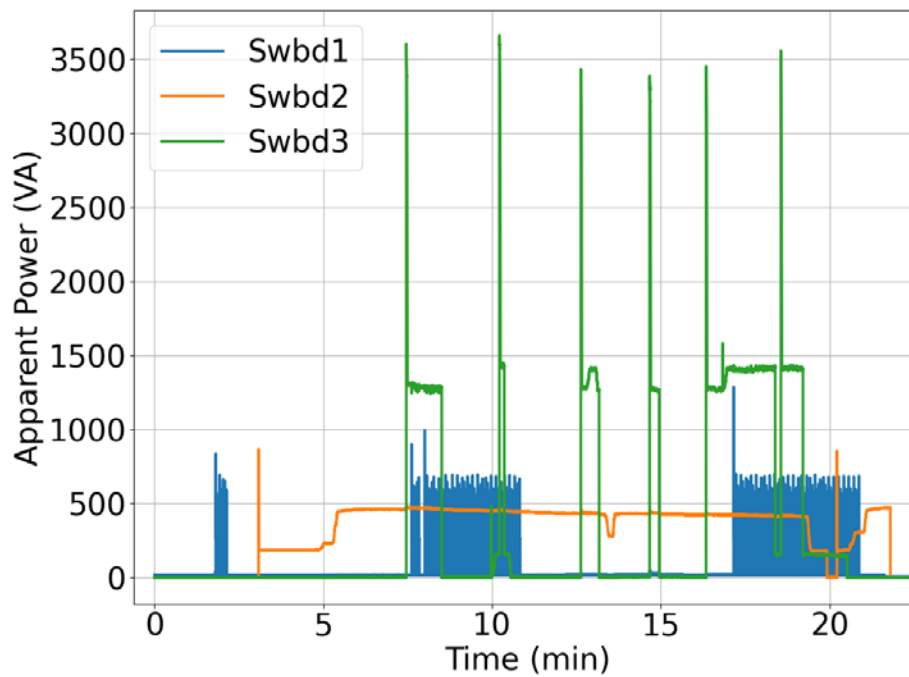
Figure 2-30: Calculated power of Switchboards 2 and 3 compared to measured.

Fig. 2-29b show the calculated power for Switchboard 1 compared to the measured power without and with time-alignment, respectively. For the non-aligned scenario in Fig. 2-29a, there are many “artificial transients.” These are a result of sample offsets between the meters causing power from Switchboard 2 and Switchboard 3 to be incorrectly attributed to Switchboard 1 when KCL is performed. In contrast, the calculated waveform in Fig. 2-29b matches closely with the measured power. Fig. 2-30a and Fig. 2-30b show the calculated power with time-alignment compared to the measured power for Switchboards 2 and 3, respectively. The root mean squared error (RMSE) calculation errors in apparent power for Switchboards 1, 2, and 3 are 3.99 VA, 10.10 VA, and 25.37 VA, respectively. Without time-alignment, the RMSE for the three switchboards are 28.44 VA, 32.74 VA, and 37.37 VA, respectively.

Next a longer 25 minute test was conducted to simulate multiple generator configurations and cycling loads on all switchboards. The generator power streams and recreated switchboard power streams are shown in Fig. 2-31. On Switchboard 1 there is a 250W light cycling in a one second on, two second off pattern. On Switchboard 2 there is a three-phase variable-speed motor. On Switchboard 3 there is a 1 hp centrifugal fan and a 150 W variable speed axial fan. Fig. 2-32 shows the generator streams and estimated switchboard streams as the configuration goes from three to

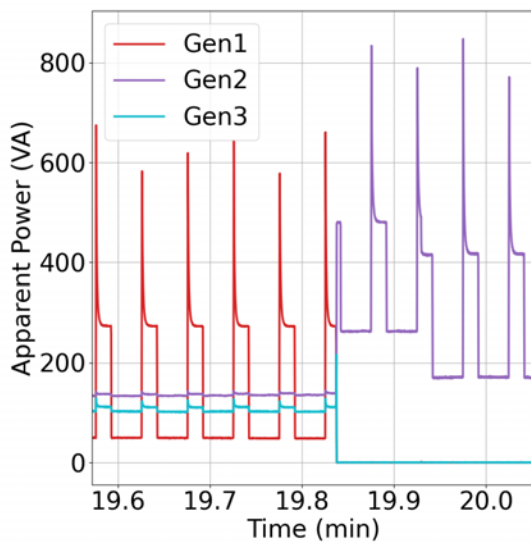


(a) Power source streams.

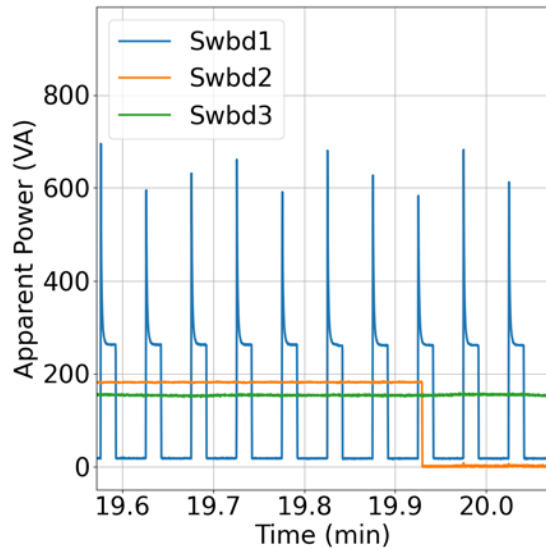


(b) Calculated switchboard streams.

Figure 2-31: Generator streams and calculated switchboard power streams for 25 minute test.

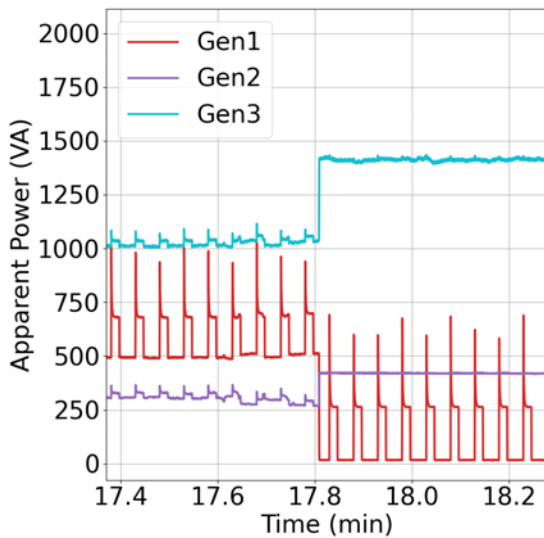


(a) Power source streams.

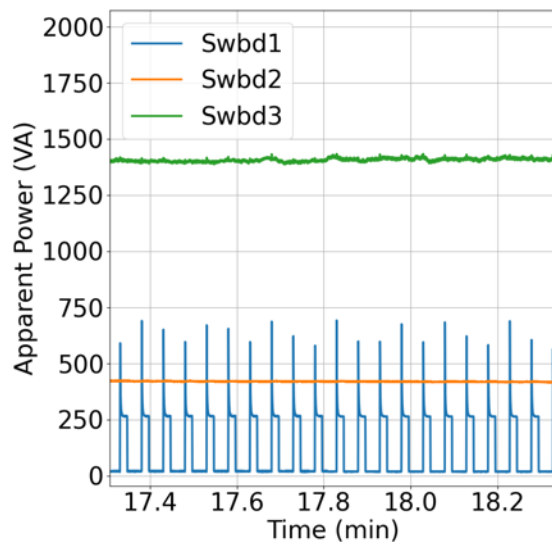


(b) Estimated switchboard streams.

Figure 2-32: Power streams for scenario in which Gen1 and Gen3 are brought offline (at about 19.84 minutes).



(a) Power source streams.



(b) Estimated switchboard streams.

Figure 2-33: Power streams for scenario in which generators are put into split plant configuration (at about 17.8 minutes).

one generators. Fig. 2-33 shows the generator streams and estimated switchboard streams as the configuration transitions from three parallel generators to the split-plant configuration. In split plant configuration, each generator supplies power to one

switchboard. These tests demonstrate the ability to accurately recreate the switchboard power streams during generation re-configurations.

2.5 Chapter Summary

This chapter presented several advancements for power system measurement and preprocessing. First, the chapter presented an analysis of spectral envelope preprocessing in the presence of prequantization noise. Spectral envelope preprocessing can enhance system resolution beyond that of the input quantizer. It was shown that prequantization noise can be either a benefit or a drawback depending on the amount introduced. A lack of prequantization noise will result in a nonlinear preprocessor transfer characteristic. An excessive amount of prequantization noise will cause preprocessor systems with different numbers of input quantization bits to operate at similarly poor effective resolutions. By adapting the concept of effective number of bits to the spectral envelope preprocessing system, this effect can be evaluated quantitatively for a proposed preprocessor design given the number of quantizer bits, number of points in the averaging window, and signal-to-noise ratio. The ratio of minimum resolvable power to maximum resolvable power can be derived for the preprocessor configuration to describe the system's ability to resolve small loads.

Next, new data acquisition hardware was presented, custom tailored to the needs of nonintrusive power monitoring on the ac utility. This hardware can automatically track and reject utility content in a nonintrusive current measurement. This hardware was integrated with the NILM software suite and installed on USCGC Marlin for slot harmonic tracking and speed estimation.

Finally, the chapter presented signal processing techniques for time-alignment of nonintrusive power measurements on ring distribution networks. Perhaps surprisingly given its familiarity, performing a practical KCL calculation is a challenge that requires precise time-alignment of sensor readings to avoid both distorted and artificial transients. Experimental results from the ring bus test bed demonstrate the feasibility and capability of a power monitoring system to track complex power dis-

tribution systems and loads without external time-alignment signals. Sources of potential time-alignment errors were identified and analyzed. The applicability of the time-alignment methods depend on several factors, such as if the initial time offset is known, the bounds of the time offset, the “stiffness” of the power system, and the occurrence of load transients. Evaluation showed that the combination of multiple time-alignment methods can provide greater data integrity. These time-alignment methods allow dual use of existing grid sensors or use of unmodified independent DAQs to feed data to a computer, which then calculates power flows around the grid and identifies loads via their on and off transients. The proposed monitoring system adds value to the management of energy resources and grid monitoring, collecting data for condition-based maintenance, fault detection, and failure response.

Chapter 3

Physics-Informed Feature Extraction and Selection

After preprocessing of measured currents and voltages into power spectral envelopes, the relevant features for load identification and diagnostics need to be extracted and selected. This chapter presents a multi-scale framework for organizing the signal processing for nonintrusive load monitoring. The framework separates out loads that operate on different time-scales. Then, feature space variability examples are presented using laboratory demonstrations and field data. Finally, a method is presented for evaluating load separability in a feature space prior to the application of a pattern classifier.

3.1 Multi-scale Framework

In an electrical system there are four main behaviors of loads: *On/Off*, *Finite State Machine (FSM)*, *continuously variable*, and *continuously on* [81]. Both *On/Off* loads and *FSM* loads have sequences of changes of state that are discrete changes in power, and consume an approximately constant amount of power at each state or operating mode. An *On/Off* load has only two states, on or off, while a *FSM* load has several operating states due to its complex operation. The load transient, i.e., when the power quickly changes when transitioning between states, acts as a load signature allowing

NILM algorithms to identify the state of the loads. A *continuously on* load can be on for days, or even weeks at a time. These loads can be identified if the initial on event is observed or with the use of optimization techniques [82]. Finally, a *continuously variable* load has a continuously changing power demand, and does not have a fixed number of states for the entirety of load operation. Accordingly, it is much more difficult to identify and track the operation of variable power loads with the same techniques used for loads that can be fully characterized with discrete levels. Loads can have variable power due to dynamic loading conditions. For example, machines like routers and mills have changing power demand based on cutting conditions, such as workpiece material and depth of cut [83]. Loads can also have changing power demand due to control by power electronics, including variable frequency drives (VFDs) and dimmable lighting. The use of power electronics enables many loads to operate over a wide variable power range; thus there may not be a unique power consumption pattern.

The multi-scale framework presented in this section is presented in [6]. The framework is focused on the fundamental physical features of the energy consumption of each load. Do transients vary over time? Are they randomly distributed or repeatable? Do they present harmonic content to the utility? These features and others form the foundation for nonintrusive load identification. The framework is structured around data acquisition, scale, and variability of power consumption. First, time-series electrical signals are collected that give insight into the physics of load behavior. Second, scales of events are examined, since events can occur on different time-scales. Finally, transients and steady-state power consumption can be variable, ranging in almost deterministic in their predictability to practically stochastic, which leads to differing approaches for load disaggregation. The framework is shown in Fig. 3-1. The main stages of the identification process are labelled, and include:

1. **Data Acquisition and Preprocessing** extracts the physical characteristics of energy consumption.
2. **Event Scale Separation** determines the scales for events.

3. **Event Detection** determines load “signatures” for detected events.
4. **Event Mapping** matches events to loads.
5. **Event Confirmation** checks constraints between load events.

This section gives a high-level overview of event scale separation, event detection, event mapping, and event confirmation. In-depth examples of the application of these techniques are presented in Chapter 4, Chapter 5, and Chapter 6.

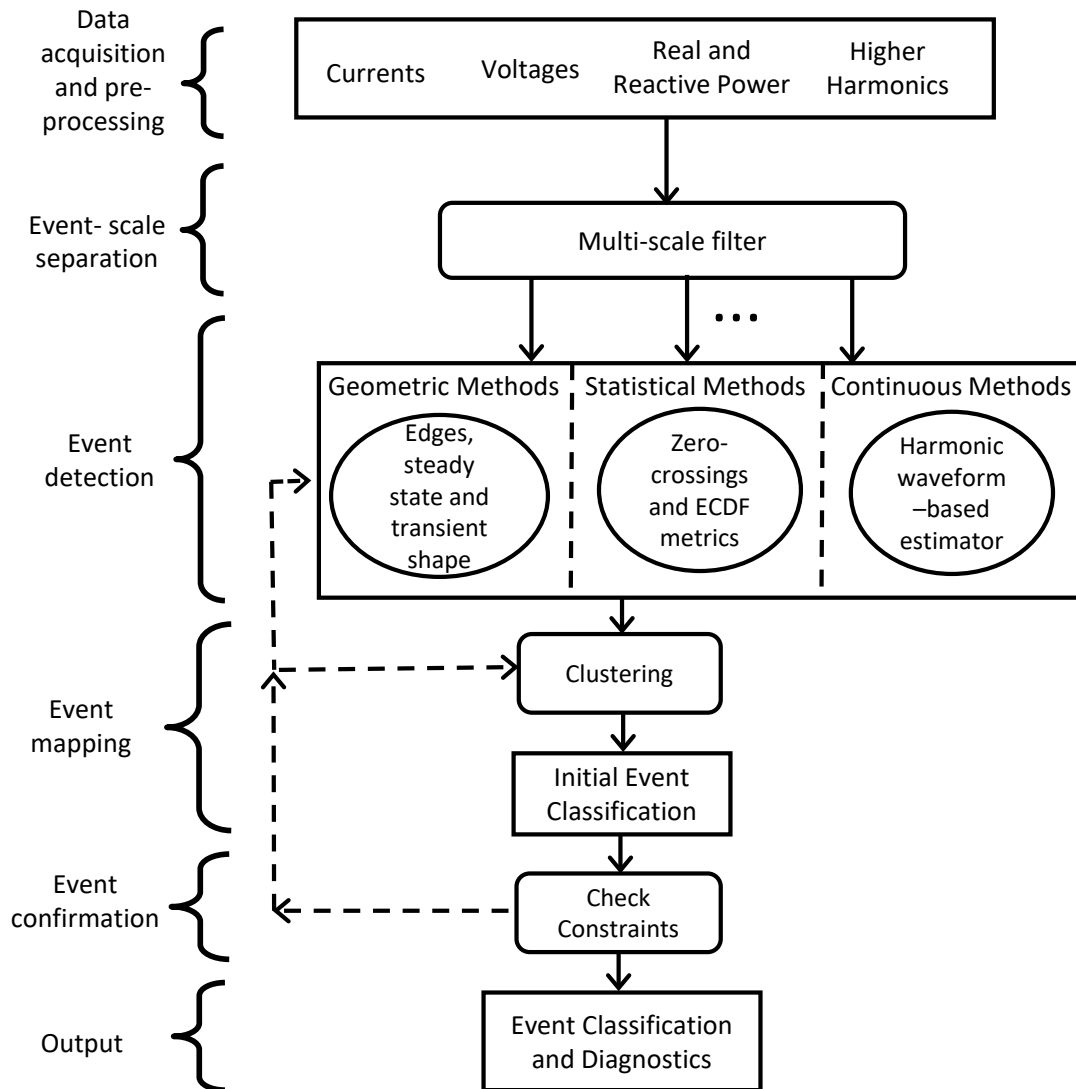


Figure 3-1: Diagram for multi-scale, multi-algorithmic framework.

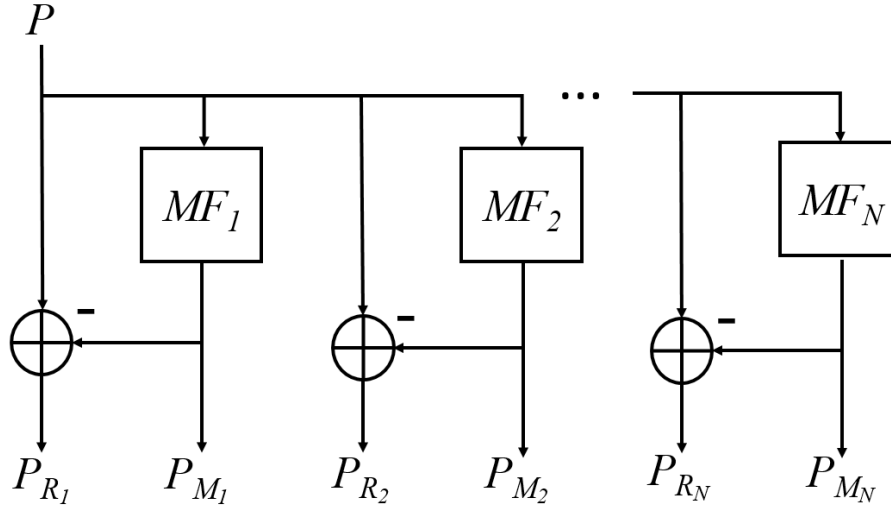


Figure 3-2: Diagram for separating short-scale versus long-scale loads, for N different length median filters, MF . P_R is the residual stream and P_M is the median filtered stream.

3.1.1 Event-scale Separation

Load events can occur on different time-scales. To detect events on different time-scales, a rolling median filter is used on the power stream. A median filter eliminates small fluctuations while preserving sharp edges [84]. Thus, a long median filter preserves the longer transient, but removes the events that occur at a smaller time-scale, such as the graywater pump. Subtracting the medianed stream (P_M) from the original data stream (P) results in the residual stream (P_R). Now the medianed stream contains the longer-scale events and the residual stream contains the smaller-scale events. Various length median filters can be employed if there are multiple time-scales present in a data stream. At each time-scale, there are median streams and residual streams. This decomposition is represented in Fig. 3-2.

3.1.2 Event Detection and Feature Extraction

The application of a multi-scale filter for event-scale separation enables the detection of different scale events. This thesis distinguishes between three types of events that occur at different time-scales: “geometric,” “statistical,” and “continuous.” Examples

for these three types of events are shown in Fig. 3-3. The main load on and off events, as well as discrete transitions between states of a load are referred to as “geometric” events, since the salient features are extracted from the shape of the transient. There is always a steady-state change after a geometric event. For example, Fig. 3-3a shows an example run of the graywater pump from USCGC Spencer. A zoomed-in window shows the on-transient. The transient events for geometric events are typically only fractions of a second to a few seconds long. In contrast, fluctuating power variations, for example due to dynamic loading conditions, are referred to in this paper as “statistical” events. Statistical events do not necessarily indicate that the load has transitioned to a different steady-state level. The identification of statistical events enables the disaggregation of power during the event itself and the detection of any change in steady state as a result of the event. For example, Fig. 3-3b shows the fluctuations in power of a controllable pitch propeller (CPP) pump from USCGC Spencer. These fluctuations in power occur on longer time-scales and do not have consistent changes in steady state. These fluctuations are viewed as an ensemble and modelled as stochastic processes. Finally, “continuous” events have smoothly varying power demand, for example, due to changing demand of variable frequency drives (VFDs). The load’s power consumption can not be strictly identified by turn-on and turn-off events because they do not have a repeatable power consumption pattern. For example, Fig. 3-3c shows the anchor windlass VFD from USS Indianapolis. This VFD is used to control anchor windlass speed while raising and lowering the anchor or adjusting and working mooring lines. A continuous event can also include loads that are continuously on, but experience changes in power due to degradation in system health or changes in operating condition. This section presents an overview of feature extraction for these three methods. Classification using geometric and statistical methods are presented in detail in Chapter 4 and Chapter 5, respectively.

3.1.2.1 Geometric Methods

There exists an abundance of physical features that reveal information about load behavior and are valuable for load identification. When a load energizes or changes

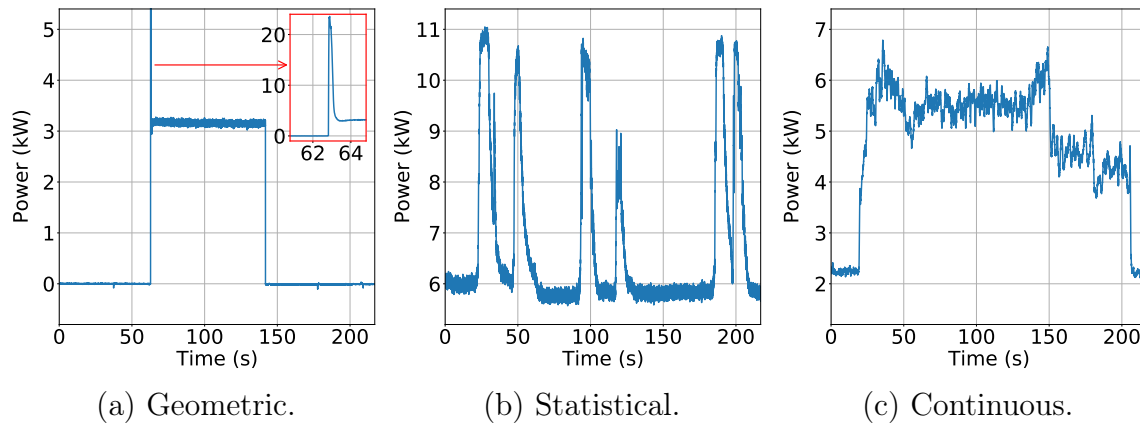


Figure 3-3: Real power streams of events suited for geometric, statistical, and continuous methods.

state it manifests in the power streams as transient behavior. This transient behavior can be detected with an edge detector or change-of-mean detector [85] and features can be extracted. A median filter can be applied to the power streams prior to edge detection in order to preserve sharp edge events, while removing noise [84]. The edge detector applied in this research uses the apparent power (S) stream, calculated as, $S = \sqrt{P^2 + Q^2}$. Converting to apparent power simplifies load detection to a single data stream. There are many approaches to edge detection. One option is to look for large peaks in the first-order difference of the stream. Another option is to convolve the stream against the Laplacian of a Gaussian [86] kernel to compute the smoothed second-derivative. This effectively maps step changes in apparent power to zero-crossings for easier detection. An empirically-determined threshold should then be set to remove zero-crossings that are due to small variations of the resulting convolution. A zero-crossing detector is then used to find the location of the events.

After a transient event is detected, the power streams for each phase are examined to calculate a set of features. An on event produces a change in steady state and a transient which can be characterized by its peak, duration, and shape. The peak is due to the in-rush current as a load turns on. The duration of the transient is the time it takes for the load to reach steady-state. For example, Fig. 3-4 shows an example load turn-on transient, representative of any power stream (e.g., P , Q , or S) or higher-

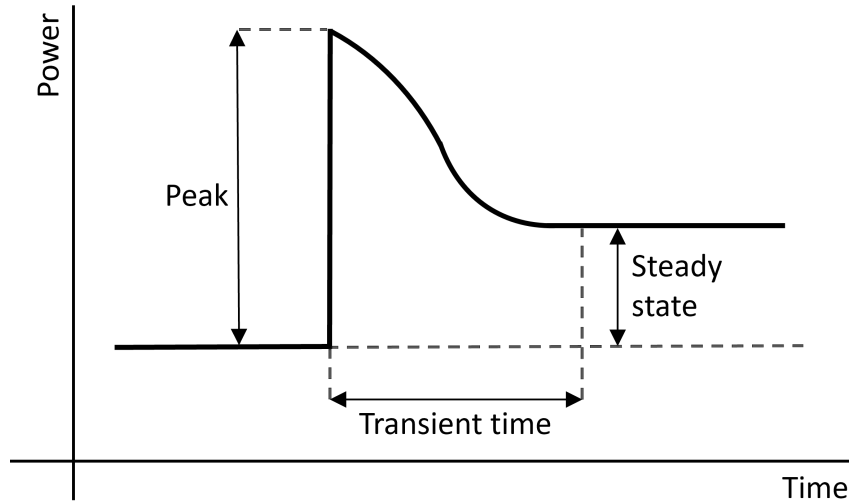


Figure 3-4: Steady-state, peak, and transient time features for an example load turn-on transient.

order harmonic current envelope. The maximum (peak) power at inrush, steady-state power, and transient time are labelled. Changes in steady state powers after a load turns on can be calculated as the difference between the median (or alternatively, the mean) values over Δt_M length windows before and after the transient. The length of Δt_M is determined empirically based on the rate of event generation at the site to establish a reasonable steady state time. The maximum power at inrush is defined as the difference between the maximum value of the transient and the median (or alternatively, the mean) value of a Δt_M length window before the transient. The transient time (i.e., the duration of the start-up transient) is effectively the time it takes for the load to reach steady-state. To determine the transient time, first a short (e.g. 10 point) moving average should be applied to the first-order difference stream of apparent power. If there is a first-order difference less than a set threshold (e.g. -500 W), indicating a large negative slope, the steady state is determined to be after this value. Then if the rolling mean of the first difference stream is less than a threshold (e.g. 5 W), it indicates that the apparent power stream has reached steady state. Since there is no transient peak when a load turns off, an off event is only characterized by its change in steady state. Changes in steady state after a load turns off are defined as the difference between the median (or alternatively, the mean)

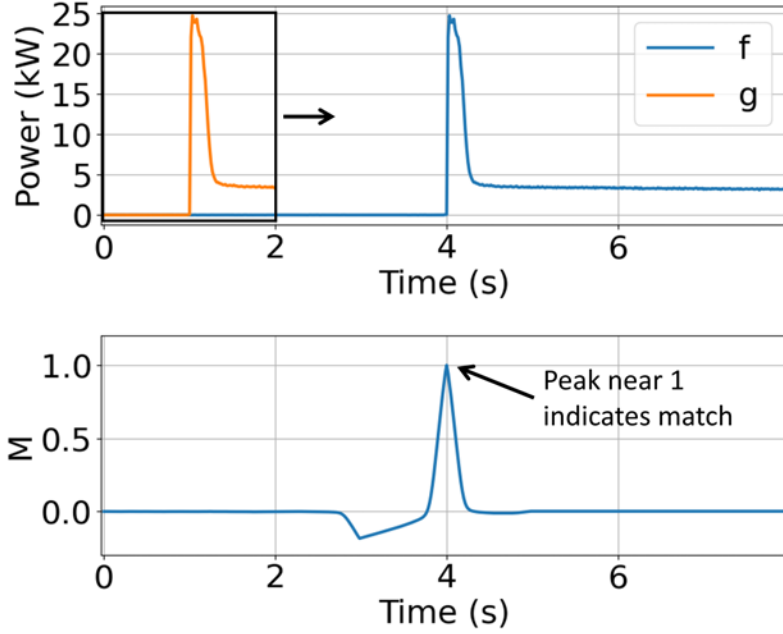


Figure 3-5: Correlation metric for transient matching.

values over Δt_M length windows, before and after the off event.

In addition to these features, the transient events can be characterized using a correlation algorithm. The correlation algorithm matches the shape of the input data to known exemplars, and is fully described in [5]. Consider two sampled waveforms f and g , where f is an observation or input signal and g is a load exemplar or example waveform. First, \bar{f} and \bar{g} , which are the mean of f and g , respectively, are subtracted from the original signals f and g , in order to remove the dc offsets. This is given as:

$$\hat{f}[m] = f[m] - \bar{f} \quad (3.1)$$

$$\hat{g}[m] = g[m] - \bar{g}. \quad (3.2)$$

Then the correlation metric is computed over sliding windows of input data, given the full waveform f is longer than g :

$$M[k] = \frac{\sum_m \hat{f}[m] \hat{g}[m - k]}{\sum_m (\hat{g}[m])^2}. \quad (3.3)$$

Here, we are computing the cross-correlation (or sliding dot product) of \hat{f} and \hat{g} and dividing by the squared magnitude of \hat{g} . When M approaches 1, this indicates that the exemplar and observation match in both shape and amplitude. Fig. 3-5 shows example observation data, an exemplar, and the resulting correlation metric as the exemplar window slides across the observation data. A peak-finding algorithm is used to locate the local maxima. The value of the peak is used to find the correlation score, where a score of 0 indicates a perfect match:

$$C = \left| 1 - \max_k M[k] \right|. \quad (3.4)$$

These geometric features are further characterized in Section 3.2 and Section 3.3, and demonstrated with an adaptive classifier in Chapter 4.

3.1.2.2 Statistical Methods

Loads can exhibit variable power demand in response to dynamic loading conditions. Fig. 3-6 shows the real power stream as a CNC router makes a cut through the shown wooden board. The vertical position of the router cutting tool is held constant. The contour of the workpiece changes the axial depth of cut. When the cutting tool is freely rotating and not engaged in the workpiece, there is an approximately steady power consumption level. However, when the cutting tool engages the wood, the power varies significantly, and clearly resembles the workpiece contour. There is no obvious way to disaggregate this fluctuating power behavior when other loads are also present.

Power fluctuations due to changing load demand occur in response to understandable physical behavior; however, these fluctuations appear as unpredictable events in the power stream. The fluctuating power behavior of these loads can be modelled with stochastic processes. Just as the identification of geometric events relies on an event detector and a feature extractor, there is an analogous event detector and feature extractor for statistical events. That is, the event detector is used to find events. At each event, a set of extracted features is used for load classification. To simplify

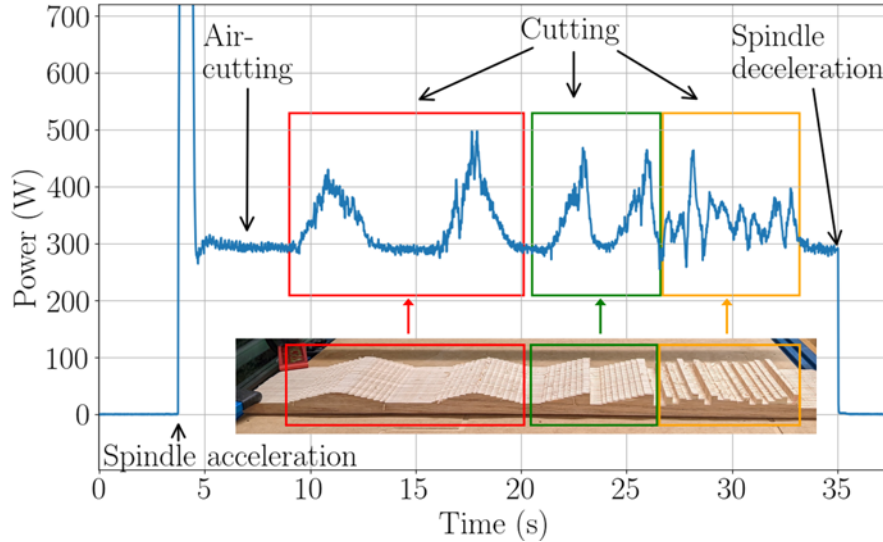


Figure 3-6: Power stream of a CNC router showing power fluctuations corresponding to the cutting of areas highlighted in the wood board.

the computational effort of identifying statistical events, the apparent stream is used for event detection. To provide greater load separation in the feature space, the feature vector for classification uses features extracted from the real and reactive power streams.

First a seven-point rolling median filter is run on each of the power streams (i.e., P , Q , and S) to reduce noise. Then a longer rolling median filter, typically on the order of seconds or minutes, is run on each of the power streams. This median filter length should be set by the user based on the expected time-scale of power fluctuations. The median streams are subtracted from the original streams. The resulting streams are referred to as the residual streams, which contain the statistical events. For example Fig. 3-7 shows the real power stream and residual streams for two loads. The CNC router, as shown in Fig. 3-7a uses a 10 second rolling median filter. The bilge and ballast pump, as shown in Fig. 3-7b uses a 30 second rolling median filter.

Residual streams are extracted from a rolling window of input data. The window size and overlap are parameters that the user can set. The window size is selected to be larger than the expected duration of variable load activity. In the case studies presented in this thesis, the rolling window length was set equal to the median filter size. The standard deviation, denoted here as σ , is calculated for the middle 50%

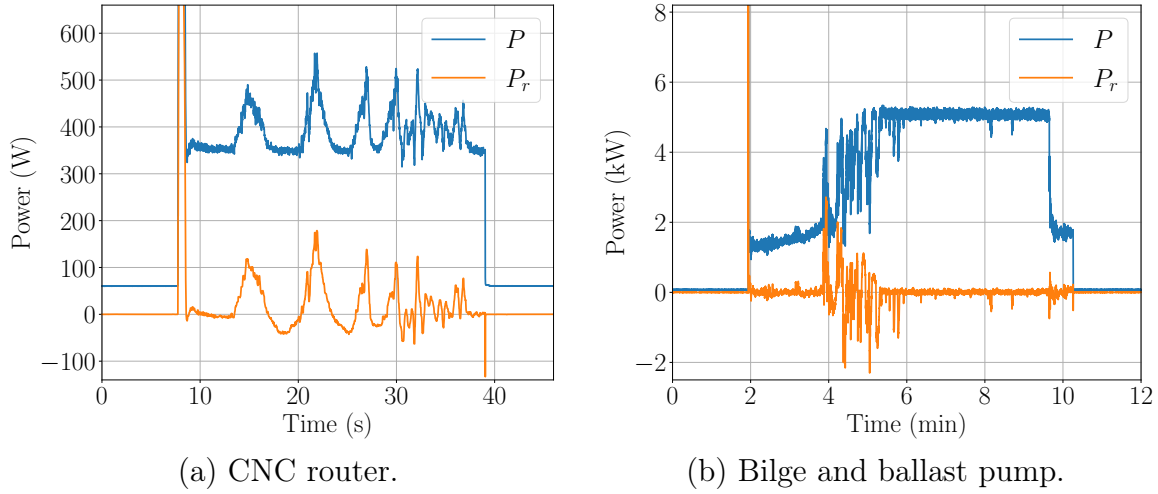


Figure 3-7: Real power streams and residual streams.

of each residual apparent power (S_r) stream window. This decreases the likelihood that large fluctuations occur only on the edges of the window. If σ is greater than a user-defined threshold, it indicates significant activity of interest (i.e., a statistical event is detected) and further statistical feature extraction is performed. For feature extraction, any stream that relates to the physics of load behavior can be used. This can include, for example, real power residual stream (P_r), reactive power residual stream (Q_r), first-order difference of the real power residual stream (P'_r) and first-order difference of the reactive power residual stream (Q'_r). Other power streams could be relevant in some scenarios, such as the higher-order harmonic current spectral envelope streams when multiple power-electronic loads are monitored.

Two statistical features are presented here. The first is a zero-crossing metric, and the second uses the empirical cumulative distribution function (ECDF). For the zero-crossing metric, the window is detrended by subtracting out the mean, which allows for analysis on the variations around zero. The zero-crossing metric is the number of zero-crossings for each detrended waveform normalized by the length of the signal N , i.e.,

$$ZC = \frac{\# \text{ of zero-crossings}}{N}. \quad (3.5)$$

The magnitude of power fluctuations may depend on load and operational condi-

tion. The fluctuation variance can change based on factors such as the steady-state power consumption of the load or the type of material being cut by a CNC machine. That is, the magnitude of fluctuating power may not be a reliable feature. If the data is normalized to a constant range, the stochastic behavior can be evaluated independent of the change in overall fluctuation magnitudes. Min-max normalization is performed on each data stream individually for each window. The range of the data is transformed into $[0, 1]$ for each stream x through the transformation,

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (3.6)$$

For each window, the ECDF is computed on the normalized residual stream, x_n by creating a histogram of data values, and then applying a cumulative sum. Just as a histogram is the empirical estimate of a probability density function (PDF), an ECDF is the empirical estimate of a cumulative distribution function (CDF). A number of bins should be chosen by the user. The value of the ECDF for a given bin b , denoted as \hat{F}_b , is given as the relative frequency of all observed values of x_n being less than or equal to the value of x_n represented by bin b (denoted as x_b in Eq. (3.7)). This is given by the following for each data window of length N :

$$\hat{F}_b = \frac{1}{N} \sum_{i=1}^N I(x_i \leq x_b), \quad (3.7)$$

where I is the indicator function, given by

$$I(x_i \leq x_b) = \begin{cases} 1, & \text{if } x_i \leq x_b \\ 0, & \text{if } x_i > x_b \end{cases}. \quad (3.8)$$

For the ECDF statistic, two nonparametric, distribution-free tests to measure the equality of one-dimensional probability distributions are considered: 1) the Kolmogorov-Smirnov (KS) test and 2) the Cramer-von Mises (CvM) test [87]. Both tests compare the distance between the ECDF of a sample and reference ECDF distribution. The KS test statistic, D , represents the least upper bound (or maximum) of the pointwise

difference between the sample distribution function, $F_n(x)$ and the known theoretical distribution function, $F_0(x)$,

$$D = \max_x \left| \hat{F}_b(x) - F_0(x) \right|. \quad (3.9)$$

Alternatively, the Cramer-von Mises (CvM) criterion, w^2 , uses the integration of the squared value of the difference between $F_b(x)$ and $F_0(x)$,

$$w^2 = \int_{-\infty}^{\infty} (\hat{F}_b(x) - F_0(x))^2 dF_0(x). \quad (3.10)$$

When comparing two empirically observed distributions, it can be observed that the KS test corresponds to the l_∞ norm and the square root of the CVM criterion corresponds to the Euclidean distance or l_2 norm. These metrics are demonstrated in Chapter 5.

3.1.2.3 Continuous Methods

Tracking the operation of continuously variable loads that demand ever-changing power arises for VFDs, light dimmers, and other loads controlled by power electronics. For example, VFDs are often used to optimize and control energy consumption, speed, and torque of induction motors. These loads do not always have repeatable transient behavior or power consumption pattern. However, the current waveforms of these loads consist of structural features that can be identified in both the time and frequency domains [7]. The power electronics contribute to nonsinusoidal current waveforms, and thus higher order harmonics for these loads. This thesis does not present in-depth methods for tracking continuous events due to VFDs, which can instead be found in [7, 88]. References [7] and [88] use a waveform-based estimator using the fundamental and higher harmonic current waveforms to disaggregate the power consumption of these loads from that of constant-power loads. Continuous events can also arise when *continuously on* loads have slow changes in power from changes in operating conditions or a degradation in system performance. For in-

stance, in ventilation systems, clogs or leaks in the system manifest as changes in power. However, these changes in power are difficult to associate with a particular machine in the aggregate stream. Fans in a ventilation system are typically driven by induction motors. Thus, using rotor slot harmonics can serve as indicators of fan blade speed. In the typical operating region of an induction motor, speed is closely linked to power. The use of power monitoring and slot harmonic tracking for fan and motor fault detection is demonstrated in Chapter 6.

3.1.3 Event Mapping and Event Confirmation

The features extracted for each event are then mapped to a specific load, using one of various pattern recognition approaches. Event mapping is demonstrated for geometric and statistical events in Chapter 4 and Chapter 5, respectively. After an initial mapping, constraints need to be checked to ensure a correct output. The first check is to ensure that two consecutive on events or off events are not outputted for a given load. This would indicate that either an event was missed, an event was misclassified, or a nonevent was classified as an event. In the case of misclassification, the process goes back to the event mapping and reclassifies or determines that it is not an event. This is represented by the feedback arrows in Fig. 3-1. In the case of a missed event, a decision is made about which event to display. Depending on the application, the decision can be made to either reduce the possibility of incorrectly displaying that a load is on or to reduce the possibility of incorrectly displaying that a load is off.

Constraints are also checked to be used for procedural oversight. The operation of FSM loads and interdependent loads are checked for standard operating procedures, ensuring that equipment or systems that go through multiple stages of operation are sequenced properly. That is, if loads are operating in the wrong sequence, it can alert operators to deviation from standard operating procedure. If a load is missing from the sequence, it can either be a deviation from standard operating procedure or indication of a missed load event due to load degradation. That is, it is possible that the load event did occur, but it is operating in a degraded state and the classifier did not identify the load.

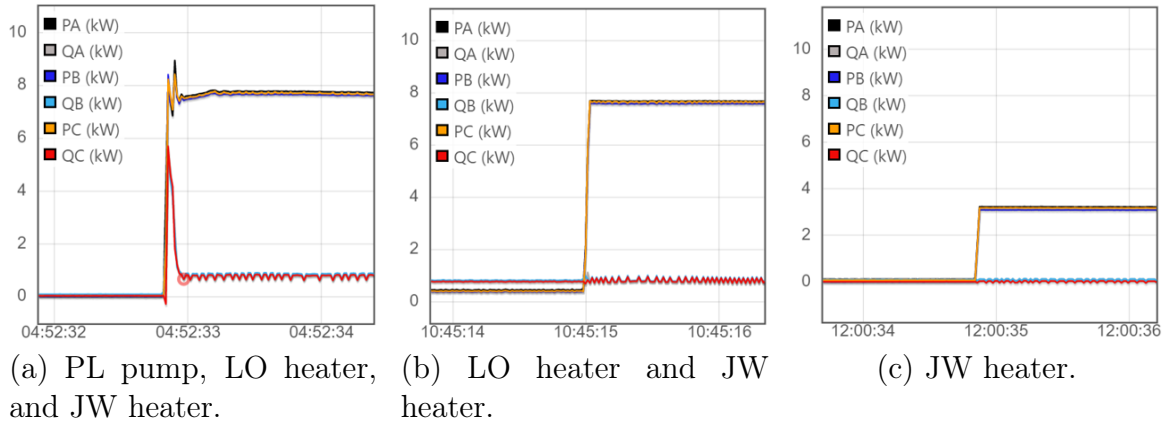


Figure 3-8: On-transients of MPDE system loads.

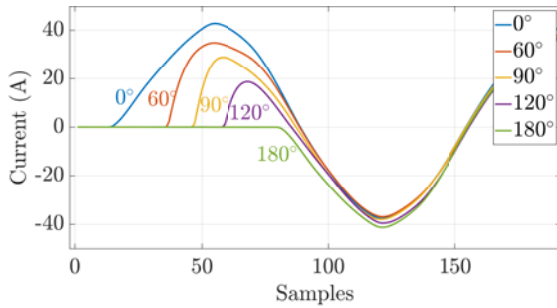
For instance, the MPDE jacket water (JW) heater, lube oil (LO) heater, and prelube (PL) pump frequently turn on and off either simultaneously or in a specific sequence. The PL pump and LO heater work in tandem and are served by the same controller. The controller is activated automatically when the engine speed falls below 150 rpm. When the controller is activated, the PL pump comes online and the LO heater enters automatic mode. In automatic mode, the heater energizes when the oil temperature falls below 90° F and secures when the oil temperature goes above 120° F. If the controller is activated when the engine is already cold, the PL pump and LO heater will activate simultaneously. If the engine is warm, the PL pump will run by itself until the engine temperature falls below the threshold and the LO heater comes online. The JW heater is activated from a separate controller and thermostat. However, the JW heater also runs automatically based on 90° F and 120° F setpoints, and lube oil and jacket water temperature typically track closely together. Fig. 3-8 shows the on-transients for a healthy JW heater operating alone and in tandem with the other MPDE system loads. Since these three loads often operate in tandem or in sequence, the identification of two of these loads without the third is indication of a possible missed load event. For instance, the JW heater was shown in Section 1.2 to operate in a faulty state with a reduced steady-state power. If an event was missed, the classification process then goes back to the event mapping step and determines if there is an event that could be the faulty JW heater.

3.2 Transient Space Characterization

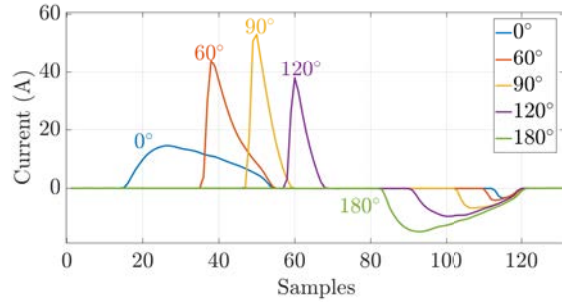
The presented geometric features are useful for load identification but are prone to drift in the feature space. The electrical characteristics of a load can change over time with aging and degradation. These changes create observation variability within the feature space that may make it difficult for a NILM to recognize a load. At the same time, this variability may be a useful diagnostic indicator. Some variability may even be present for a healthy load, depending on features of the power system itself, like turn-on angle and voltage amplitude, or depending on reasonably varying features of mechanical load operation [89]. Successful recognition of healthy and faulted loads requires an informed choice of features. In some scenarios, load variability can be tested for with laboratory or on-site testing. Laboratory testing of loads was conducted to characterize load variability due to factors such as turn-on phase angle, operating conditions, and environmental factors. Other times, the variability needs to be recorded on-site. Examples are presented of load variability of monitored shipboard loads.

3.2.1 Voltage Phase Angle

Many electrical loads, such as lighting, induction motors, and power supplies, can draw large inrush currents, which in some cases may exceed ten times the steady-state power. The magnitude and characteristics of the inrush are dependent on the physical operation of the load. Load transients often serve as the “fingerprint” for identifying loads. One factor affecting the inrush transient is the utility line cycle. The phase angle of the electric utility affects the inrush current size, duration, and shape for some loads. Thus, characterizing a load’s inrush current transient at various voltage turn-on angles can allow for load identification and can further be used for diagnostic evaluation of equipment. This implies that knowledge of the actuation mechanism can also aid in determining if changes in transient shape are due to natural variations or due to an underlying fault condition. If the load is activated with a mechanical switch or an instantaneous/random solid state relay, the load can turn on at any point in the voltage line cycle. Contrarily, if the load were actuated with a solid-state relay



(a) Centrifugal fan.

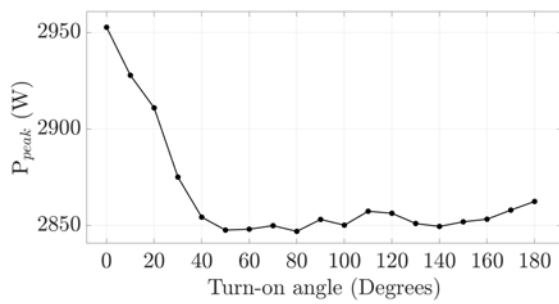


(b) Electronically commutated motor fan (at maximum speed).

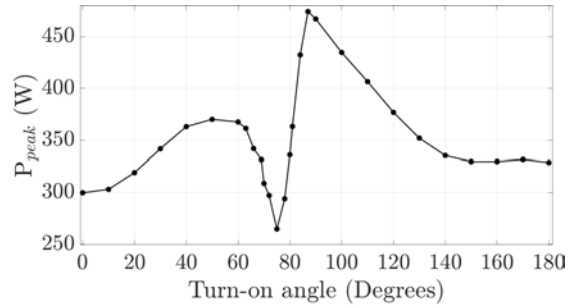
Figure 3-9: Inrush current transients at various voltage turn-on angles.

with zero-crossing detection, the variability in the transient behavior would become significantly smaller, and a large variation in the transient peak would be a sign of a possible anomaly.

Using a TRIAC-based phase controlled switch, built by Erik Saathoff [89], loads can be turned on at a desired phase-angle of the utility voltage, permitting sweeps over the turn-on angle. Testing was conducted on two fans, a 1-hp induction motor centrifugal fan and an electronically commutated motor (ECM) variable speed fan, to demonstrate how the turn-on voltage angle can affect the inrush transient. The phase-controlled switch started the loads at a selection of phase angles: 0° , 60° , 90° , 120° , and 180° . The resulting inrush currents are shown in Fig. 3-9. Like most inductive loads, the centrifugal fan's worst case current is near the voltage zero crossings. During startup, the core inductance of the machine significantly impacts the inrush current. The turn-on angle that can maximize the volt-second integral on the induction-machine model will generate the largest current, which happens when starting near a zero-crossing. However, the series capacitor will shift the position of the worst case somewhat. The electric variable speed fan presents significantly different inrush behavior. The ECM motor uses a full-bridge rectifier to convert the ac input to dc. For testing consistency, the fan was run at its maximum speed. In contrast to the centrifugal fan in which the current peaks at the voltage zero-crossings, the variable speed fan's current inrush is highest when the voltage nears the peak, i.e., when the turn on angle is close to 90° or 270° . When the turn-on angle is 0° or 180° ,



(a) Centrifugal fan.



(b) Electronically commutated motor fan (at maximum speed).

Figure 3-10: Inrush power peak at various voltage turn-on angles.

the initial voltage is zero and the current rises steadily from zero without any spiking. After the initial inrush, the fan continues to consume power in pulses during each half line cycle. The 0° and 180° cases look very similar, with the latter being delayed by half a line period and inverted. Any load that does not have line voltage polarity dependence, such as this variable speed fan, will behave similarly. The inrush current for a turn-on angle of α will be the same as $\alpha + n180^\circ$.

It is also useful to use spectral envelopes to view the power consumed by the load averaged over an ac line cycle with rather than the pulsing current itself. Each fan was energized ten times for each turn-on angle from 0° to 180° in increments of 10° . For the ECM variable speed fan, the region between 60° to 90° contained large first derivatives. This region was retested with smaller 3° increments. Fig. 3-10 shows the peak inrush power of both fans as the average of ten activations for each turn-on angle. As expected from the current waveforms, these two fans have different inrush power behavior. For the centrifugal fan, the maximum power at inrush is at 0° . For the variable speed fan, the maximum power at inrush is near 90° .

As shown, the initial inrush changes based on the turn-on angle. Thus, by using the phase-controlled switch to keep the phase angle constant, the effects of other operating or environmental conditions can be tested. By changing how a load is cycled on or off, the turn-on waveform changes and provides more information about the load. For the variable speed fan, the mechanical inertia, combined with the speed controller, causes different power draw signatures as the fan approaches steady state.

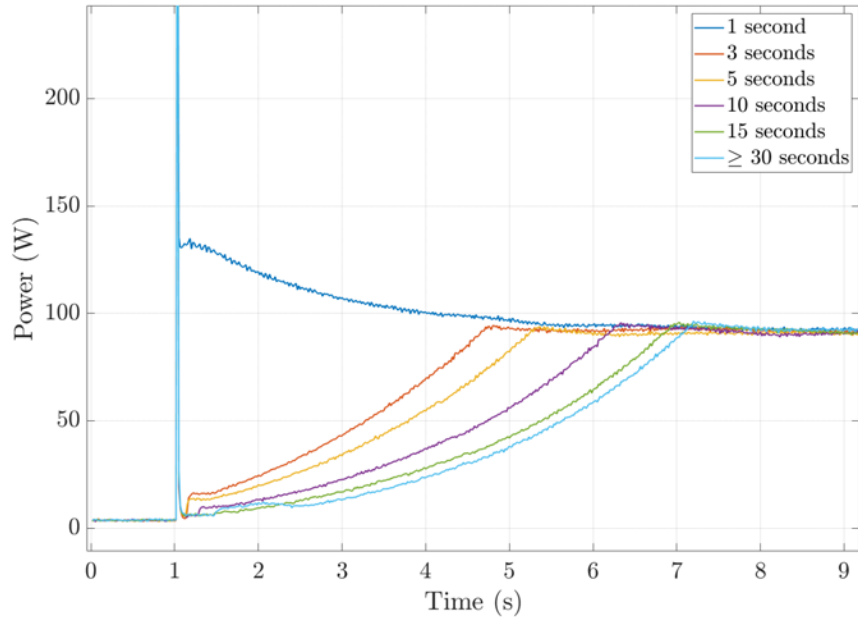


Figure 3-11: Power of the variable speed fan with varying off-time between load activations.

Fig. 3-11 shows the full turn-on transient after variable times between an off-event and the next load activation. These tests are performed at a turn-on angle of 75° . In each test, the transient starts with a large peak as the dc buffer capacitor is charged. Afterward, the speed controller draws a variable level of power depending on the starting conditions of the fan, e.g. the initial rotation speed of the fan. As the time between load activations increases, the time to reach steady state increases. This effect stops when the off-time exceeds 30 s, as this is the time required for rotational velocity to reach zero. Unlike the other cases, the 1 s off-time case does not drop to low power after the initial inrush spike, and instead consumes more than steady state power. The speed controller likely has both soft-start and running modes. In the 1 s case, the fan doesn't slow down enough to reach soft-starting territory, so the speed controller tries to quickly return to the correct speed.

3.2.2 Fault Conditions

Loads experiences changes in their overall power consumption due to degradation of system components. For example, with knowledge about the type of fan, a NILM

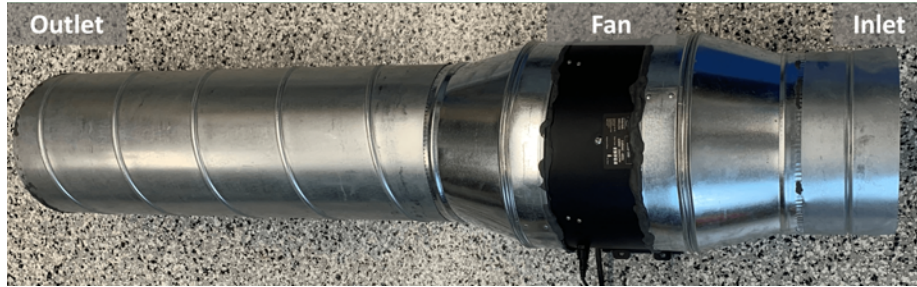


Figure 3-12: Experimental setup for variable speed fan in ventilation system.

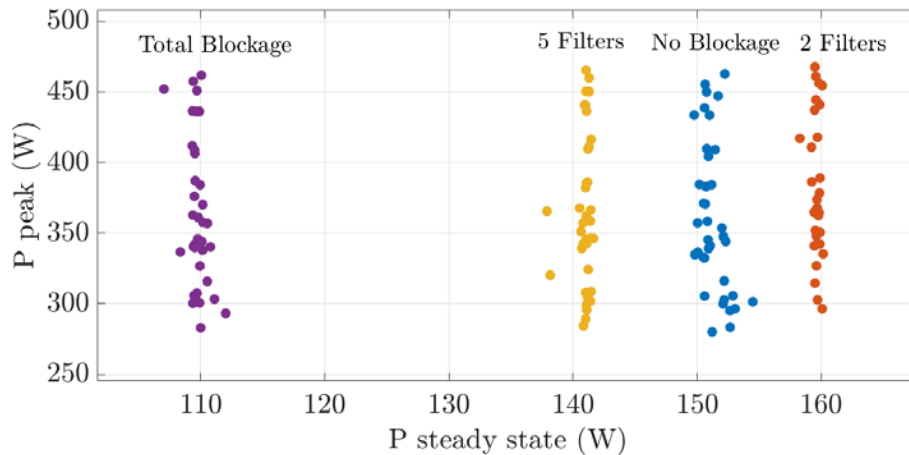


Figure 3-13: Inrush power peak versus steady state for the variable speed fan with varying filter blockage levels.

can detect when a ventilation system has a leaky duct or clogged filter by observing changes in the fan’s power consumption. The variable speed fan was used in a ventilation set-up to explore the relationship between filter blockage and power consumption. The ventilation setup is shown in Fig. 3-12. To emulate blockages in an air ventilation system, 0.5 inch polyester plastic filters, rated to Minimum Efficiency Reporting Values (MERV) 5, or 3 micron permeability, are introduced on the intake side of the system. Complete obstruction of the fan inlet was also emulated, effectively reducing air flow to zero. At each blockage level, the fan is run once at each turn-on angle from 0° to 360° in increments of 10°. Fig. 3-13 shows the inrush peak real power versus the steady-state power level for each run at various blockage levels.

This fan is a constant rpm electronically commutated motor (ECM), which means that the the blades will spin at the same speed regardless of external static pressure changes. Based on the fan pressure curve, a small decrease in flow rate caused by the

first two filters will cause a sharp increase in static pressure, increasing the torque on the motor shaft and raising the overall power consumption. However, the curve quickly flattens, resulting in lower torque for further flow restrictions. As more filters are added and the flow rates drops toward zero, the fan is no longer producing effective work. The shaft is only loaded by drag, causing the power consumption to decrease. The inflection point of the power curve, and the power difference between zero and full blockage, are dependent on the fan design. Constant flow rate ECM fans produce similar results, but their variable rpm operation can make the curve more exaggerated [90]. Despite the steady-state power changing as the blockage level increases, the range of peak power values does not change. This range of values is consistent with the peak power characterization in Fig. 3-10b. The inrush transient of this load is dominated by the charging of the large internal dc capacitance, and this process is completed before a blockage can affect the fan blades. This series of tests, performed with the phase-controlled switch, confirms that a blockage fault does not affect the variability-space peak-power axis for this type of fan. Thus, a load identification algorithm focused on the inrush transient can recognize this load even as it experiences this type of fault without any additional training. It can further be used to diagnose the fault condition by observing the changing steady-state power levels.

3.2.3 Environmental Factors

Investigation of a load's feature space can also extend into the environmental variables, such as temperature. A 0.7-hp single-phase axial fan driven by an induction motor is used in the same ventilation set-up as the variable speed fan. A permanent split capacitor motor, such as this axial fan, does not have a speed controller to maintain a constant rpm. Thus, normally when the external static pressure of the system changes it leads to power changes based on the fan's characteristic fan curve [91]. However, this system does not exhibit significant changes in steady-state power based on filter blockage. Instead, the fan's power draw exhibits a significant temperature dependence. A majority of loads do not operate under strict temperature control and their temperature may change with both the ambient and with

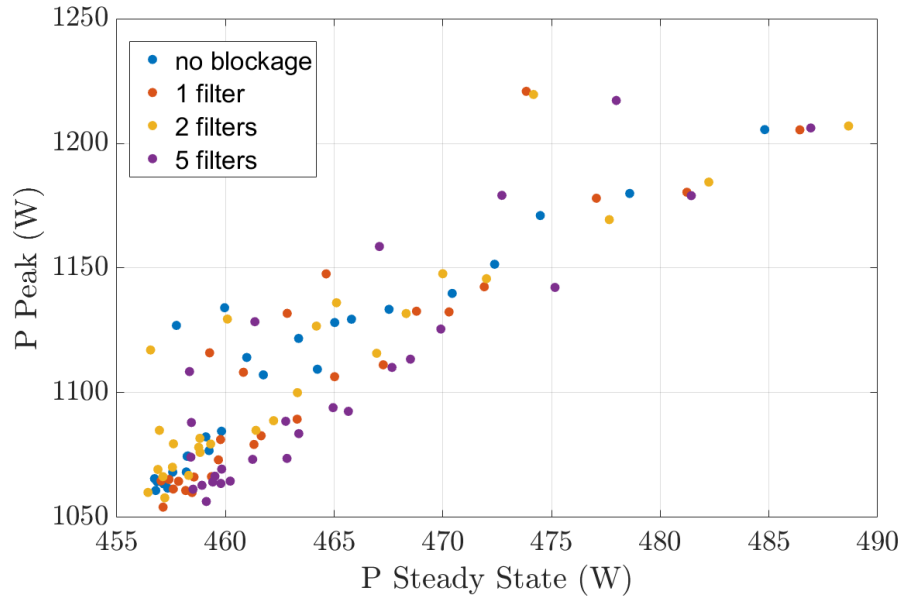


Figure 3-14: Peak inrush and steady state power consumed by an axial fan at various blockage levels. Temperature decreases as points move up and to the right.

internal heating. Motors generate heat from various losses, which will slowly increase their temperature until they reach steady state with the ambient. A high thermal mass leads to long thermal time constants, and thus investigating inrush at different temperatures is useful. Since the rotor bars in an induction machine change conductivity with temperature, the power consumed should also change during a turn-on transient.

An experiment is performed in which the axial fan ventilation setup is subjected to colder temperatures, then starts with a random turn-on angle for 27 runs each at various filter blockage levels. Fig. 3-14 shows the inrush peak power versus the steady-state power level for each run. There is no clear dependence of the steady-state power level on filter blockage, as is the case for the variable speed fan, as shown in Fig. 3-13. Instead, both the steady-state and peak power depend on the temperature, as shown in Fig. 3-15. Here, the temperature is measured at the outlet of the ventilation system. As the system warms up, the steady-state and inrush power both decrease. This additional information can improve load identification algorithms by including new extremes on expected operating behavior. In addition, the lack of dependence on blockage contrasts with the previous ventilation example, indicating

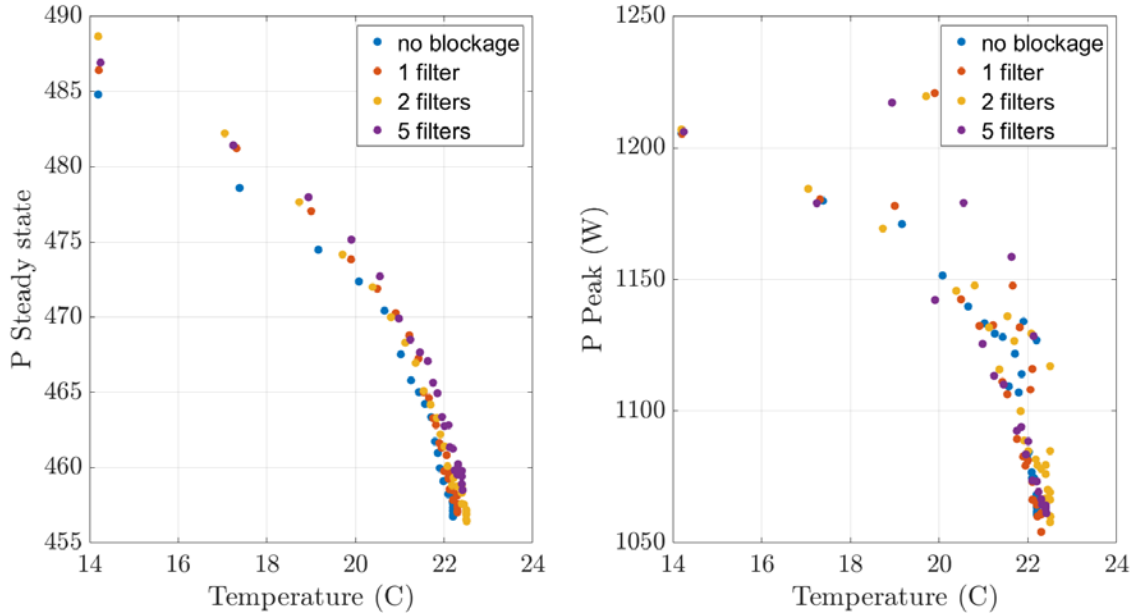
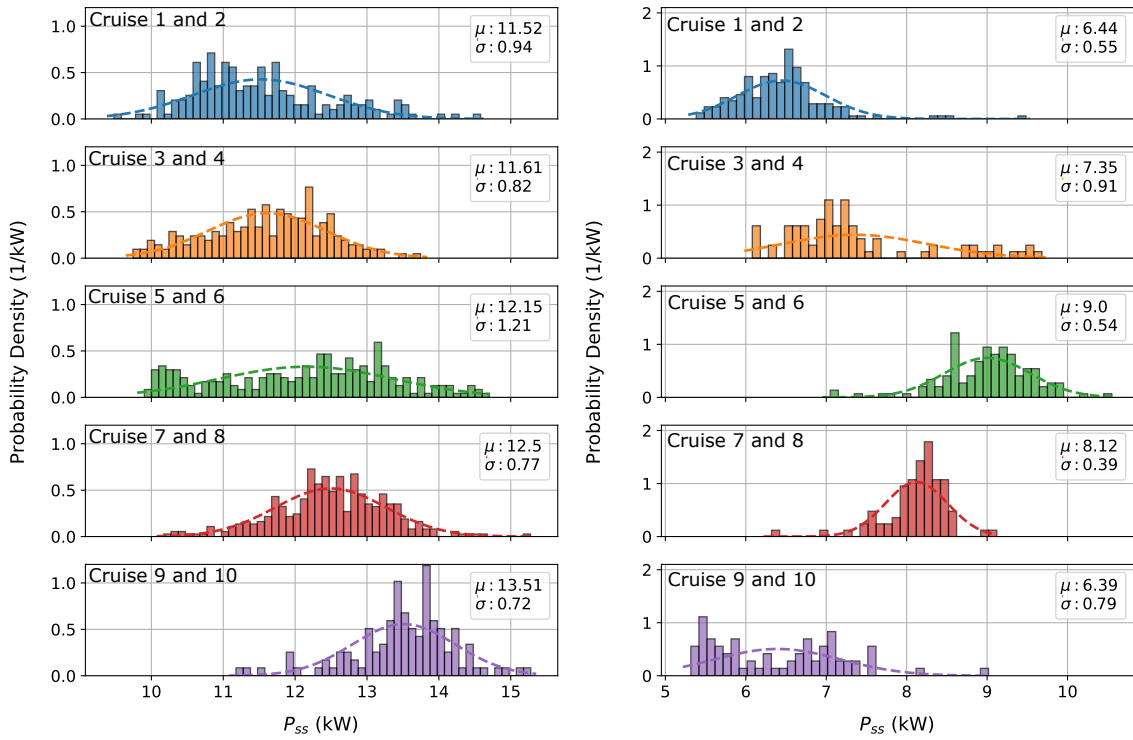


Figure 3-15: Axial fan steady state real power (left) and peak inrush real power (right) versus temperature.

that correlations between load variables are not necessarily universal and must be characterized for each load. Incorrectly applying assumptions to this duct fan can result in mistakenly attributing a decrease in power to a leaky duct, rather than a hot motor. The temperature dependence can also be used as a diagnostic indicator of the associated temperature control system or the fan itself, as increased temperature can indicate a winding short, failed bearing, or other lossy faults.

3.2.4 Shipboard Load Variability

Domain knowledge about how loads drift in a physically-informed feature space is necessary to guide the characterization of this drift for separability testing. Two shipboard loads are presented as examples for characterizing load behavior, a fuel oil purifier (FOP) centrifuge and a controllable pitch propeller (CPP) pump. For illustrative purposes, consider steady-state real power (P_{ss}) and steady-state reactive power (Q_{ss}), where the steady-state power was extracted from load on-events, calculated for 0.5 seconds after each event. In a Q_{ss} versus P_{ss} feature space, the drift is likely to occur along a single major axis. Fig. 3-16a shows the normalized



(a) FOP centrifuge.

(b) CPP pump.

Figure 3-16: Normalized histograms of steady-state real power (P_{ss}) for two cruises at a time. Gaussians are fitted with labelled means and standard deviations.

histograms of the FOP centrifuge’s P_{ss} for five different periods in time. The data is divided into “cruises,” where each cruise represents a period of one to three months in which the vessel is mostly underway at-sea. There is a break in data collection during the vessel’s in-port periods after each cruise. Each subplot shows two cruises of data, e.g., the first row is the first and second cruise, the second row is the third and fourth cruise, and so on. A Gaussian probability density function was fit to each dataset with the mean (denoted as μ) and standard deviation (denoted as σ) from the data, as shown in the labels on the plots. The mean of the FOP centrifuge’s steady-state real power increases as time progresses. Although this example shows only one dimension having a Gaussian distribution, a similar trend can be seen in the steady-state reactive power. In Fig. 3-17, the Gaussian-like distribution of P_{ss} and Q_{ss} over the entire dataset of ten cruises can be observed for the FOP centrifuge. The increase in power can likely be attributed to load aging and wear.

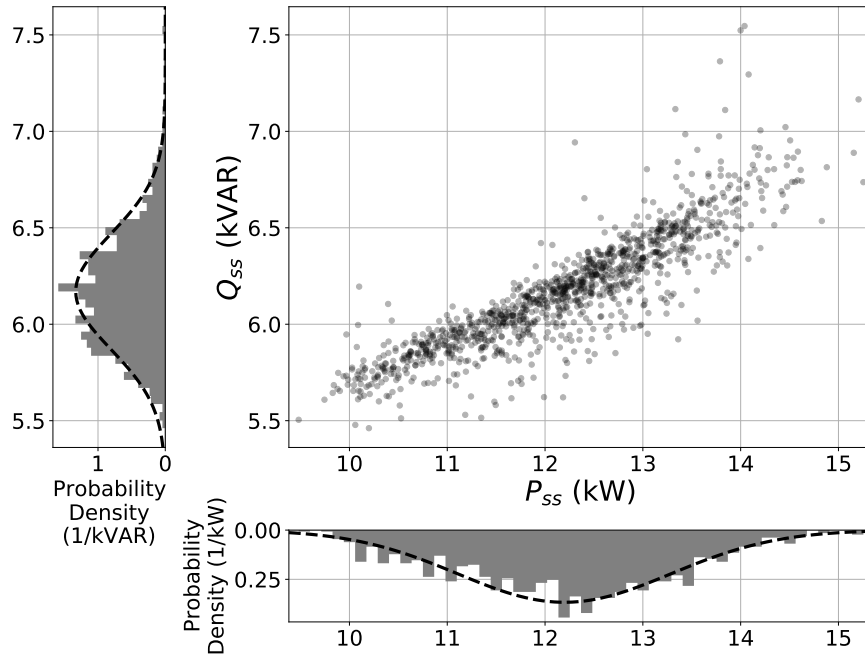


Figure 3-17: Q_{ss} versus P_{ss} for the FOP centrifuge after ten cruises of data plotted with the normalized probability density for each axis.

A plot of normalized histograms over five periods in time for the CPP pump is shown in Fig. 3-16b. These are the same time periods as for the FOP centrifuge’s histograms in Fig. 3-16a. It can be observed that unlike the FOP centrifuge for which P_{ss} generally trends in one direction over time, the CPP pump’s power first drifts higher before drifting back to a similar power level as its original state. Intuition for the drift of the CPP pump is provided by examining the change in operating condition. The drift in the pump’s steady-state real power and reactive power consumption over the monitored four-year period correlates with the operating fluid pressure normalized by temperature, as was shown in Section 1.2 and plotted again in Fig. 3-18a. There was a large increase in both the power draw and the normalized operating pressure in February 2018, after the replacement of the hydraulic control valves. Then, as the normalized operating pressure decreased over time, the power draw also subsequently decreased. At the same time, the variability in the peak inrush power (P_{peak}), as shown in Fig. 3-18b, does not correlate with the operating pressure. Similar to the laboratory example previously presented, this variability is due to the time instant

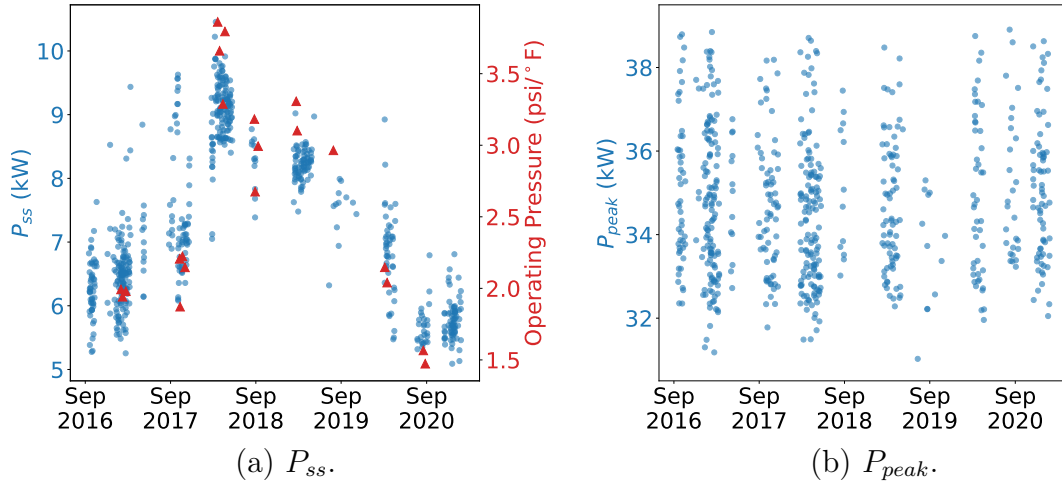


Figure 3-18: CPP pump characteristics over time.

with respect to the voltage line cycle that the load turns on.

These two loads demonstrate that with a limited dataset in time it is difficult to determine how much of the load’s total distribution is accounted for. Accordingly, the NILM feature selection process should take all known data into account when evaluating the suitability of a feature space. As operating conditions change or faults occur and are fixed, a load can return to a previously observed region in the feature space. Although not all loads appear perfectly Gaussian, a Gaussian approximation still can provide a sufficient characterization while being easy to compute and describe. With this assumption of load characterization, the following section presents the hyperellipsoid characterization for checking load separability.

3.3 Feature Space Evaluation

As shown, a load’s electrical characteristics are prone to variation and drift. The applicability and utility of pattern classifiers for fault detection and diagnostics relies on having a well-chosen or “informed” feature space with predictable characteristics for recognizing observations that arise outside of initial training data. That is, the accuracy of any classifier depends on the separability of the data. In this context, a separable feature space is one that, regardless of classifier choice, permits classification

of load events, even with future load variability. Since many classification techniques are not readily interpretable, their classification decision is unclear and potentially inaccurate as more data is collected. This is compounded when the underlying distribution of new data is different than the distribution of the training data, known as concept drift. A load may drift to an entirely different region of the feature space than the classifier expected, and may become unseparable from other loads. It is difficult to determine the optimal separator for future data because there are likely many large-margin, low-density separators that can accurately classify the training dataset [92]. That is, concept drift can cause performance degradation, regardless of classifier choice. Load features will change with changing operating conditions, machinery aging and wear, or abnormal load behavior. In evaluating feature spaces for separability, the question becomes: is the data uniquely recognizable, and are there, or will there be, unresolvable overlaps in the feature space now or as equipment ages and operating conditions change? With physically-informed assumptions made on the character of the concept drift, the feature space should be assessed to find the most meaningful set of electrical characteristics that provide adequate separability between classes. The scope of this section assumes the drift in electrical behavior is gradual, analogous to incremental concept drift [21].

Several methods exist to evaluate feature importance in a given feature space. Neighborhood component analysis and minimum redundancy maximum relevance have been used in the context of nonintrusive load monitoring, due to their physical interpretability [93]. Other methods such as random forest variable importance analysis, recursive feature elimination, and Boruta [94] are effective in ranking feature importance. Techniques for evaluating class separability based on distance measures include interclass distance, intraclass distance, the Fisher ratio, and scatter matrices [95,96]. However, these methods do not allow for the possibility that a load's behavior in the feature space will adapt over time.

This section provides a check for electromechanical load separability to inform the decision of adding features to the feature space. The work in this section was in collaboration with Aaron Langham. The separability check seeks to determine if a

classifier trained on a dataset limited in time will be reliable in the future. The separability check does not assume that the data is independent and identically distributed (i.i.d.) or that classes are balanced. The separability check is independent of the type of classifier used, and does not prescribe any particular classification methodology. Using every feature available can increase data separability, but can result in overly fit, non-generalizable decision boundaries due to the sparsity of training data [97]. In addition, adding extra features may incur extra costs if they require additional sensors or signal processing development. Thus, it is desirable to start with a compact set of features and only add features when necessary.

The proposed separability check is based on geometric overlap using hyperellipsoidal regions. An initial feature space is selected based on known electromechanical load behaviors. This feature space is then assessed by applying the load separability check to a limited set of labelled data, which defines regions of load behavior within the feature space. The presence of overlapping regions indicates current or future regions of uncertainty for a potential NILM classifier. As more labelled data is obtained, this check can be periodically run to re-evaluate the feature space for potential conflicts. The proposed load separability check is demonstrated with a dataset of loads collected from USCGC Spencer.

3.3.1 Feature Space Evaluation

The existence of concept drift in load behavior leads to the question: how can feature spaces be evaluated for their ability to provide separability as loads undergo faults or changing operating conditions? With an understanding of load behavior based on prior observations and knowledge of specific operating patterns, a preliminary set of features can be selected. However, prior to selecting and training a classifier and establishing a diagnostic process within this feature space, the selected features should be evaluated. Even if data is linearly separable, the proposed separability check will identify loads that may become ambiguous in the future if drift occurs. This can inform feature space modification. The separability check is given as an overlap test of geometric characterization regions for each load. An ideal characterization region

must be able to represent trends in the load data and anticipated drift, but still be physically reasonable and fairly compact to the region occupied by the load events. Finally, there must be a well-defined and computationally tractable test for overlap between two characterization regions. For illustrative purposes, this method is shown with two-dimensional feature spaces of steady-state real power (P_{ss}) and steady-state reactive power (Q_{ss}). However, this method can be used to assess N -dimensional feature spaces [98].

The features in the presented examples use units of power. However, for features that do not have common units, it is necessary to apply some form of feature scaling such as standardization or normalization [99]. When using the separability check, it is imperative that the dataset in consideration be correctly labelled, such that even points that appear to be outliers can be assumed to represent actual load behavior and not incorrect labelling.

3.3.1.1 Separability Check

For loads that do not drift in the feature space, such as a healthy heater, their distributions in the feature space typically can be characterized as multivariate Gaussians with constant mean and variance. The probability density function (PDF) of a multivariate Gaussian distribution with an N -dimensional random vector \mathbf{X} , is given as:

$$\mathbf{f}_{\mathbf{X}}(x_1, \dots, x_N) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{\Sigma}|}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right), \quad (3.11)$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of \mathbf{X} , $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix, and $\boldsymbol{\mu}$ is the N -dimensional mean vector of \mathbf{X} [100]. The covariance matrix and mean can be approximated with the covariance and mean calculated from collected data. When a load exhibits drift, it often resembles a multivariate Gaussian distribution with non-constant mean and covariance. That is, for some subset of data in time, the data can be approximated as Gaussian. However, over time the distribution may change. With this assumption, the load separability check is performed by first fitting an N -dimensional hyperellipsoid to each load's data. N -dimensional hyperellipsoids

are able to represent variance in several orthogonal axes and are the equiprobability contours of a multivariate Gaussian distribution. A hyperellipsoid region of arbitrary dimension can be represented by:

$$(\mathbf{x} - \mathbf{m})^T \mathbf{E} (\mathbf{x} - \mathbf{m}) \leq 1 \quad (3.12)$$

where \mathbf{m} is an $N \times 1$ vector representing the centroid of the hyperellipsoid and \mathbf{E} is a real symmetric positive-definite $N \times N$ matrix representing the shape and orientation of the hyperellipsoid. Any point \mathbf{x} , that satisfies the inequality in Eq. (3.12) is either inside or on the surface of the hyperellipsoid [101].

In order to generate the parameters of a hyperellipsoid region enclosing a load's range of behavior within the feature space, principal component analysis (PCA) is used to generate a new set of axes from the feature space axes. This expresses the most variance in the data possible in each generated principal component axis [102]. The variance (denoted as σ^2) of the data in each of the principal component axes is computed. The standard deviation (denoted as σ) of the data in each principal component axis is obtained by taking the positive square root of these variances. PCA can be computed using eigendecomposition of the covariance matrix or with singular value decomposition [102]. A component matrix is yielded with rows equal to the principal component axes, which can be used as a "transformation matrix" from the feature axes to the principal component axes. The N -dimensional hyperellipsoid's radii are obtained by multiplying the principal component standard deviations, sorted in descending order, by a user-defined parameter, α . The choice of α is described below. The hyperellipsoid's centroid is obtained by taking the mean of the data points in the feature space. Finally, the hyperellipsoid's rotation is represented using the transformation matrix obtained from PCA. Hyperellipsoidal regions are created in this way for each load in a chosen feature space. These regions are then subjected to a test for overlap, as will be later described. Overlapping regions indicate that a feature space will not be robust at separating the given loads.

One strategy for choosing the hyperellipsoid size involves taking a constant number

of standard deviations in each principal component direction. For example, three standard deviations will cover 99.7% of all hypothetical data, assuming a normal distribution. A disadvantage of this approach, however, is that it is not responsive to the number of load events obtained. Instead, the number of standard deviations for each hyperellipsoid, denoted as α , should be selected based on the amount of confidence in the underlying class distribution. As the number of load events M for a class increases, the confidence that the data is representative of the underlying distribution also generally increases. There is more uncertainty in potential load drift for smaller M . By adjusting α based on class size, this method is able to handle class sizes that are different between loads. Classes that do not have many load events have more uncertainty in their underlying distribution and as a result are assigned a looser hyperellipsoid. The proposed method for selecting α adds an exponentially decaying term to shrink the region size as more load events are collected:

$$\alpha = 3 \cdot (1 + e^{-M/K}), \quad (3.13)$$

where K is a tunable parameter. For any choice of K , the number of standard deviations α , is bounded between three and six. When M is zero, then α is six. When M approaches infinity, the exponential term goes to zero and α approaches three. When α is close to three there is high confidence that the load has been well-characterized and will no longer drift in the feature space.

By adjusting the parameter K , the rate at which the exponential decays can be adjusted. As K increases, the rate of decay of the exponential decreases, and α will converge to three slower. One consideration for the choice of K is the availability of extra features. If there are abundant features that can be added to the feature space, a larger hyperellipsoid (and thus larger K) can be used in order to give a greater chance of detecting problematic drift. There will also be a greater chance of overlapping regions of non-drifting loads. If additional features are easily obtained, these overlaps can be resolved by adding these additional features to the feature space. However, if additional useful features are not available or are expensive to

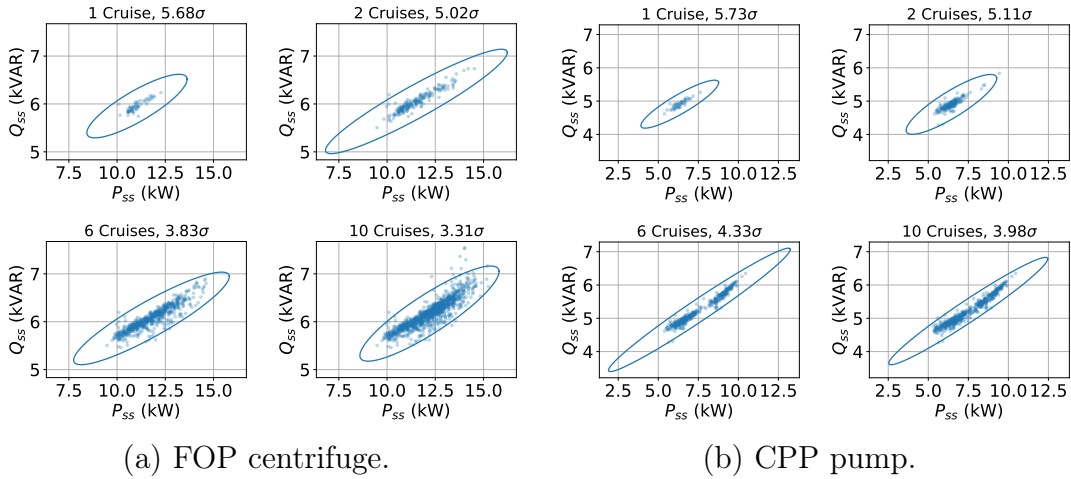


Figure 3-19: Hyperellipsoids in the Q_{ss} versus P_{ss} feature space for two loads at various points in time.

obtain, overlap detection from non-drifting loads should be avoided in order to avoid unnecessarily adding features. Thus, smaller hyperellipsoids (and thus smaller K) should be used. With smaller hyperellipsoids, problematic drift will be detected later than with larger hyperellipsoids.

The two presented shipboard loads from Section 3.2.4 are used as an example. Using $K = 500$ in the Q_{ss} versus P_{ss} feature space, the fuel oil purifier (FOP) centrifuge and controllable pitch propeller (CPP) pump data points and hyperellipsoids are shown in Fig. 3-19a and Fig. 3-19b, respectively. The hyperellipsoids are drawn after various number of cruises to show the evolution of the hyperellipsoid characterization regions as the number of load events increases and as α subsequently decreases. The FOP centrifugal motor starts with 57 labelled load events after the first cruise. Its hyperellipsoid region is created with 5.68 standard deviations (noted as 5.68σ in the figure), which is on the upper end of possible values for α . After ten cruises there are 1131 load events. The new hyperellipsoid region uses 3.31 standard deviations, which is now on the lower end of possible values for α . For the CPP pump, after one and ten cruises, there were 48 and 561 labelled load events, respectively. Thus, the hyperellipsoid regions were created with 5.73 and 3.98 standard deviations, respectively. For both loads, the rotation of the hyperellipsoid is approximately the

same after one cruise as it is after ten cruises, meaning the drift has continued along approximately the same major axis.

3.3.1.2 Test for Overlap

To check for overlap of the hyperellipsoid regions, the approach of [103] is used, in which the overlap condition is developed as a root-counting problem of a convex polynomial $K(\lambda)$. This polynomial represents the hypothetical circumscribing hyperellipsoid appearing in the region between the two hyperellipsoids under consideration. $K(\lambda)$ is generated using the shape and center of both hyperellipsoids, where \mathbf{E} and \mathbf{m} from Eq. (3.12) represent the shape and center, respectively. For two hyperellipsoids with shapes \mathbf{A} and \mathbf{B} , and centers \mathbf{c} and \mathbf{d} , respectively, every point, \mathbf{x} , in the union of both hyperellipsoids must satisfy

$$\lambda(\mathbf{x} - \mathbf{c})^T \mathbf{A}(\mathbf{x} - \mathbf{c}) + (1 - \lambda)(\mathbf{x} - \mathbf{d})^T \mathbf{B}(\mathbf{x} - \mathbf{d}) \leq 1, \quad (3.14)$$

where $\lambda \in [0, 1]$. This inequality is then transformed into the following representation of a hypothetical hyperellipsoid with shape \mathbf{E}_λ and center \mathbf{m}_λ that circumscribes the region of intersection of the two hyperellipsoids:

$$(\mathbf{x} - \mathbf{m}_\lambda)^T \mathbf{E}_\lambda(\mathbf{x} - \mathbf{m}_\lambda) \leq K(\lambda), \quad (3.15)$$

where

$$\mathbf{E}_\lambda = \lambda \mathbf{A} + (1 - \lambda) \mathbf{B}. \quad (3.16)$$

The convex polynomial $K(\lambda)$ is given as:

$$K(\lambda) = 1 - (\mathbf{d} - \mathbf{c})^T \left(\frac{1}{1 - \lambda} \mathbf{B}^{-1} + \frac{1}{\lambda} \mathbf{A}^{-1} \right)^{-1} (\mathbf{d} - \mathbf{c}). \quad (3.17)$$

A necessary and sufficient condition for the two hyperellipsoids to not overlap or touch is for there to be a value of λ in $(0, 1)$ such that $K(\lambda) < 0$. Since $K(\lambda)$ is convex, all that is required is to count the roots of $K(\lambda)$ on $(0, 1)$. If there are two such roots,

the hyperellipsoids do not overlap or touch; otherwise, the hyperellipsoids will either overlap or be tangent. Rather than using symbolic math to solve for the roots of $K(\lambda)$, [103] presents an algorithm to compute $P(\lambda) = \det(\mathbf{E}_\lambda) \cdot K(\lambda)$, which is also convex and has the same roots in $(0, 1)$ as $K(\lambda)$. Using Sturm’s theorem as described in [104], the number of roots of this polynomial in $(0, 1)$ can be efficiently computed. Sturm’s theorem is used to generate a “Sturm sequence” of polynomials using polynomial differentiation and division. By counting the number of sign alternations in these sequences, the number of polynomial roots in the given interval can be computed.

This method works as long as $\mathbf{A} - \mathbf{B}$ is invertible. It is assumed that this will be the case since the radii of two given hyperellipsoids are very unlikely to be exactly the same. With this overlap test defined for two hyperellipsoids, several load characterizations can be checked for overlap two at a time. Detecting overlap between a set of loads will require at most $L(L - 1)/2$ overlap checks, where L is the number of loads. An advantage of the hyperellipsoid region is that its overlap check does not depend on the number of load observations. That is, the overlap test is computed with the N -dimensional hyperellipsoid parameters, not with individual points. As a result, the computational power required for this check does not increase as the number of load events increases.

Fig. 3-20 shows the hyperellipsoid characterization regions for the FOP centrifugal motor and CPP pump from Fig. 3-19a and Fig. 3-19b, respectively. After one cruise, even though the two loads are clearly linearly separable, the hyperellipsoid characterization regions are overlapping. The FOP centrifuge and CPP pump data points have high enough variability for the separability check to anticipate future misclassification of the CPP pump and FOP centrifuge. The hyperellipsoid regions continue to overlap for subsequent cruises.

3.3.2 Separability Check Demonstration

To validate the utility of the separability check on a larger dataset of loads, nine loads from USCGC Spencer are presented, including the FOP centrifuge, FOP feed pump, MPDE jacket water heater, MPDE lube oil heater, MPDE prelube pump, SSDG

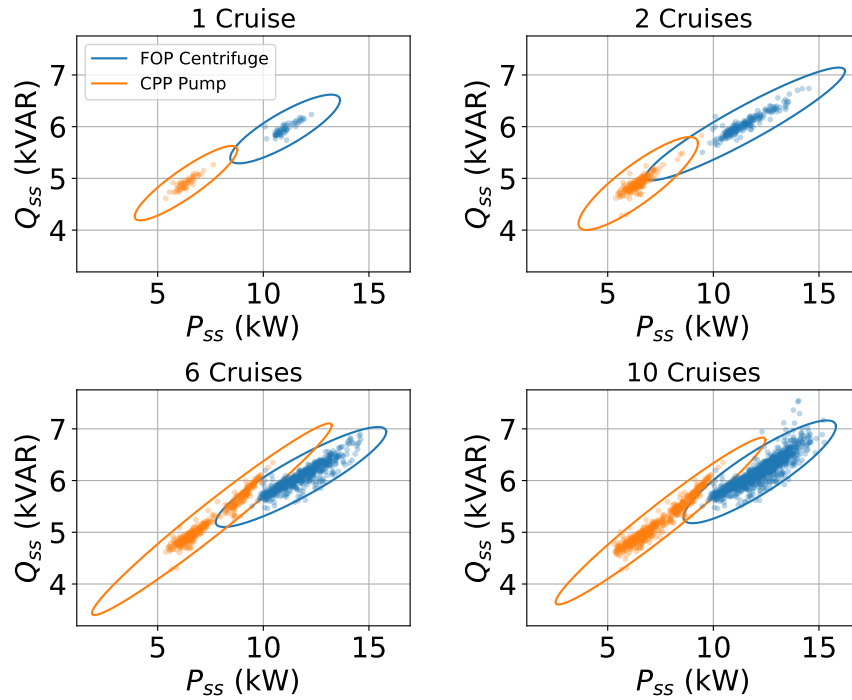
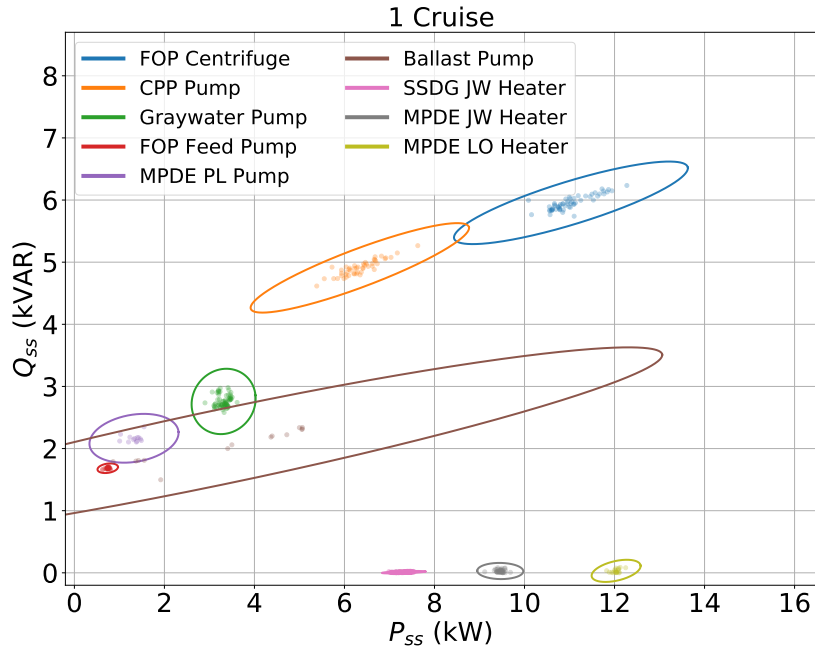


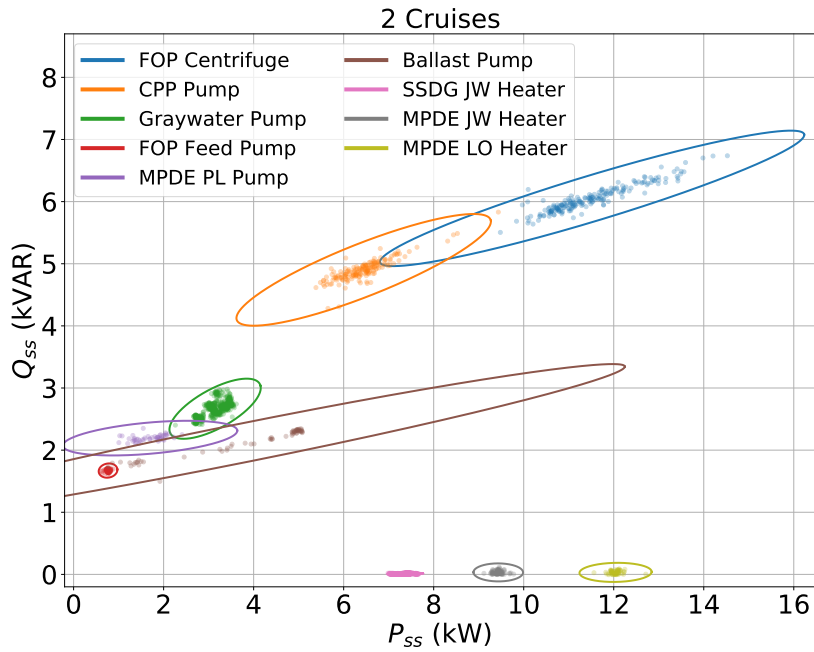
Figure 3-20: Hyperellipsoids in the Q_{ss} versus P_{ss} feature space for the FOP centrifuge and CPP pump at various points in time.

jacket water heater, CPP pump, graywater pump, and bilge and ballast pump. A dataset was assembled from the first two underway cruises of the vessel after installation of the NILM, from August 2016 to March 2017. The hyperellipsoid boundaries are plotted in Fig. 3-21a and Fig. 3-21b for the nine loads after the first cruise and second cruise of data collection, respectively. After only one cruise, the CPP pump overlaps with the FOP centrifuge region, as was previously described. Also after only one cruise, the bilge and ballast pump region overlaps with the graywater pump, MPDE prelube pump, and FOP feed pump regions. This shows that the output of a given classifier is highly unpredictable for these four loads. The drifts of the MPDE prelube pump and graywater pump are not significant enough after one cruise to fail the overlap test. However, after two cruises the ensuing load drift results in the regions overlapping. The three heater loads do not overlap with any other loads, indicating that the feature space provides sufficient separability for these three loads.

Two classifiers were trained with the dataset from the first two cruises to vali-

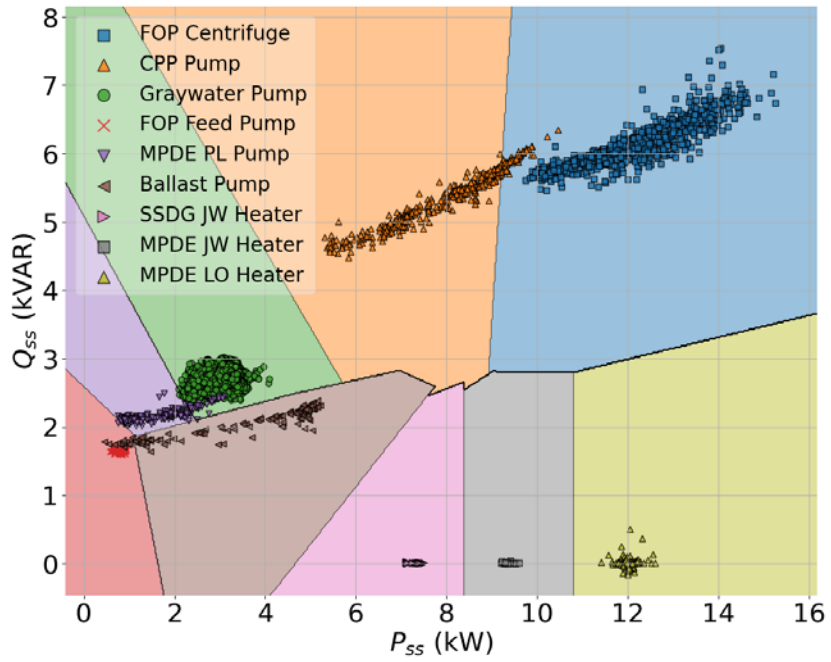


(a) One Cruise

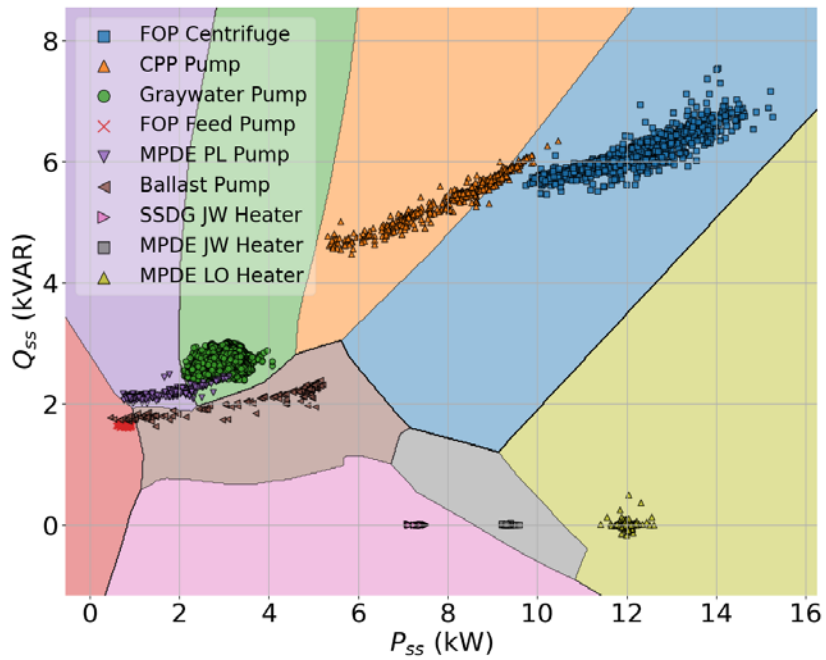


(b) Two Cruises

Figure 3-21: Hyperellipsoids in the Q_{ss} versus P_{ss} feature space for nine loads after one and two cruises of data collection.



(a) Linear SVM.



(b) DNN.

Figure 3-22: Q_{ss} versus P_{ss} boundaries, trained with a linear SVM and DNN using the first two underway cruises of data collection. The data points for eight later cruises are plotted.

date the problematic overlap identified by the separability check. A linear support vector machine (SVM) and deep neural network (DNN) were chosen as examples to demonstrate how concept drift can cause performance degradation for both linear and non-linear classifiers. Different classifiers may have slightly different results. The data was split into 80% training and 20% validation, with data stratification to allocate load events evenly based on class frequency. The trained models had perfect classification accuracy for all load events in the validation dataset for both classifiers. That is, without any consideration for load drift, this would appear to be a sufficient feature space. However, when the model was then tested on the data collected for the eight later cruises (from April 2017 to September 2020), drift of several loads in the feature space resulted in significantly reduced accuracy. Fig. 3-22 shows the decision boundaries (from the models trained on the first two cruises) and the data points from the last eight cruises. Fig. 3-23 shows the normalized confusion matrices for all loads that did not have an perfect accuracy. For both classifiers, as anticipated by the separability check, the CPP pump is often misclassified as the FOP centrifugal motor, the MPDE prelube pump is often classified as the graywater pump, and the bilge and ballast pump is often misclassified as the FOP feed pump. Also as expected from the separability check, the three loads that do not show any degradation in classification performance are the SSDG JW heater, MPDE JW heater, and MPDE LO heater.

After two cruises of data, although the trained classifiers showed that the data was classifiable at the time, the classifiers were not reliable as more data was collected. The problematic drift is anticipated by the separability check. In fact, the hyperellipsoids for two cruises of data and ten cruises of data are relatively similar even with further load drift and with a large difference in number of load events. For example, Fig. 3-24 shows the hyperellipsoids (as solid lines) and data points after ten cruises of data for the six loads that did not have perfect accuracy. Also plotted are the hyperellipsoids (in dashed lines) of the hyperellipsoids after two cruises of data, the same as in Fig. 3-21b. As shown, the drift is relatively well characterized after only two cruises since the hyperellipsoids are drawn with a higher number of standard deviations for fewer load events.

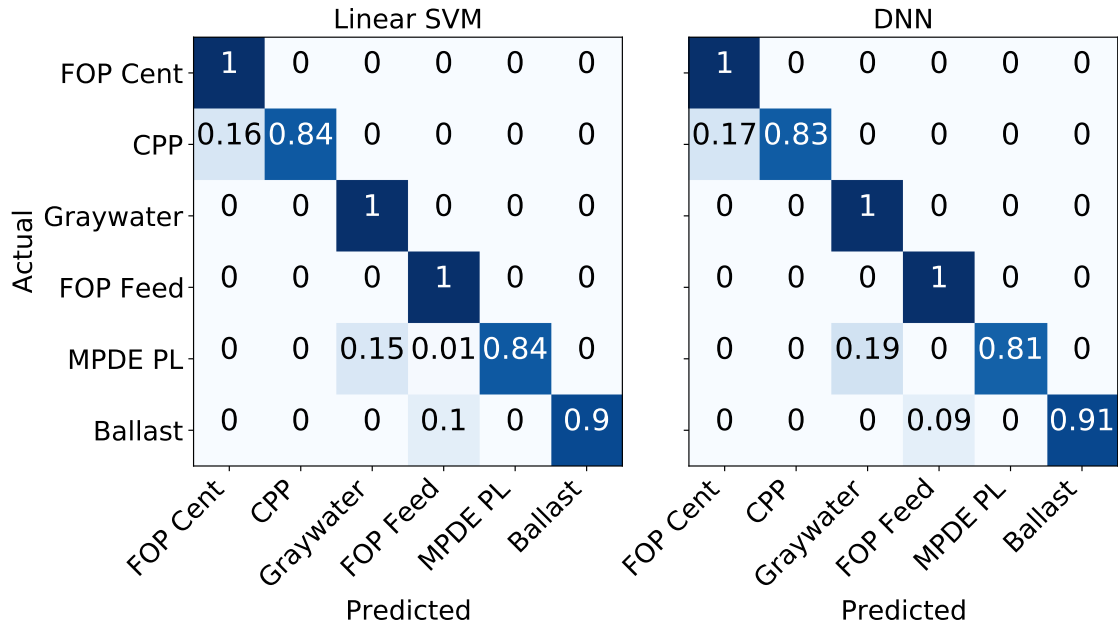


Figure 3-23: Normalized confusion matrices for linear SVM and DNN classifiers trained on the first two cruises of data and tested on the eight later cruises.

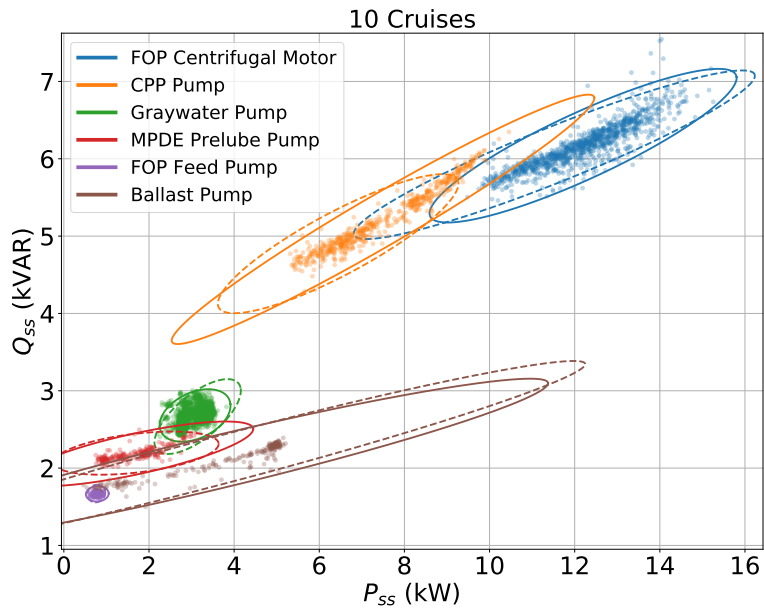


Figure 3-24: Hyperellipsoids (solid) and data points in the Q_{ss} versus P_{ss} feature space for six loads after ten cruises of data collection. Hyperellipsoids (dashed) are also drawn from the first two cruises of data.

3.4 Chapter Summary

This chapter presented the tools for physics-informed feature extraction and selection. A multi-scale framework was presented which separates events into three categories: geometric, statistical, and continuous. Example feature extraction techniques for the three methods were given. The remainder of the chapter focused on transient space characterization and selection for geometric events. Electromechanical load behavior and the resulting geometric features extracted may change due to normal variation, such as inherent mechanical variability or changes in operating condition. Changes in load behavior could also be indicative of an underlying fault or degradation of various internal mechanisms that may require repair or replacement. For a nonintrusive load monitoring classifier, this manifests as concept drift that can reduce classification performance on future data or even make the load completely indistinguishable. Understanding the physics and time-dependency behind changing load behavior can inform feature space selection improvements, enabling the applicability of nonintrusive monitoring for equipment health monitoring and diagnostics. A separability check was presented for ensuring the selection of a physically-informed feature space that allows for load disaggregation, even when loads drift over time. This is especially important for isolated, microgrid, and generation-constrained systems, where system reliability is critical.

Chapter 4

Adaptive Classification and Drift Detection

After selecting a feature space, detected events need to be classified as the correct load. This chapter presents a classification framework for geometric events that adapts to drifting load behavior. The work in this chapter was in collaboration with Aaron Langham and Rebecca Agustin.

Many pattern classifiers fail to account for the time-evolution or dynamic behavior of observed data [105–107]. Concept drift can obliterate the effectiveness of a classifier trained on a static dataset [21, 108]. The order of observations and the relationship between the timing and the evolution of trends contain valuable information. A lack of representative training data in an otherwise tractable domain exacerbates these problems, as data often cannot be assumed to be independent and identically distributed (i.i.d.) or stationary. Decision boundaries in a feature space are frequently not readily explainable or based in physical understanding. Power monitoring of electromechanical loads is an illustrative stream learning problem with practical industrial applications. Effective nonintrusive load monitoring requires accurate load signatures and load identification, i.e., the identification of individual loads from the aggregate power stream.

Nonintrusive identification of geometric events is fundamentally a non-stationary problem exhibiting concept drift. Changes in load behavior can be related to various

forms of concept drift. A load’s electrical behavior may change over time, potentially due to load aging and degradation. As shown in Section 3.2, variability is expected even for a healthy load, arising from normal mechanical variation, changing operating conditions, and environmental factors. A sudden change in machine health is analogous to sudden concept drift. A slow change over time is a form of incremental concept drift. A return to a previous health condition or operating state is similar to recurring concept drift. The challenge in load identification is ensuring correct results even amidst changing operating conditions and fault scenarios. Most nonintrusive load monitoring research assumes training data is forever representative of new data, without regard to changing load behavior [109].

Many machine learning applications, such as image recognition, rely on large, generalizable datasets. The training data for a practical nonintrusive load monitoring classifier will likely need to be collected by a NILM on the system of interest. Nonintrusive load monitoring is therefore an example of few-shot learning [110]. At its extreme, it becomes a one-shot learning problem, in which the model must train starting with only a single example for each load [111]. The few-shot nature of the problem means that the limited training data is not likely to be representative of the load’s long-term operation. As such, deep learning models are prone to overfitting [112].

Many concept drift detection methods use error rates to detect drift using labelled data [113, 114]. This is an unrealistic constraint for real-time nonintrusive load monitoring, as it would require periodic manual labelling of data. Instead, extreme verification latency (i.e., ground-truth labels are never received after the initial dataset) must be assumed. Relatively little work has addressed semi-supervised or unsupervised drift detection and adaptation [21]. An automated solution is desired.

This chapter introduces a multi-level framework of classification techniques applicable to machine learning problems facing the following challenges: 1) concept drift, 2) one-shot or few-shot learning, and 3) extreme verification latency. In order to be robust to outliers, the proposed method is designed to handle incremental and recurring concept drift, whereas sudden concept drift is out of the scope. “Coarse” and “fine” classification levels enhance existing pattern classifiers’ abilities. Given the

scarcity of training data and few-shot nature of the problem, coarse classifiers use extracted features and physically realistic boundaries to avoid overfitting and remain robust to load concept drift. Fine classifiers use higher-dimensional data to resolve any ambiguities in the extracted course classifier feature space. An online clustering algorithm provides drift metrics to enable continual load identification and diagnostics even in the presence of concept drift. These steps work together to ensure confidence in classification, given that ground-truth labels are not available after initialization. Experimental results are presented on real-world, non-stationary data. Power data was collected over four years aboard USCGC Spencer and used for framework evaluation.

4.1 Previously Reported Techniques

Drift detection algorithms can generally be divided into error rate-based and data distribution-based [21, 115]. Ensemble methods incorporate multiple classifiers. Error rate-based approaches focus on tracking some error-related metric and form the largest category of algorithms [21]. Common methods include the Drift Detection Method (DDM) [116] and Early Drift Detection Method (EDDM) [108, 117], which signal that drift has occurred when there is a statistically significant change in the error rate or distance between classification errors, respectively. The drift detection method for online class imbalance (DDM-OCI) [118] uses the reduction in minority-class recall to detect drift, thus staying more robust to class imbalance. Data distribution-based methods use a distance metric to quantify the dissimilarity between the distribution of historical data and new data [21, 115]. A drift is signaled when the dissimilarity is proven to be statistically significant.

All error rate-based methods and most data distribution-based methods are supervised approaches which assume that an instance’s ground-truth class label is available immediately after prediction. However, this is an unrealistic constraint in many real-world applications. Labelling data is often a costly process that involves manual labelling by domain experts. With this motivation, some semi-supervised and unsu-

pervised drift detection algorithms have been proposed [113]. One semi-supervised approach is the Semi-Supervised Adaptive Novel Class Detection and Classification (SAND) algorithm [119]. This method bases drift detection on classifier confidence. When drift is detected, a new model is created, using predicted labels for instances with high confidence. True labels are requested for instances with low classifier confidence. Another approach is the Dynamic Selection Drift Detector algorithm which bases drift detection on a pseudo-error rate [120]. It requests true labels for instances that are near a set warning level. Both of these methods request labels for a subset of data after concept drift is detected.

For nonintrusive load monitoring, it is not realistic to assume that even a subset of labelled data will be available when requested. Extreme verification latency must be assumed. A semi-supervised framework that handles drifting environments with extreme verification latency is proposed in [121], referred to as Compacted Object Sample Extraction (COMPOSE). It uses a base classifier trained on the labelled data at the initial step, then extracts core supports from the classified data to retrain the classifier. The core supports represent the geometric center of each class distribution to serve as labelled instances. Different core support shapes have been proposed such as an α -shape [121] and Gaussian mixture models [122], while in [123], all the classified samples are used instead of core support extraction. For affinity-based COMPOSE [124], an affinity matrix is formed between the labelled and unlabelled samples and those with high similarity scores are classified and used to retrain the classifier. While these methods do address extreme verification latency, they are not applicable for one-shot or few-shot learning. The methods assume a good base classifier without mentioning how to choose such a classifier. The choice of base classifier and its resulting decision boundaries has a large affect on any subsequent classification, especially if it overfits the limited initial data. To the best of our knowledge, there is no work in literature that simultaneously addresses incremental and recurring concept drift, one-shot or few-shot learning, and extreme verification latency.

4.2 Classification and Drift Detection

The framework consists of four steps for classifying and detecting drift. This multi-level framework ensures a high level of confidence in labelled events since ground-truth labels are assumed to not be available. Events that do not pass the multi-level check are determined unclassifiable. A one-vs-all check, referred to as the “preliminary” check, is used first as a “negative” classifier that eliminates the classification of physically implausible events. This first step appeals to the physical expectations for load behavior to ensure physically realistic boundaries. Next, classification operates with two levels of granularity. The two levels are designed to improve classification while gradually increasing dimensionality as needed. The first, “coarse” level examines high-level features extracted from the waveform (e.g., features such as peak and steady-state power) to avoid overfitting due to the few-shot nature of the problem. The second, “fine” classification step uses a more in-depth examination of sampled data (e.g., time-domain shape recognition). That is, the coarse classifier uses extracted features from the power waveform, whereas the fine classifier uses windows of the time-domain power waveform directly. Finally, load drift is detected and tracked using “drift clusters” to characterize evolving load behavior and concept drift. The concept of an “exemplar” is defined as a load event that is representative of a load’s short-term behavior. Each load has an “initial exemplar,” i.e., the initial load event. Each load also has an “active exemplar,” which is updated to represent the most recent operation state based on the drift clusters.

4.2.1 Framework Overview

The framework process is presented in Algorithm 4.1. To initialize the framework, a list of all load classes is obtained and denoted as L . Initial data is collected and the initial exemplar is set for each load to be the load’s first event. During the initialization step, if there is only a single event for each load, the active exemplar is set to be the initial exemplar. Otherwise, the drift clustering algorithm is run on the load’s initial data to determine the active exemplar. Then, the preliminary and

Algorithm 4.1 Algorithm for organizing classifiers.

```
1:  $L \leftarrow$  list all loads
2: for each  $l \in L$  do
3:   Set initial exemplar
4:   Set active exemplar
5:   Set up preliminary and coarse boundaries
6: end for
7:
8: while True do
9:    $x \leftarrow$  incoming feature vector
10:
11:    $M \leftarrow$  PreliminaryCheck( $x, L$ )
12:   if  $M$  is empty then
13:     Determine  $x$  as unclassifiable
14:     continue
15:   end if
16:
17:    $N \leftarrow$  CoarseClassifier( $x, M$ )
18:   if  $N$  contains only one load then
19:     Classify  $x$  as the load in  $N$ 
20:     TrackDrift( $x$ )
21:     continue
22:   end if
23:
24:    $P \leftarrow$  FineClassifier( $x, N$ )
25:   if  $P$  contains only one load then
26:     Classify  $x$  as the load in  $P$ 
27:     TrackDrift( $x$ )
28:     continue
29:   end if
30:   Determine  $x$  as unclassifiable
31: end while
32:
33: function TrackDrift( $x$ )
34:   Update drift clusters with  $x$ 
35:   Update active exemplar
36:   Update preliminary and coarse boundaries
37: end function
```

coarse boundaries are drawn in the feature space. When features that do not have common units are used in this framework, min-max normalization is performed so that Euclidean distances in the feature space are well-defined. For each feature axis,

i , the range of the data is transformed into $[0, 1]$ through the transformation,

$$x_s^i = \frac{x^i - x_{min}^i}{x_{max}^i - x_{min}^i}. \quad (4.1)$$

Min-max parameters are obtained using the initialization data. As a result, it is possible that incoming data may be scaled outside $[0, 1]$.

Once initialization is complete, every incoming feature vector is initially examined via the preliminary check. A list of loads that pass the check is generated and denoted as M . There are two high-level possibilities that could occur: 1) M is empty, indicating the feature vector falls outside any known load preliminary boundaries, and 2) M contains at least one load. If M is empty, the event is considered unclassifiable. That is, the event does not go through the remaining checks, and the procedure moves on to the next incoming feature vector. This is represented in Algorithm 4.1 with the continue statement. If M is not empty, the loads that passed the preliminary check are passed to the coarse classifier.

With a power sampling rate of 60 Hz, a transient can easily have a dimensionality on the order of several hundreds. The few-shot nature of the problem implies that working with such a high-dimensionality feature space may lead to overfitting, also known as the curse of dimensionality [97]. Thus, the coarse classifier operates on the lower-dimensionality feature space of extracted features. The coarse level performs classification using one-vs-all classifiers for each load. This permits overlapping load decision boundaries and allows for the possibility that a load's decision boundaries can change over time independent of other loads. The coarse classifier returns a list of loads denoted as N . There are three possible outcomes for the coarse classifier: 1) N is empty, indicating that the feature vector falls outside any known load coarse boundaries (but still within a preliminary boundary), 2) N contains exactly one load, and 3) N contains more than one load, indicating overlap of the coarse boundaries. When a feature vector is inside a single load coarse boundary, it is classified as that load without running the fine classifier. If N contains zero loads, the fine classifier is run on the M loads that passed the preliminary check. If N contains more than one

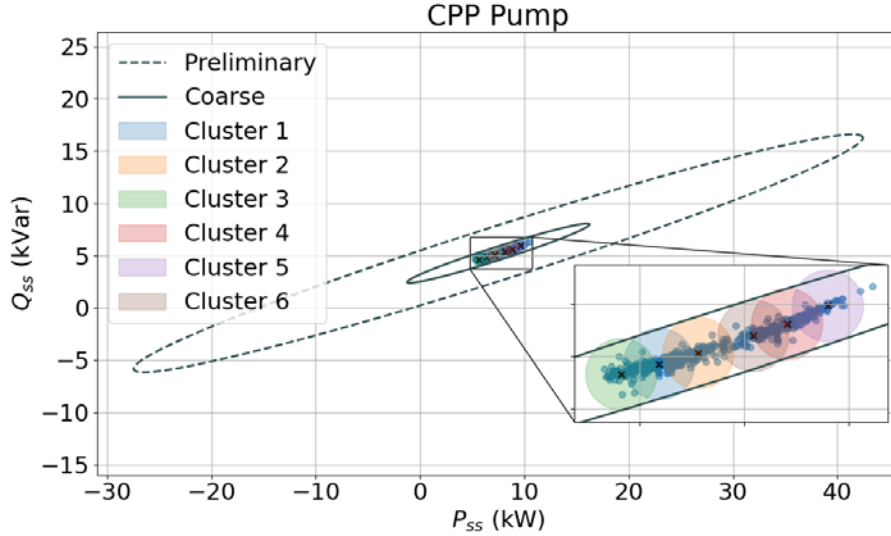


Figure 4-1: CPP pump preliminary boundary, coarse boundary, and drift clusters for the Q_{ss} versus P_{ss} feature space.

load, the fine classifier is run on those N loads.

The fine classifier uses the higher-dimensionality transients if the coarse classifier is unable to confidently identify one load. The fine classifier returns a list P , which either contains a single load or is empty. If P contains a single load, the event can be classified as that load. Otherwise, the event is considered unclassifiable. When an event is classified, a clustering algorithm is run using geometric distances in an easily-conceivable extracted feature space. This makes possible the tracking of drifting power signatures and designation of the active exemplar. The preliminary boundaries, coarse classifiers, and fine classifiers are adapted as necessary to track diagnostic changes and ensure accurate load recognition. Example preliminary boundaries, coarse boundaries, and drift clusters are shown in a two-dimensional feature space for two shipboard loads, the controllable pitch propeller (CPP) pump and graywater pump, in Fig. 4-1 and Fig. 4-2, respectively.

4.2.2 Preliminary Check

Decision boundaries created by classifiers are not guaranteed to be compact or physically realistic. Thus, it is necessary to establish which loads an incoming feature

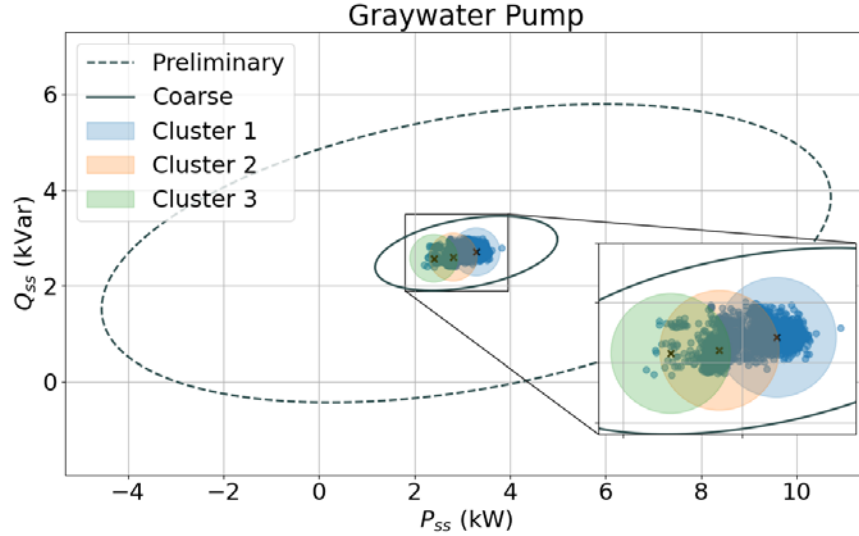


Figure 4-2: Graywater pump preliminary boundary, coarse boundary, and drift clusters for the Q_{ss} versus P_{ss} feature space.

vector could plausibly belong to before attempting classification. For the domain of power monitoring and incremental concept drift, it is reasonable to assume that features that are physically relevant to the load’s drift will change gradually. As a result, an extracted feature space-based “preliminary check” rules out loads that are physically implausible candidates for a given load event. An N -dimensional hyperellipsoid boundary is created for each load in the feature space, where N is the number of dimensions of the feature space. The hyperellipsoid can represent a spread of points with a small number of parameters [100]. As demonstrated in Section 3.3, a hyperellipsoidal region can be represented by

$$(\mathbf{x} - \mathbf{m})^T \mathbf{E} (\mathbf{x} - \mathbf{m}) \leq 1, \quad (4.2)$$

where \mathbf{m} is an $N \times 1$ vector representing the centroid of the hyperellipsoid and \mathbf{E} is a real symmetric positive-definite $N \times N$ matrix representing the shape and orientation of the hyperellipsoid. For the preliminary check, any point \mathbf{x} , satisfying the inequality in Eq. (3.12), is either inside or on the surface of the hyperellipsoid [101]. Estimates of variance in observed load behavior can define a loose hyperellipsoidal boundary where load observations can be expected to be located with high confidence. Using

principal component analysis (PCA) [102], the variance of the data can be obtained in each of the principal component directions, as well as the mean in each feature dimension. PCA also yields a “transformation matrix” whose rows are each principal component axis. The standard deviations of the data in each principal component axis are obtained as the positive square root of the variances. A tunable number of standard deviations, denoted here as A , is chosen by the user to generate the radii for the hyperellipsoid. For the USCG vessel, $A = 28$ standard deviations has proven effective for construction of hyperellipsoidal boundaries for the preliminary check.

Since PCA cannot be reliably used with few data points, the implementation does not set up a given load’s PCA hyperellipsoid until 10 events have been recorded for the load. Before then, a provisional preliminary boundary is used, consisting of an unrotated N -dimensional hyperellipsoid whose radius in axis i is equal to $\max(0.2, x_i)$, where x_i is the initial feature vector’s value in that axis (using min-max normalization). By setting a minimum axis value of 0.2, events close to a feature axis (e.g., heaters that consume no reactive power) are not assigned impractically small preliminary boundaries.

Only loads whose preliminary boundaries contain the incoming event’s feature vector are considered as candidate loads for the following coarse classifier. If no loads pass the preliminary check, then the load event is considered unclassifiable. Every time a new event is classified to the load, the load’s preliminary boundary fits a new PCA hyperellipsoid to the load data.

4.2.3 Coarse Classifier

Coarse boundaries drawn in the feature space can further reduce the number of candidate loads and potentially perform final classification. Since drift in the feature space can result in load feature vectors occupying the same general space, the coarse classifier is chosen to be a one-vs-all rather than multiclass classifier. Once a list of candidate loads has been obtained from the preliminary check, each candidate load’s coarse boundary is checked as to whether it contains the incoming feature vector. If only one candidate load’s boundary contains this feature vector, the incoming event

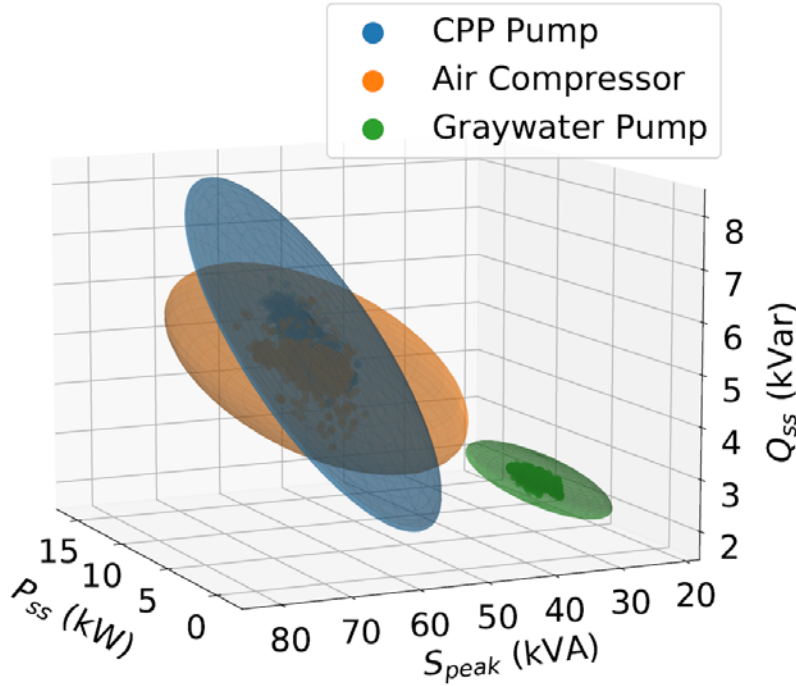


Figure 4-3: Coarse boundaries drawn in the P_{ss} , S_{peak} and Q_{ss} feature space for the CPP pump, air compressor, and graywater pump.

is classified to that load. If multiple loads' coarse boundaries contain this feature vector, then there is not yet sufficient information to classify the event. These containing loads are retained as candidate loads for the fine classifier. If no loads' coarse boundaries contain the feature vector, then no classification is made, and the same list of candidate loads are used as the input for the fine classifier.

The coarse classifier should be chosen such that it yields a binary result. Classifiers such as deep neural networks (DNN) and random forests (RF) provide nonlinear boundaries that can take on complicated shapes. Another option is to use the N -dimensional standard deviation hyperellipsoid described earlier, but with a much smaller number of standard deviations. This is the method demonstrated in this work, using 7 standard deviations. Eq. (3.12) is used to determine which hyperellipsoids contain a given feature vector. This method has the advantage that the hyperellipsoid can likely be obtained faster than a DNN or RF can be trained, and only uses data from the load in question (as opposed to the DNN and RF, which require other load data to train for binary classification).

Just as for the preliminary boundaries, some amount of data must be collected until a coarse classifier can be trained. Until 10 events have been recorded for a load, the same technique used for the provisional preliminary boundaries is used, but here the radii of the N -dimensional hyperellipsoids are set to $\max(0.05, 0.25 \cdot x_i)$ (using min-max normalization). Fig. 4-3 shows an example three-dimensional feature space with three shipboard loads and hyperellipsoidal coarse boundaries for demonstration. If an event is within only a single load coarse boundary, such as a point inside the graywater pump coarse boundary, it can be classified as that load. If an event is within multiple load coarse boundaries, such as a point that is inside both the CPP pump and air compressor coarse boundaries, that event goes to the fine classifier. If an event is not within any coarse boundaries, but is still within a preliminary boundary, it also goes to the next and final classifier. Using hyperellipsoids as coarse classifiers involves fitting a new PCA hyperellipsoid to the load data every new event for that load.

4.2.4 Fine Classifier

For feature vectors that were in multiple coarse load boundaries or were not in any coarse load boundaries, but were within at least one preliminary load boundary, a multiclass fine classifier is run as the final classification step. The fine classifier operates on higher-dimensional data than the coarse classifier in order to resolve overlap in the extracted feature space. The fine classifier relies on having an accurate representation of recent load operation through the active exemplar and drift clusters. This work uses six seconds of the real and reactive power time-domain transients associated with the incoming event as the fine classifier feature space. For fine, time-series classifiers, this work demonstrates both a correlation matching algorithm [5] and gated recurrent units (GRUs) [125].

Using a correlation matching algorithm requires only a single exemplar transient for each load, i.e., that of the active exemplar. This method can be used for one-shot learning, in which only a single event is available for each load. The initial exemplar is used as the active exemplar until the drift clustering algorithm identifies

a new active exemplar. To resolve a set of loads using the correlation matching algorithm, the incoming load event’s transient is matched to each of the candidate loads’ active exemplars’ transients, and a correlation score (from Eqs. (3.1)-(3.4)) is generated for each. When the correlation score from (3.4) approaches zero, this indicates that the exemplar and observation transients match in both shape and amplitude. The minimum correlation score for the candidate loads is found. If it is less than a tunable threshold, the event is classified to the corresponding load. For events within multiple coarse boundaries, the threshold is 0.25. If an event is not within any coarse boundaries and the fine classifier uses candidate loads from the preliminary check, a lower maximum correlation score of 0.10 is used, since there was originally less confidence in the candidates.

When using a GRU, the initial training cannot be done on a single event. For these classifiers, it is assumed that there is sufficient data to train an initial classifier. As shown in Section 4.3, an adaptive GRU approach is demonstrated for this work. The initial multiclass model is implemented with a GRU layer with 50 ReLU-activated neurons, then a densely connected layer with 30 ReLU-activated neurons, and finally a softmax-activated layer. The training approach used Adam-optimized backpropagation with a learning rate of 0.001 and categorical crossentropy as the loss function. The initial data was split into 80% training and 20% validation with data stratification. A batch size of 64 was used for mini-batch gradient descent. Validation loss was used for early stopping, such that after fifteen epochs of no significant improvement, training was stopped.

After an initial multiclass GRU model is trained, the model must be incrementally retrained in order to learn the detected load drift. However, models retrained using only recent data often suffer from “catastrophic forgetting,” that is, the degradation of performance on previous tasks when learning new tasks [105, 126]. One approach to prevent catastrophic forgetting is the replay-based or rehearsal approach, in which some of the previous samples are stored and repeatedly reused when the model is retraining on new data [105, 127, 128]. Another approach is regularization-based, in which a penalty consolidates the important weights, and selectively slows down

learning on those weights [129]. The third approach is dynamic architecture-based, which iteratively updates the network architecture by network masking or network pruning [130]. In this work, the first method is applied, in which a memory buffer is used to store a limited number of transients. For each load, the thirty most recent transients are stored in a temporary memory buffer. Every time a new drift cluster is formed for a given load, the twenty most recent transients are stored in a permanent memory buffer. This is analogous to storing the active exemplar for the correlation matching technique, assuming that the load drift is incremental. Every time a new drift cluster is formed, backpropagation is run on the model using every load’s temporary and permanent memory buffer of transients. A learning rate of 0.0001 is used. This is smaller than the learning rate of the initial classifier to ensure only incremental improvements are being made. The same data stratification split is used as for the initial classifier. Training is stopped after five epochs of no improvement in the validation loss.

For each load the number of transients stored is at most $30 + 20 \cdot x$, where x is the number of drift clusters for that load. Using the same size temporary memory buffer for each load helps prevent biasing the classifier towards the class with the most abundant number of recent events. A GRU classifier uses the output score of the softmax output layer associated with each candidate load to perform classification. Specifically, for the list of candidate loads in multiple coarse boundaries, the event is classified as the load with the maximum softmax output score. If an event is not within any coarse boundaries, the event is classified with the load with the maximum softmax output score, as long as the score is greater than 0.5.

4.2.5 Drift Clusters

Over time, as a load ages or operating conditions change, load observations may drift away from the initial exemplar, for both the coarse classifier and the fine classifier. Keeping track of load concept drift and distribution changes serves an important role for classification. It is especially important due to the few-shot nature of the problem, as the initial dataset likely does not capture a load’s possible drift. In

Algorithm 4.2 Algorithm for drift clustering.

Input: r_0 : micro-cluster radius

Input: γ : density threshold

Output: ST: short-term drift metric

Output: LT: long-term drift metric

```
1:  $E \leftarrow$  incoming event
2:  $d_{min} \leftarrow$  distance of  $E$  to closest micro-cluster center
3: if  $d_{min} \leq r_0$  then
4:   Add  $E$  to nearest micro-cluster
5:   Set active exemplar as nearest micro-cluster exemplar
6:   if  $d_{min} \leq \frac{r_0}{2}$  then
7:     Update micro-cluster center
8:   end if
9: else
10:  Add  $E$  to outliers
11:   $y \leftarrow$  outlier with most outliers within distance  $r_0$ 
12:   $Z \leftarrow$  outliers within distance  $r_0$  of  $y$ 
13:  if  $Z$  contains at least  $\gamma$  points then
14:    New micro-cluster  $\leftarrow Z$ 
15:    New micro-cluster center  $\leftarrow$  mean of  $Z$ 
16:    New micro-cluster exemplar  $\leftarrow E$ 
17:    Set  $E$  as active exemplar
18:  end if
19: end if
20:  $ST \leftarrow$  distance from  $E$  to active exemplar
21:  $LT \leftarrow$  distance from  $E$  to initial exemplar
22: return ST, LT
```

the framework, drift is tracked by designating certain load events as representative exemplars. Furthermore, the distance in the feature space between an observation and an exemplar serves as an important metric for diagnostics [131].

A density-based online clustering algorithm is used in order to determine the representative exemplars. Example density-based clustering algorithms include clustering online data in arbitrary shapes clusters (CODAS) and clustering evolving data streams into arbitrary shapes (CEDAS) [132, 133]. These algorithms both use the concept of a micro-cluster, which is formed based on local density. In this context, each micro-cluster represents a load drift cluster. When a new micro-cluster is formed, a new exemplar is designated to represent it.

The density-based clustering can be focused on the features that are most signif-

icant for recognizing load drift. As an illustrative example, steady-state power level changes (P_{ss} and Q_{ss}) often indicate an underlying gradual shift in load performance. Since these features both use power units, min-max normalization is not used for this clustering process. The clustering and drift tracking process is summarized in Algorithm 4.2. Each load's initial exemplar is set as its first event. Until a load's first drift cluster is formed, the load's active exemplar is its initial exemplar. After every new data sample labelled by the preliminary check, coarse classifier, and fine classifier, as outlined above, the clustering algorithm is run. A micro-cluster is formed when an area in the feature space reaches a fixed density threshold, γ , in this work fixed to five points for all loads. The density threshold should be chosen based on an estimate for when events can be deemed a repeatable occurrence. For example, in power monitoring, a single event would not be enough to declare a new drift state. If a load has energized five times in the same area in the feature space, it is more likely to be a repeatable event. The micro-cluster radius, r_0 is calculated for each load as $\max(0.5 \text{ kW}, P_{ss}/10)$, where P_{ss} is the steady-state real power of the first load event. The micro-cluster radius should be chosen based on expected normal variation. For power monitoring, we set a minimum radius of 0.5 kW so that loads with small steady-state values are not assigned impractically small micro-cluster radii. For larger loads, we select 10% of the steady-state value to represent incremental drift. When the criteria for creating a micro-cluster are met, the most recent event is selected as the exemplar to represent the load state for events that belong to the micro-cluster. For example, Fig. 4-4 shows the drift clusters and representative exemplars for the port-side CPP pump on USCGC Spencer. The numbering of the drift clusters correspond to the chronological order in which they were formed.

In the original CEDAS algorithm, a decay rate is used to remove defunct micro-clusters. However, in our modified approach, each of the previously active micro-clusters is maintained. This is necessary for continued load diagnosis and to take into account recurring concept drift. That is, the micro-clusters are never removed, merged, or split once identified. The micro-cluster center is updated when a new event is added that is within half the radius of the micro-cluster. In addition to

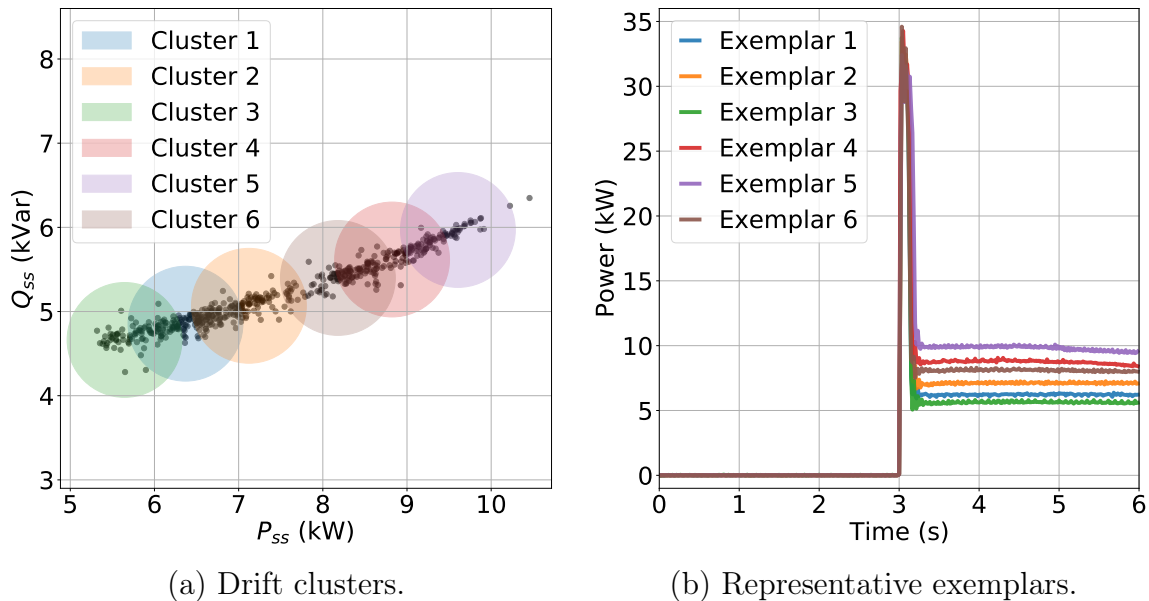


Figure 4-4: CPP pump drift clusters and representative exemplars.

keeping all previous active exemplars, a short-term and long-term distance metric are calculated, where the distance is defined as the Euclidean distance within the steady-state feature space. The short-term metric is the distance from an incoming event’s feature vector to the active exemplar’s feature vector and should remain small for incremental drift. The long-term metric is the distance from the incoming event’s feature vector to the initial exemplar’s feature vector. This metric is useful for load diagnostics, as an increasing long-term drift metric provides indication of possible load degradation. These drift metrics are demonstrated in Section 4.4.

4.2.6 Unclassifiable Loads

The framework is designed with the assumption that ground-truth labels will never be available. The four-step process ensures high confidence in those that are classified, in order to prevent the cascading result of an incorrect label. There are several underlying reasons for why an event could be unclassifiable. First, it could simply be the result of an imperfect event detector, and the detected event may not correspond to an actual load event. It is important that the model not use these events, so as to

not be impacted by noisy data. Second, if a feature vector is in multiple coarse load boundaries and is further unidentifiable by the fine classifier, it likely means both the extracted feature space and transient features need more dimensions in order to increase separability. Third, the event could be an instance of a new or unknown load class. Although out of the scope of this work to automatically identify a new load class, a human operator can identify and label a new load that was added to the system, or that was not accounted for during the NILM installation. The new load would then be added to the load list, L , and initialized as described in Algorithm 4.1 by setting the initial and active exemplars and preliminary and coarse boundaries for the load. When using adaptive correlation matching as the fine classifier, drift clusters can be used to identify the exemplar for correlation matching. When using an adaptive GRU, the GRU can be incrementally trained with the new labelled data. Finally, it is possible that the unclassifiable load is the result of a load experiencing a sudden fault leading to a sudden concept drift. If the fault causes its events to be outside the load’s feature space boundaries and also unrecognizable by the fine classifier, it is an instance of sudden concept drift, and operator supervision will again be necessary to identify the faulty load.

4.3 Experimental Results

The framework was tested on a nonintrusive load monitoring dataset containing observations from three sub-panels on USCGC Spencer: the port-side engine room sub-panel, starboard-side engine room sub-panel, and auxiliary room sub-panel. Data was collected over the course of four years, with gaps in the data during some of the in-port periods. Many of the loads are part of larger controlled systems, including the port-side main propulsion diesel engine (MPDE) keep-warm system, the port-side ship service diesel generator (SSDG) keep-warm system, and the fuel oil purifier (FOP) system. The remaining loads are additional engine room loads and auxiliary room loads. Table 4.1 lists the individual loads, their rated powers, and the number of events in the dataset. For the graywater pump, a random subset of 2000 events

Table 4.1: Loads from USCGC Spencer in dataset.

System	Equipment	Rated Power	# of Events
<i>Port-side MPDE keep-warm system</i>	Jacket water heater	9.0 kW	117
	Lube oil heater	12.0 kW	278
	Prelube pump	2.2 kW	242
<i>Port-side SSDG keep-warm system</i>	Jacket water heater	7.5 kW	1178
	Lube oil heater	1.3 kW	2732
<i>Fuel oil purifier system</i>	Centrifugal motor	9.5 kW	1131
	Feed pump	2.6 kW	341
<i>Additional engine room loads</i>	Controllable pitch propeller pump	7.5 kW	561
	Graywater pump	3.7 kW	2000
	Bilge and ballast pump	5.6 kW	213
<i>Auxiliary room loads</i>	Air compressor	7.5 kW	1390
	Air conditioner	17.0 kW	74

from the dataset was used to keep the event count on the same order as the remaining loads. Fig. 4-5 shows the initial time-domain transients for each of the loads. Since the data collection for the auxiliary room panel was started later than the port and starboard sub-panels, the data was aligned such that the start of data for each sub-panel was set to $t = 0$.

The extracted features for the preliminary check and coarse classifier include the real power steady-state (P_{ss}), reactive power steady-state (Q_{ss}), the maximum apparent power at inrush (S_{peak}), and transient time. Changes in steady-state levels are calculated as the difference between the median values over 0.5 s windows before and after an identified event. The maximum power at inrush is defined as the difference between the maximum value of the transient and the median value of a 0.5 s window before the transient. For transient time, first a ten-point rolling mean is applied to the first difference stream of apparent power. If there is a first difference of -500 W or less, indicating a large negative slope, the steady state is determined to be after this value. Then, if the rolling mean of the first difference stream is less than 5 W, it indicates that the apparent power stream has reached steady state. The feature axes are normalized with min-max normalization. The fine classifier features are six seconds of the P and Q waveforms, centered at the detected transient.

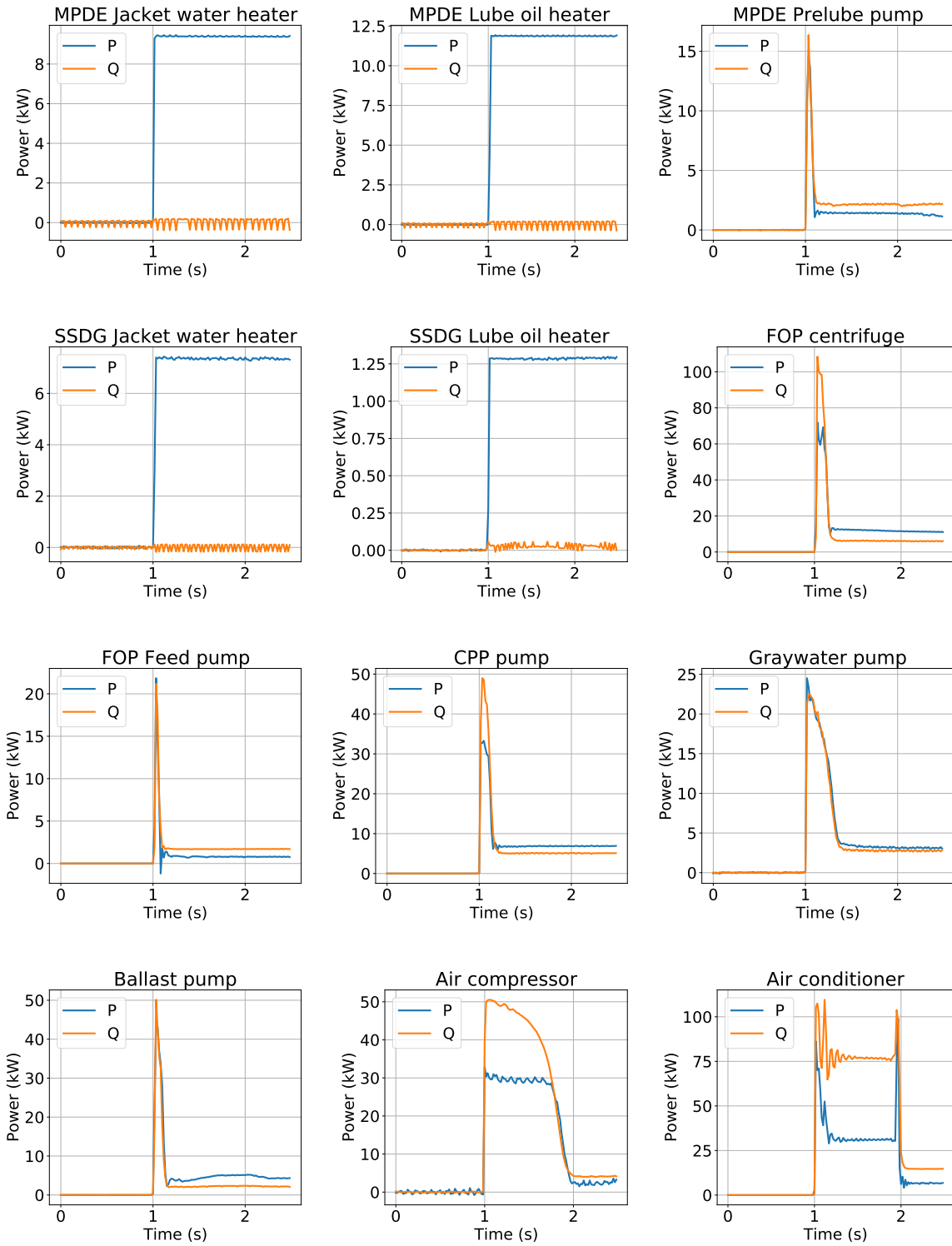
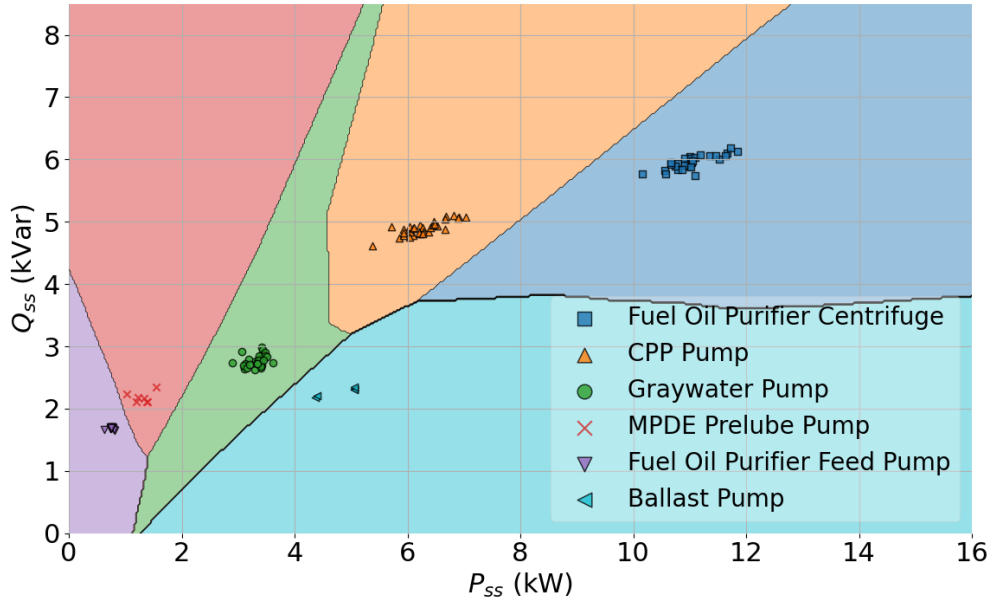


Figure 4-5: Zoomed-in view of the initial time-domain transients for each of the loads in the dataset.

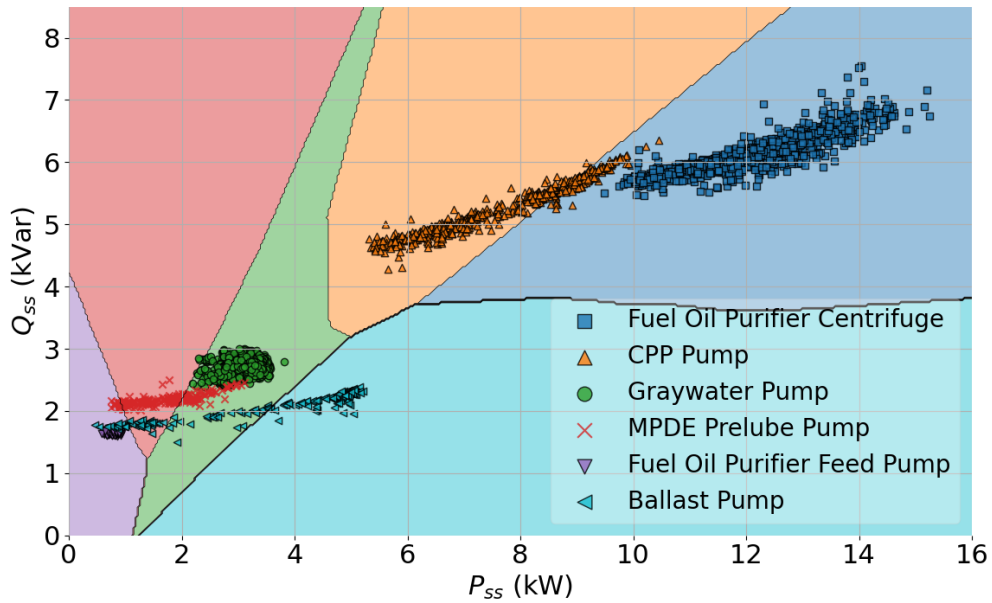
4.3.1 Two-Dimensional Static Classifier

First, a subset of these loads are used in a two-dimensional feature space of P_{ss} and Q_{ss} as illustration of the reduced performance of a static classifier due to changing load behavior over time. A deep neural network (DNN) is trained and used as a classifier for six loads from USCGC Spencer. A limited dataset is used for training consisting of the first month of data after installation of the NILM. This is a practical scenario for nonintrusive load monitoring classifiers, as algorithms may need to be trained on data as it arrives in time. Ideally, the end user will not need to wait for years of data collection before the utility of NILM can be realized for energy scorekeeping and fault detection. The DNN was implemented with two hidden layers of 50 and then 30 ReLU-activated neurons, followed by a softmax-activated layer. The DNN was trained with Adam-optimized backpropagation and categorical crossentropy as the loss function. The data was split into 80% training and 20% validation with data stratification. Mini-batch gradient descent was used with a batch size of 64. The validation loss was used as a stopping criterion, such that training was stopped after fifteen epochs in which the validation loss did not significantly improve.

For these six loads, the fuel oil purifier (FOP) centrifuge, controllable pitch propeller (CPP) pump, graywater pump, FOP feed pump, main propulsion diesel engine (MPDE) prelube pump, and bilge and ballast pump, the trained model had perfect classification for all load events in the validation dataset. The limited set and the resulting decision boundaries are shown in Fig. 4-6a. When the model was tested on the rest of the data collected on the USCG vessel (from October 2016 through September 2020) there was significantly reduced performance due to the drifting in the feature space of several loads. Fig. 4-6b shows the decision boundaries from the model trained on the training data with the data points from October 2016 through September 2020 plotted. The CPP pump has drifted into the FOP classification region, the MPDE PL pump has drifted into the FOP feed pump and graywater pump classification regions, and the bilge and ballast pump has drifted into the graywater pump, MPDE prelube pump, and FOP feed pump classification regions. This perfor-



(a) Limited dataset plotted.



(b) Full dataset plotted.

Figure 4-6: Q_{ss} versus P_{ss} decision boundaries, trained on a limited dataset.

mance degradation is due to the fact that with limited data, there is usually more than one large-margin low-density separator that can accurately classify the load points. There could be many separators that are consistent with the limited labelled data, but are very diverse with respect to the feature space [134]. It is difficult to determine

the optimal separator based on the limited data alone.

4.3.2 Framework Verification

Two main scenarios are tested with the adaptive framework. In the first, classification with the framework starts with only a single event for each load, also known as one-shot learning [111]. In the second scenario, classification with the framework begins after one month of data has been collected by the NILM and all events have been hand-labelled. To use the framework in the first scenario, with only a single exemplar for each load, provisional preliminary and coarse boundaries are implemented in the four-dimensional feature space, as described in Section 4.2. Each load’s initial exemplar is set to be that load’s active exemplar for the correlation matching fine classifier, to start with. For each load, after 10 events have been classified, the preliminary and coarse boundaries are updated by fitting PCA hyperellipsoids to the classified load data. Algorithm 4.2 is used to identify and update the active exemplar. For the second scenario, the preliminary and coarse boundaries are implemented as PCA hyperellipsoids and the two fine classifiers described in Section 4.2.4 are tested: correlation matching and a gated recurrent unit (GRU) classifier. The correlation matching algorithm is deterministic, so it is only run once. Since the GRU training is stochastic, the test using the GRU fine classifier is run ten times and averaged.

Several static techniques are also trained and tested. Since the static techniques cannot be trained well using only a single data point for each load, these classifiers are only used for the second scenario, i.e., after one month of data has been collected. A DNN classifier and support vector machine (SVM) classifier are trained using the same four-dimensional feature space that the preliminary checks and coarse classifiers use: P_{ss} , Q_{ss} , S_{peak} , and transient time. A GRU classifier is trained using the same fine feature space as the fine classifier, six seconds of P and Q . For all three models, the data was split into 80% training and 20% validation, with data stratification to allocate samples evenly based on sample class. For the SVM classifier, a radial basis function kernel is used with a regularization parameter of one. The kernel coefficient is one over the number of features times the variance of the training data. Both the

DNN and GRU were trained with Adam-optimized backpropagation with a learning rate of 0.001 and categorical crossentropy as the loss function. The validation loss was used as a stopping criterion, such that training was stopped after fifteen epochs of no significant improvement. The DNN was implemented with two hidden layers of 50 and then 30 ReLU-activated neurons, followed by a softmax-activated layer. The GRU was implemented with a GRU layer with 50 ReLU-activated neurons, then two densely connected layers with 30 ReLU-activated neurons each, and finally a softmax-activated layer. For the DNN and GRU, ten different models were trained and the results were averaged. The tests are summarized below:

- Starting with a single event for each load (one-shot)
 - Adaptive correlation matching
- Starting with one month of data
 - Adaptive correlation matching
 - Adaptive GRU
 - Static SVM
 - Static DNN
 - Static GRU

The precision, recall, and F_1 -score are calculated for each load,

$$\text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}, \quad F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \quad (4.3)$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives [135]. Precision is the proportion of reported events that are correct. Recall is the proportion of load events that are reported. The F_1 score is the the harmonic mean of precision and recall. Precision and recall values of one indicate perfect performance in identifying a specific class, leading to “perfect” F_1 scores equal to one. Lesser values indicate imperfect classification. The results for the one-shot scenario and one month scenarios are presented in Table 4.2 and

Table 4.2: Accuracy of classifying on-events (one-shot).

Equipment	One-Shot	
		Adaptive Correlation Matching
	# of Events	F_1 score
MPDE jacket water (JW) heater	116	0.991
MPDE lube oil (LO) heater	277	0.993
MPDE prelube pump	241	0.966
SSDG jacket water (JW) heater	1177	1.0
SSDG lube oil (LO) heater	2731	0.999
FOP centrifuge	1130	0.714
FOP feed pump	340	0.974
CPP pump	560	0.991
Graywater pump	1999	0.999
Bilge and ballast pump	212	0.990
Air compressor	1389	0.996
Air conditioner	73	0.993

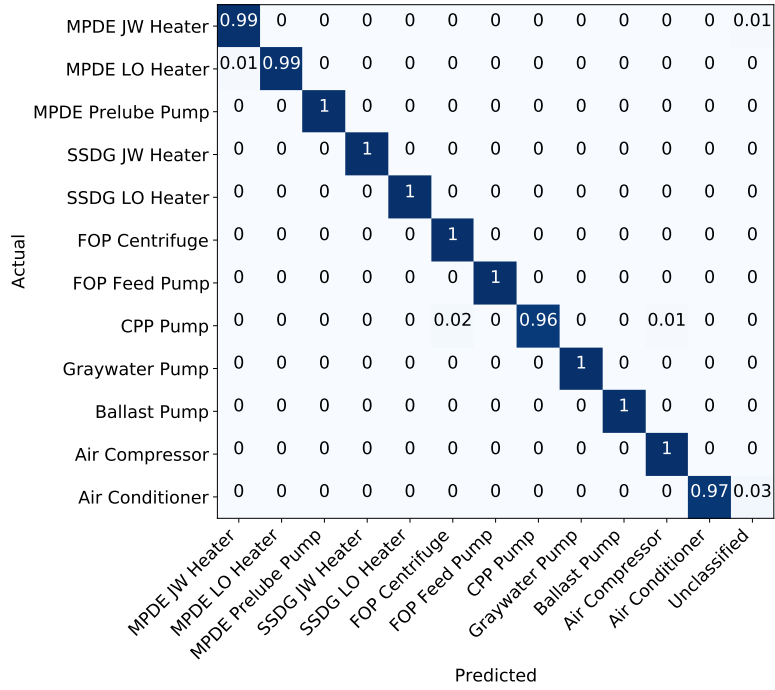
Table 4.3, respectively. The adaptive classifiers show improved performance over the static classifiers for several loads, including the CPP pump, MPDE prelube pump, and bilge and ballast pump. The one-shot scenario using adaptive correlation matching performs suboptimally on only the FOP centrifugal motor, due to an abnormal load operating pattern described in the following section. Fig. 4-7 shows two normalized confusion matrices to compare the adaptive versus static classifiers starting with one month of data. Fig. 4-7a shows the results of adaptive correlation matching and Fig. 4-7b shows the results of the static SVM. For the static SVM, the CPP pump is often misclassified as the FOP centrifuge, the MPDE prelube pump is often misclassified as the graywater pump, and the bilge and ballast pump is often misclassified as the MPDE prelube pump and graywater pump. These results are consistent with the demonstration in the previous two-dimensional demonstration of Fig. 4-6. The separability issues persist even in higher dimensions.

Table 4.3: Accuracy of classifying on-events (one month initial dataset).

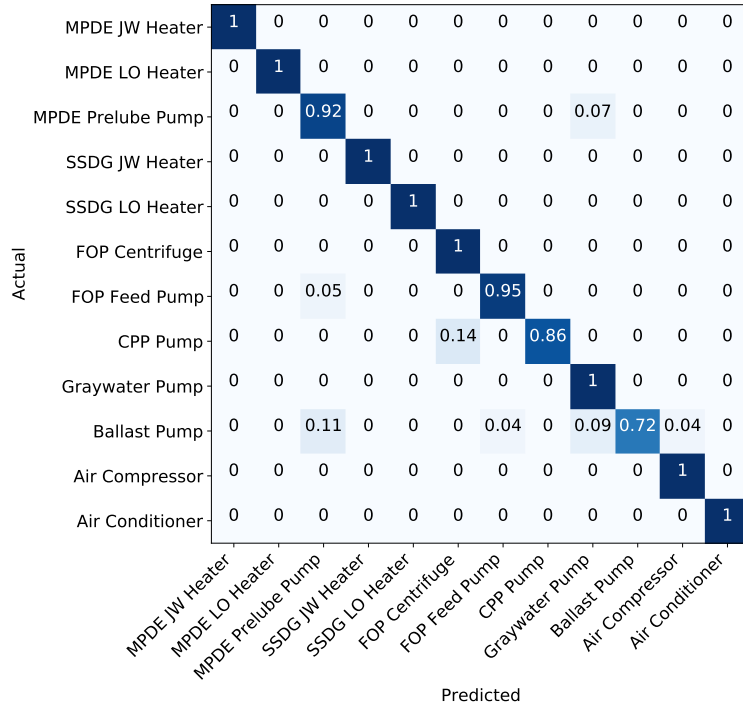
	Trained with One Month of Data					
		Adaptive		Static		
		Correlation Matching	GRU	SVM	DNN	GRU
Equipment	# of Events	F_1 score				
MPDE jacket water (JW) heater	91	0.978	0.997	0.995	0.991	0.830
MPDE lube oil (LO) heater	268	0.992	0.996	0.998	0.995	0.980
MPDE prelube pump	241	0.998	0.998	0.884	0.903	0.675
SSDG jacket water (JW) heater	956	1.0	1.0	1.0	1.0	0.993
SSDG lube oil (LO) heater	2705	0.999	0.999	0.999	0.999	0.992
FOP centrifugal motor	1103	0.994	0.996	0.968	0.972	0.934
FOP feed pump	329	1.0	1.0	0.953	0.910	0.814
CPP pump	530	0.982	0.971	0.925	0.934	0.830
Graywater pump	1965	0.999	0.999	0.990	0.990	0.944
Bilge and ballast pump	208	0.998	0.972	0.838	0.791	0.540
Air compressor	229	0.983	0.954	0.977	0.986	0.874
Air conditioner	39	0.987	0.997	1.0	0.996	0.920

4.4 Physical Interpretation of Results

The improved performance of the adaptive framework over the static classifiers can be explained by analyzing the physical operation of these loads over time. Load electrical characteristics are sometimes dependent on changing operating conditions. These conditions are manifested in the feature space as concept drift, which the adaptive methods seek to cope with. For instance, the steady-state power of the CPP pump was shown in Fig. 1-8 to vary based on operating pressure. Other loads also have drifting steady-state power behavior, as shown with example turn-on transients of the ballast pump and MPDE prelube pump in Fig. 4-8a and Fig. 4-8b, respectively.



(a) Adaptive correlation matching.



(b) Static SVM.

Figure 4-7: Normalized confusion matrices rounded to two decimal points with one month of training data.

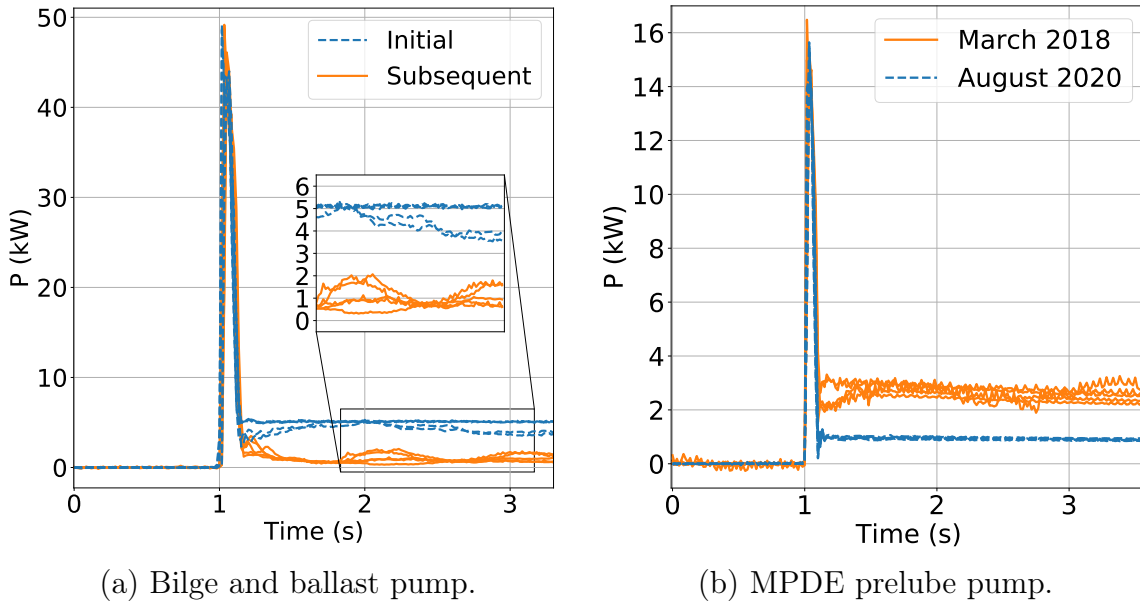


Figure 4-8: Example turn-on transients demonstrating drifting steady-state values.

This section gives provides analysis for this behavior. The drift metrics tracked by the framework are also useful diagnostic indicators, for tracking both short-term and long-term behavior.

4.4.1 Bilge and Ballast Pump

The bilge and ballast pump is used for emptying machinery space bilges of excess water and for taking on ballast water for stability purposes [27]. The bilge and ballast pump electrical signature is highly variable, likely due to air pockets within the bilge and ballast pumping system. When pumping bilges and ballast tanks, operators try to get the tanks and bilges to the lowest level possible. As a result, the pump takes in a mixture of air and water. After the pump is turned off and suction is shifted to a new tank, the air remains in the system, resulting in a prolonged start sequence in which the pump draws a variable amount of power. The variable nature of the power consumption results in drift in the feature space. Over time, six micro-clusters are formed in the drift tracking process, as shown in Fig. 4-9a. The representative exemplars for these six micro-clusters are shown in Fig. 4-9b.

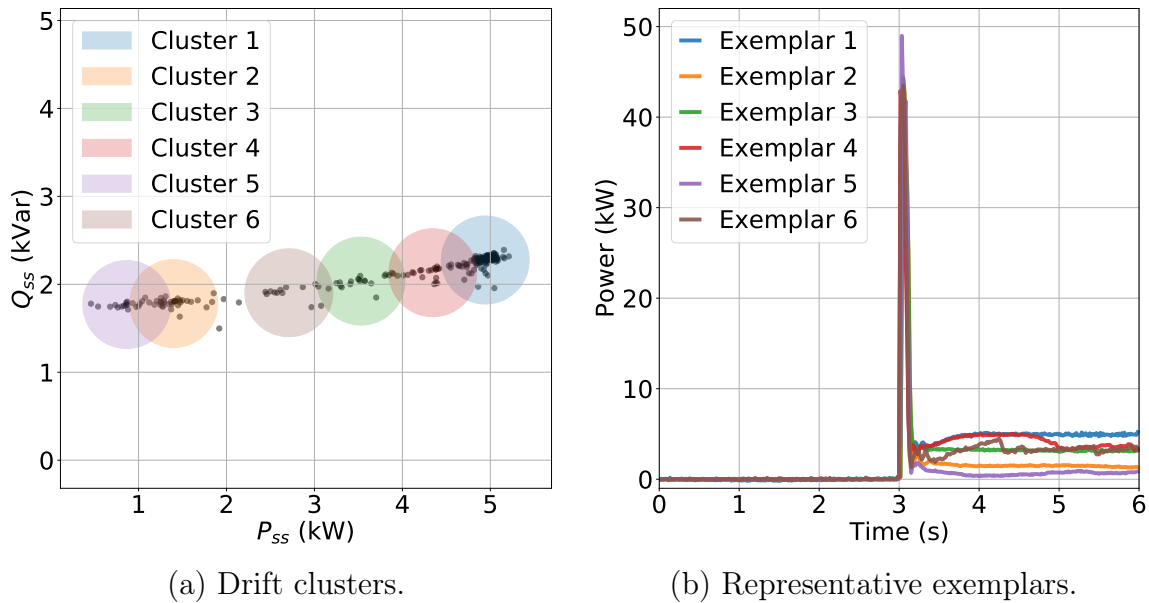


Figure 4-9: Bilge and ballast pump drift clusters and representative exemplars.

Fig. 4-8a shows the first five transients of the bilge and ballast pump, corresponding to the first month of data, showing that in each of these transients, the pump quickly reaches steady state. Each of these transients are within the first drift cluster. Also shown in Fig. 4-8a are five subsequent transients for the most extreme case, in which the pump draws approximately one-fifth of the power of the initial transients. These transients are all within the leftmost drift cluster. Because there are no instances of load drift in the first month of data, it would not be possible to predict this variable nature of the pump, even if synthetic data was added to the first month's training and validation data. All of the static classifiers have poor performance on the bilge and ballast pump, due to misclassification of the bilge and ballast pump as other loads. The large number of false negatives can be demonstrated by viewing a graph of the bilge and ballast pump recall over time, as shown in Fig. 4-10. The recall value is calculated at every load event for all the data up to that time index. Although the recall is initially poor for this load, it improves over time for the adaptive classifiers.

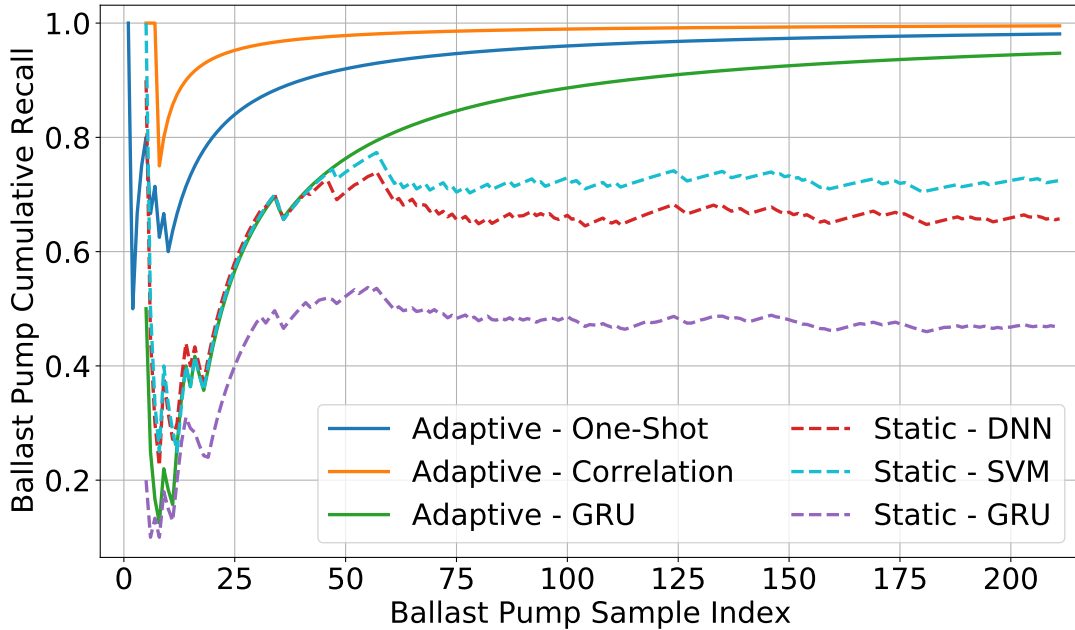


Figure 4-10: BP cumulative recall for six classifiers run on the full dataset.

4.4.2 Controllable Pitch Propeller Pump

As was detailed in Section 1.2, the NILM detected changes in the monitored CPP ‘C’ pump’s steady-state power consumption (P_{ss}) for over a four-year period. The change was slow; however, at its worst the difference in real-power steady state between two different turn-on events is more than 4 kW. As the power of the CPP pump increases, the radii of the preliminary and coarse boundaries for this load also increase. Over time, six micro-clusters are formed in the drift tracking process, which was shown in Fig. 4-4a. Fig. 4-11 shows the cumulative recall for the CPP pump to analyze the performance of both the static and adaptive classifiers over time as the load drifts, in particular the false negatives of the static classifiers. As noticed by the large negative slope in the recall, the static classifiers show a reduction in performance starting at around the 259th CPP pump sample. This corresponds to February 2018, when the CPP pump showed an increase in power after replacement of the hydraulic control valves. When the CPP pump’s electrical characteristics begin to drift back to its initial state, the performance of the static classifiers begin to improve. This improvement begins at around the 395th CPP pump sample (August 2018) for the

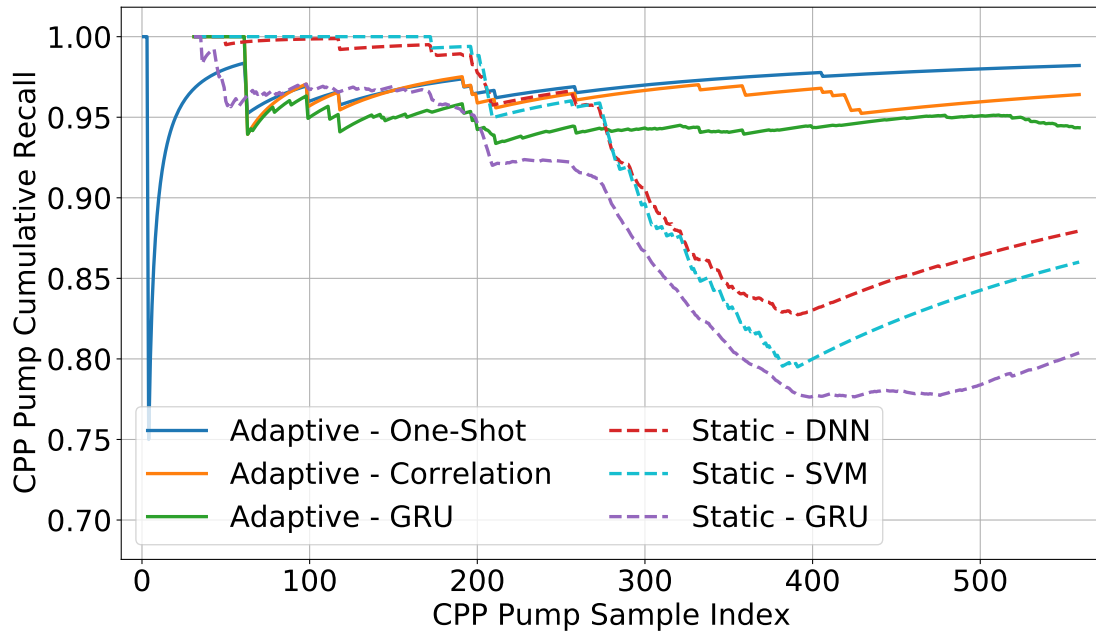


Figure 4-11: CPP pump cumulative recall values for six classifiers run on the full dataset.

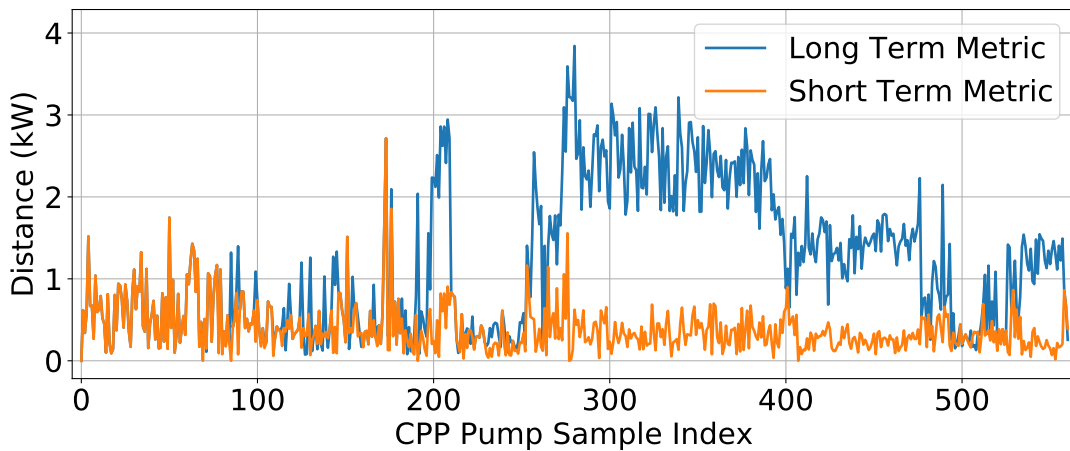


Figure 4-12: CPP pump long-term and short-term distance metrics.

static classifiers.

In contrast, the adaptive classifiers do not exhibit this trend of its performance being correlated with the load drift. Potential errors of the static classifier are revealed by examining the long-term and short-term drift metrics. As shown in Fig. 4-12, the long-term metric increases at around the 259th CPP pump sample point, the same time that the static classifiers show a reduction in performance. However, the short-

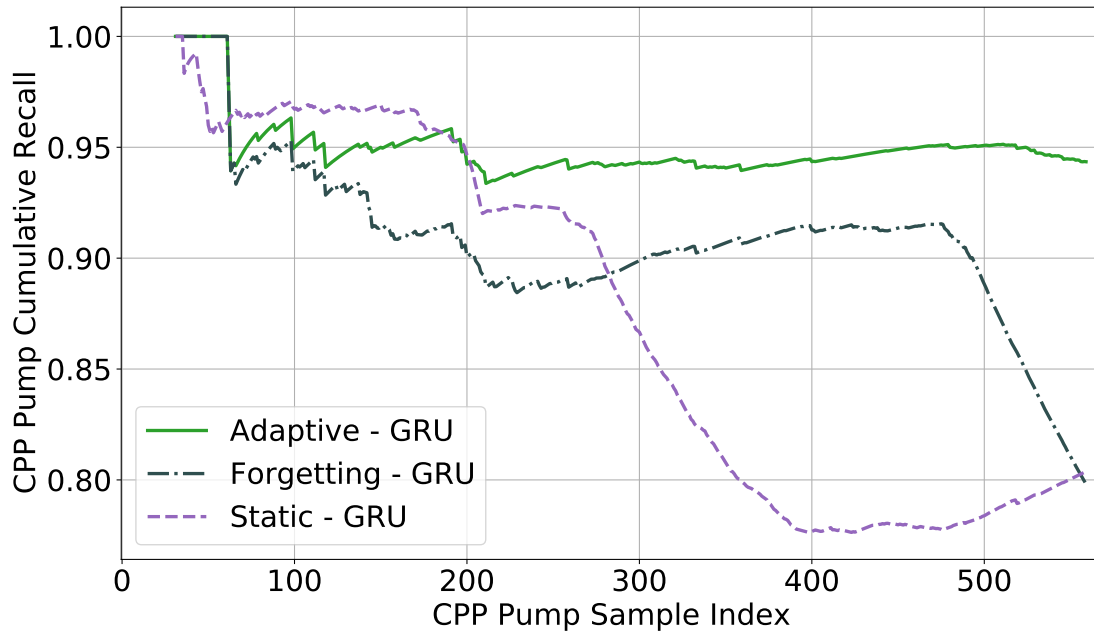


Figure 4-13: CPP pump cumulative recall for three GRU implementations.

term drift metric remains relatively small, corresponding to small changes between load events, indicating an incremental concept drift. Thus, the adaptive framework can still accurately identify and track the CPP pump. Then, the long-term metric can be used as an indicator of changing machine behavior.

The behavior of the CPP pump is used to demonstrate the proposed framework’s ability to avoid catastrophic forgetting in the presence of recurring concept drift. For demonstration, an alternative GRU adaption scenario is tested, such that the framework was kept identical to before except for the implementation of the GRU retraining. In this alternative adaptive implementation, the GRU was retrained after the formation of every drift cluster, as before, except this time the memory for each load is the twenty most recent transients. Over time, the GRU will not have access to the older data when retraining. This test was run ten times and averaged, with the recall values labelled “Forgetting” shown in Fig. 4-13. Although this alternative adaptive implementation is able to produce a high recall score until around the 479th sample (July 2019), after this point the performance decreases drastically, presumably because it has no memory of this previously known state anymore. For the static GRU

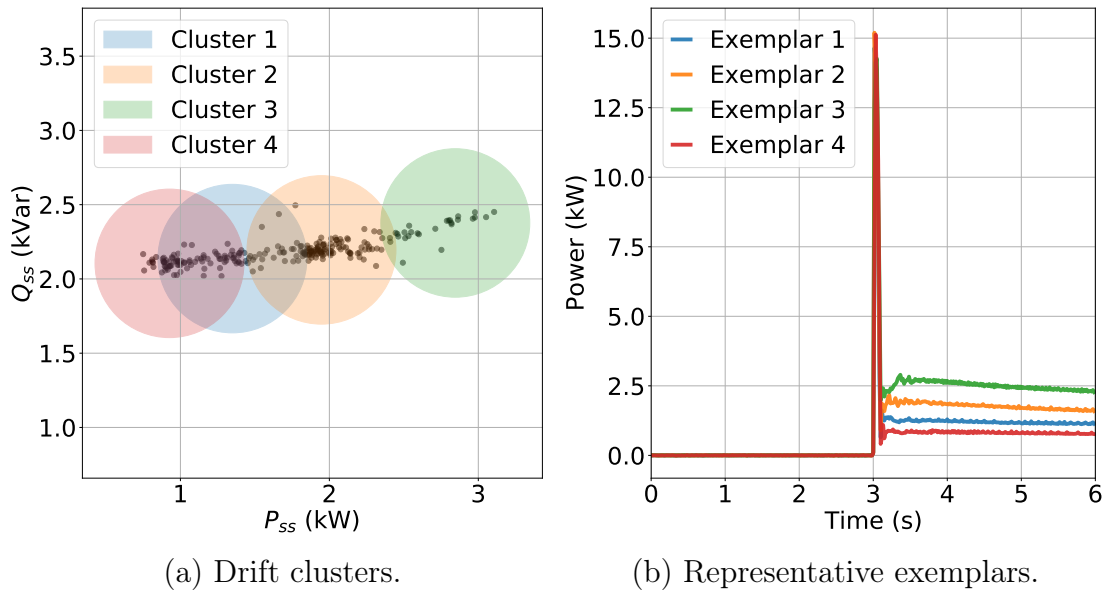


Figure 4-14: MPDE prelube pump drift clusters and representative exemplars.

classifier, in the plot labelled “Static,” the score improves when the load drifts back to the initial state, since this is the state the model was trained on. Both the adaptive GRU classifier with only recent memory (“Forgetting”) and the static GRU classifier (“Static”) exhibit many false negatives. In contrast, the proposed adaptive GRU implementation, labelled “Adaptive,” has high recall values for the entire duration, even during recurring drift when the load drifts back to the initial state.

4.4.3 Main Propulsion Diesel Engine Prelube Pump

The main propulsion diesel engine (MPDE) prelube pump ensures adequate lubricating oil distribution during startup and shutdown of the engine. The pump is manually energized prior to engine startup and operates automatically during engine shutdown. The MPDE prelube pump steady-state power showed similar trends to the CPP pump. The control valves of the MPDE prelube pump were replaced at the same time as for the CPP pump, and there was subsequently a similar increase in the power of the MPDE prelube pump in February 2018, and a similar decrease in power over time afterwards. Fig. 4-8b shows an example of five transients in the time domain from March 2018 and five transients from August 2020. Over time, four

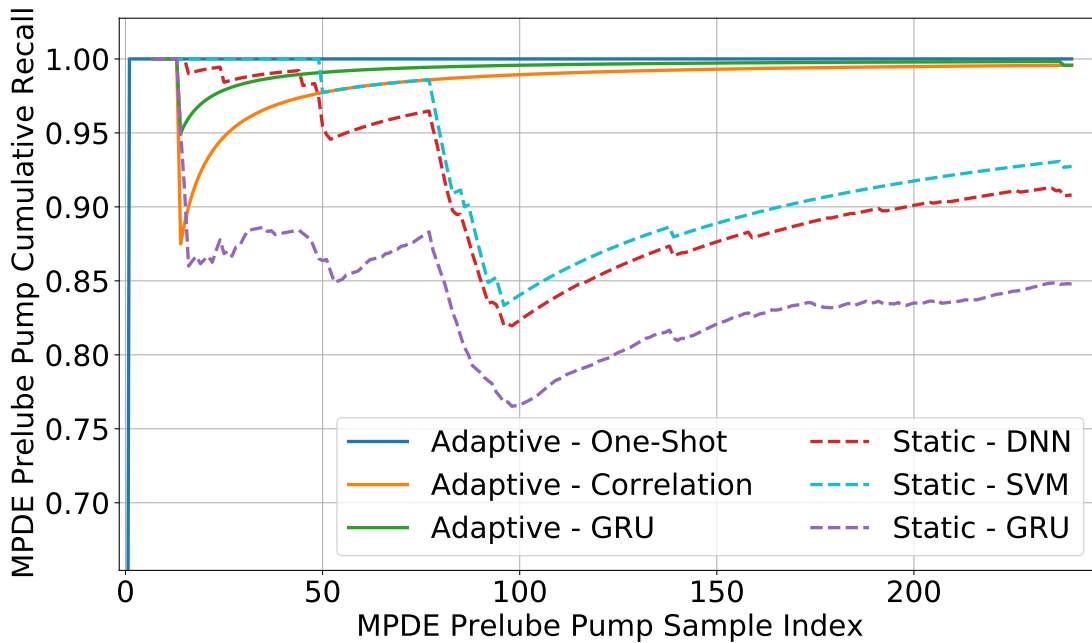


Figure 4-15: MPDE pre-lube pump cumulative recall values for six classifiers run on the full dataset.

micro-clusters are formed in the drift tracking process, as shown in Fig. 4-14a. The representative exemplar for each of these micro-clusters is shown in Fig. 4-14b. As the load drifts away from its initial state, the MPDE pre-lube pump is misclassified by the static classifiers, leading to an increasing number of false negatives. This is shown in the plot of recall over time in Fig. 4-15. The steep decrease in recall at about the 80th MPDE pre-lube sample correlates exactly to February 2018. Meanwhile, the adaptive classifiers all show almost perfect performance. The short-term and long-term drift metrics are shown in Fig. 4-16, showing an increase in long-term metric also at the 80th sample point. The short-term metric briefly increases at the time, but quickly returns to a relatively small distance.

4.4.4 Fuel Oil Purifier

A shipboard fuel oil purifier (FOP) presents an example of a sudden change in operating condition. The FOP is run frequently while underway to clean the diesel oil before use in the MPDEs or SSDGs. The FOP consists of a feed pump, which

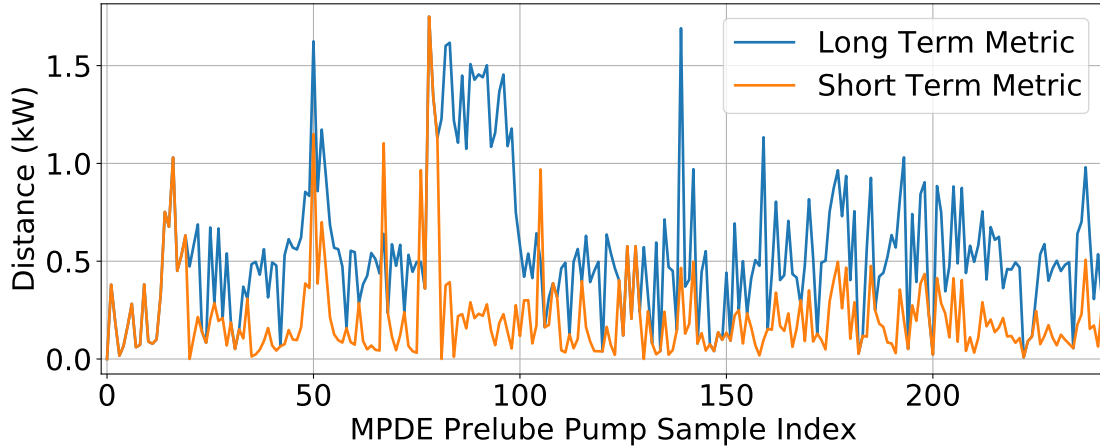


Figure 4-16: MPDE prelube pump long-term and short-term distance metrics.

draws fuel from various storage tanks around the ship and a motor-driven centrifugal separator system [27]. Observations showed two distinct operating conditions for the FOP centrifuge, a cold-start and a warm-start. Under normal operating conditions, the feed pump will energize, followed shortly by the centrifuge. The centrifuge energizes in the cold-start condition; that is, the motor is starting up after being off for a sufficiently long period. Large current is needed to begin rotation of the motor shaft. Often, while the feed pump is still energized, the centrifugal system’s motor cycles. This means the centrifugal system’s motor turns off, then re-energizes shortly after in a warm-start condition, often only a few seconds after turning off. It is likely that the motor shaft is still spinning, so the inrush current is significantly smaller. This faulty scenario puts unnecessary wear on the centrifugal system’s motor, as these warm-starts do not have any function for the system. Example cold-start and warm-start transients are shown in Fig. 4-17.

Fig. 4-18 shows the recall and precision scores for the FOP centrifuge. The decreasing recall for the one-shot adaptive classifier is because the warm-start events were deemed unclassifiable. The first instance of the FOP centrifuge was in the cold-start condition. The warm-start condition acts as a sudden concept drift. Although the warm-start instances lead to an increase in false negatives for the FOP centrifuge, it is significant that they did not become false positives for another load. By identifying the instances as unclassifiable, an operator could then identify the unclassified

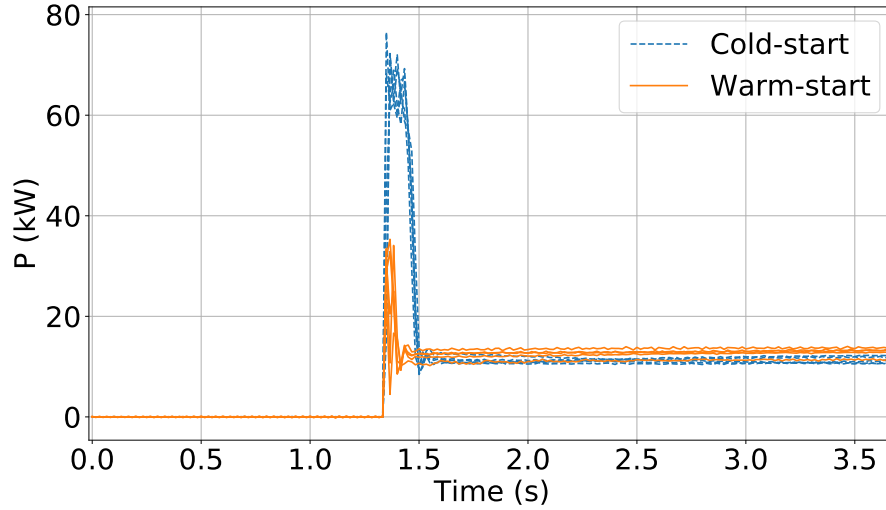


Figure 4-17: Five examples each of the cold-start and warm-start turn-on transients of the fuel oil purifier centrifuge motor.

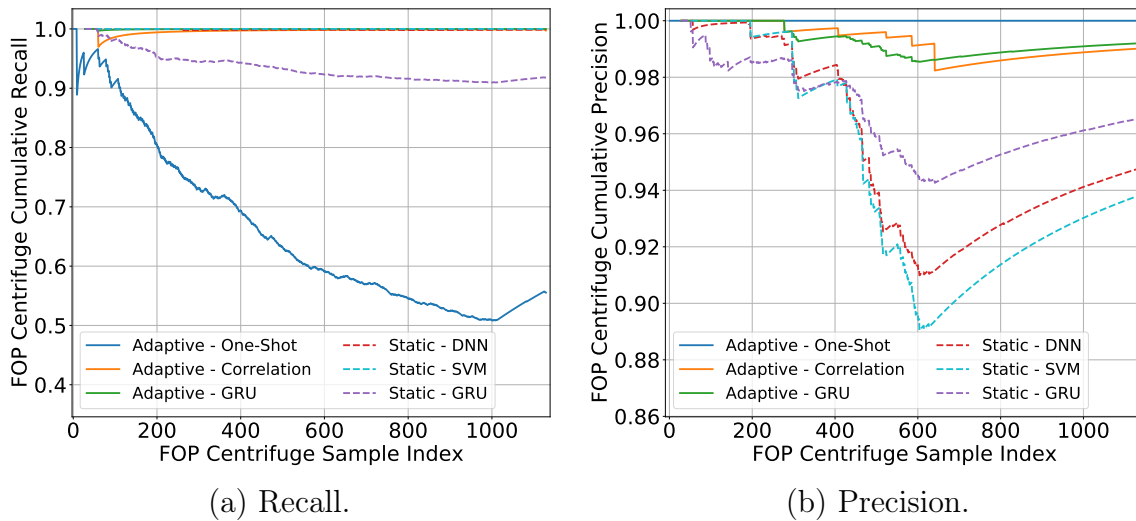


Figure 4-18: Cumulative recall and precision values for six classifiers run on the full dataset for the fuel oil purifier centrifuge motor.

events as the centrifuge operating in a faulty warm-start condition. For the adaptive correlation matching which started with one month of data, because there was an example of the warm-start condition in the training data, the classifier was able to correctly identify both the cold-start and warm-start conditions.

Unlike the adaptive framework, which is robust to outlier load events, the static classifiers will label every incoming load event. This means every false negative of

one load results in a false positive in another load. This is illustrated by examining the precision score over time of the FOP centrifuge. The steep decrease in precision starting at about the 408th FOP centrifuge sample is in February 2018. As expected, this is exactly when the CPP pump recall starts to decrease, because the CPP pump is being misclassified as the FOP centrifuge. The precision begins to improve at around the 630th FOP centrifuge sample (August 2018), the same time that the CPP pump recall begins to improve.

4.5 Chapter Summary

An adaptive classification framework was presented in this chapter to simultaneously address incremental and recurring concept drift, one-shot or few-shot learning, and extreme verification latency. The results of running the framework on a collection of shipboard loads demonstrate the ability of the framework to accurately detect load geometric events even as they drift in the feature space over time. Because the classifiers are physically informed, they are effective in both one-shot and few-shot scenarios, while allowing for adaptation as data is collected in real time. For loads with time-dependent changes in steady state, the drift metrics present an indicator of possible load degradation. This information can be utilized by a watchstander as a condition-based maintenance aid, so equipment and systems can be repaired or replaced before complete failure occurs. Using the proposed semi-supervised framework, a dataset that includes drifting load behavior can be built that will make supervised training more effective.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Energy Disaggregation of Stochastic Power Behavior

The advantage of nonintrusive load monitoring relies on the ability to disaggregate the energy consumption of individual loads from the aggregate power stream. For some loads, accurate identification of geometric events is sufficient for accurate energy estimation. The energy consumption of these loads can be computed by identifying steady-state power changes during transient behavior, i.e., when the power usage of a system quickly changes when transitioning between states, and tracking the operating duration. That is, a constant steady-state value can be assumed for the entirety of load operation. However, for many loads this is not the case. Many loads have continuously changing power demand (i.e., continuously variable loads) [7]. Loads with continuously changing power demand do not have unique power consumption levels or a fixed number of states for the entirety of load operation. Accordingly, it is much more difficult to identify and track the operation of continuously variable loads with the same techniques used for loads that can be fully characterized with distinct, relatively repeatable power consumption transients.

Power electronic loads such as variable frequency drives (VFDs) and light dimmers may exhibit smoothly variable power demand. References [7] and [88] present identification methods for power electronic loads. These methods rely on significant higher-order current harmonics to distinguish the fundamental component power con-

sumption of power electronic loads from that of loads with constant steady-state power demand. Both methods assume that only a single variable load is present in the aggregate stream, and both methods rely on higher harmonic signatures that are not always present for all loads.

As was shown in Section 3.1.2.2, some loads have non-constant power demand that varies with loading conditions. These loads, such as computer-controlled machine tools, remain stubbornly resistant to conventional nonintrusive electrical monitoring methods. The power behavior of these loads can be modelled as stochastic processes. Fluctuating power demands, e.g., in a CNC machine tool, can also correspond to loaded conditions and mechanical processes that introduce wear into the system. Fortunately, the fluctuations typically have certain statistical properties characteristic of the load operation. New statistical features for stochastic power behavior were introduced in Section 3.1.2.2 and are demonstrated in this chapter.

Advances in pattern classification and deep learning have enabled automated feature extraction and classification on high-dimensional data. On the surface, identification of stochastic power behavior using their statistical features would appear to be an ideal problem domain for such deep classifiers. Classifiers such as convolutional neural networks can extract features and “learn” the stochastic behavior of the loads of interest. However, these approaches require a relatively large amount of training data and effort to obtain high generalization [97]. A practical NILM in industrial sites will likely need to collect its own training data, making training data a scarce resource. Rather than use high-dimensional time-domain data for classification, the proposed system extracts lower-dimensional features that describe the observed distributions of the power values. This makes it possible to avoid the burdens of requiring large training datasets and long training times associated with deep learning classifiers.

This chapter presents a method for the disaggregation of fluctuating power behavior in an aggregate power stream. New techniques are presented that estimate individual load energy consumption. These techniques are also useful for load condition monitoring. Laboratory and field results are presented from two case studies: machining equipment and load operation on a shipboard microgrid. The work in this

section was in collaboration with Aaron Langham.

5.1 Statistical Feature Extraction

As was shown, variations in power draw can occur at multiple time-scales as a result of different mechanical processes [6]. The power variations of interest in this chapter occur on slower time-scales than the main load on and off events, but at faster time-scales than smoothly varying power demand from VFDs. The fluctuating power variations, referred to in this thesis as “statistical” events, do not necessarily indicate that the load has transitioned to a different steady-state level. This section presents a method for disaggregating statistical events of multiple loads from an aggregate power stream.

For feature extraction, the examples presented in this chapter use four streams: real power residual stream (P_r), reactive power residual stream (Q_r), first-order difference of the real power residual stream (P'_r) and first-order difference of the reactive power residual stream (Q'_r). After the ECDFs are computed with the process described in Section 3.1.2.2, the ECDFs are used as a feature vector representation of the statistical properties of the window of interest. This work uses the k -nearest neighbors (k -NN) classifier, since many distance metrics are applicable. The feature vector used as input to the k -NN classifier in this work is a concatenation of four ECDF curves, computed for $P_{r,n}$, $Q_{r,n}$, $P'_{r,n}$, and $Q'_{r,n}$. The classification process involves finding the k nearest training data points to the input feature vector, using the Euclidean distance (l_2 norm). An alternate choice of vector norm could be used if desired, e.g. the l_∞ norm. The classification process involves finding the k nearest training data points to the input feature vector, using the Euclidean distance. The input is classified to the class whose points makes up the plurality of these k nearest neighbors. Typically, choosing a larger value of k makes the classifier less likely to overfit, but ignores more local patterns in the data [136]. A larger value of k will also require more computation in searching for the neighbors of an input point. For binary classification (i.e., there are only two classes), choosing k to be odd results

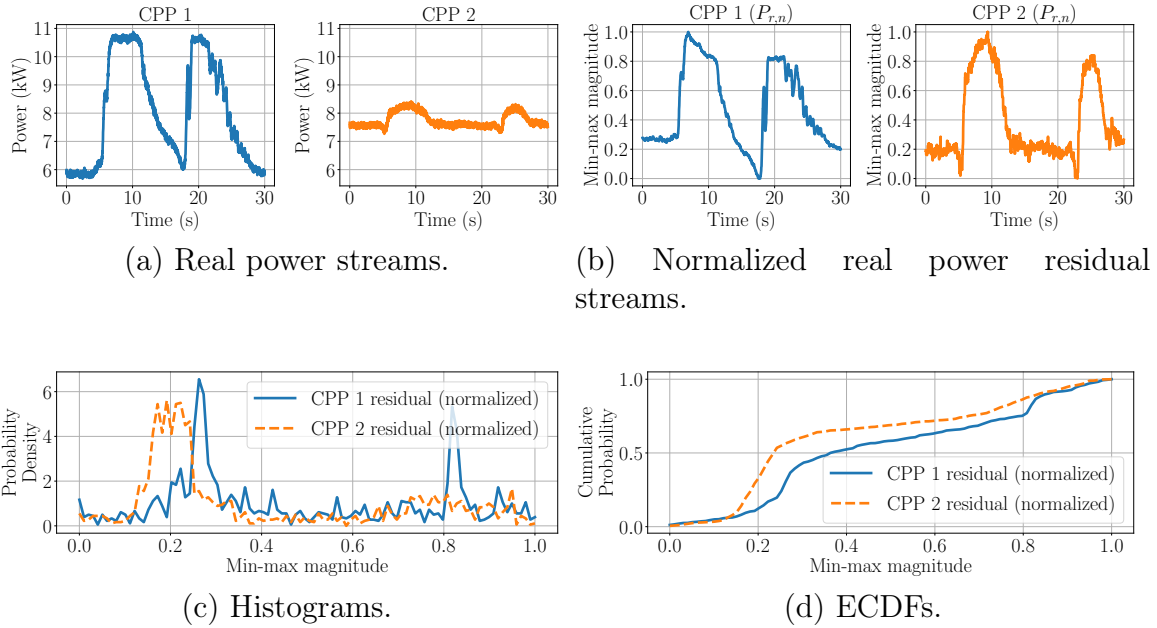


Figure 5-1: Process for calculating ECDF curves for two example windows of the CPP pump.

in there never being a tie between classes. Neighbors can also be weighted based on the distance to the input feature. This work assumes that statistical regions of multiple loads do not overlap. The overlap condition could be handled by adding classes representing each overlap combination.

The feature extraction process is demonstrated for two scenarios. First, it is demonstrated for two windows of the controllable pitch propeller (CPP) pump, shown in Fig. 5-1. Fig. 5-1a shows two 30-second windows of the original P stream of the CPP pump. As was shown in Section 1.2, the steady state power of the CPP pump is correlated with the normalized system operating pressure. The two windows in Fig. 5-1a shows two significantly different steady state values. Generally, as the steady state increases, the magnitude of the fluctuations in power decreases. That is, because the pump is already at a higher power and pressure, it does not need to “surge” as much to compensate to reach the required demand. However, with min-max normalization, these smaller, but still indicative surges in power have similar statistical characteristics to the larger surges. Fig. 5-1b shows the resulting normalized residual streams ($P_{r,n}$) using a 30-second median filter. Fig. 5-1c shows the histograms with 100 bins for

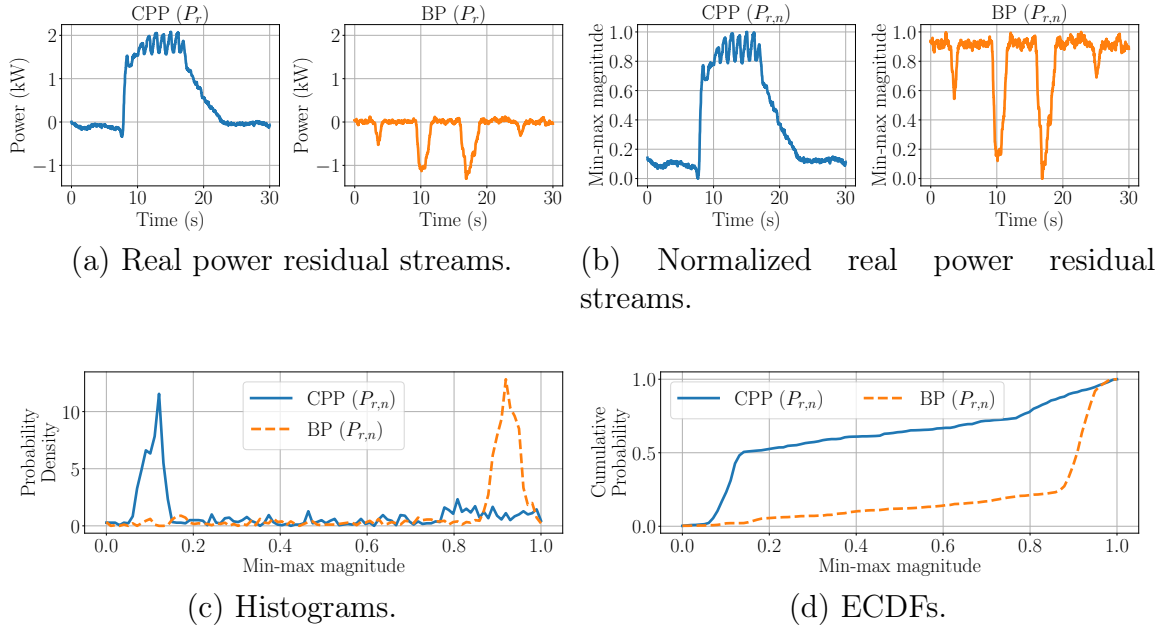


Figure 5-2: Process for calculating ECDF curves for an example window of the CPP and BP.

the two windows in Fig. 5-1b. Finally, Fig. 5-1d shows the ECDF curves for the two windows. Even though the overall magnitudes of the fluctuations are different, the ECDF curves are similar using the normalized data.

The next comparison is of an example CPP pump window and an example bilge and ballast pump (BP) window, as shown in Fig. 5-2. The residual real power streams using a 30-second median filter are shown in Fig. 5-2a. Then, the normalized residual streams are shown in Fig. 5-2b. Fig. 5-2c shows the histograms with 100 bins for the two windows. Finally, Fig. 5-2d shows the ECDF curves for the two windows. There is a large separation between these two ECDF curves.

5.2 Energy Estimation

Large fluctuations in power draw need to be attributed to the correct load for accurate energy estimates. These large fluctuations are not accounted for when using event-based algorithms that assume approximately discrete steady-state levels [16, 18]. Using the proposed statistical features, once the stochastic behavior has been classified,

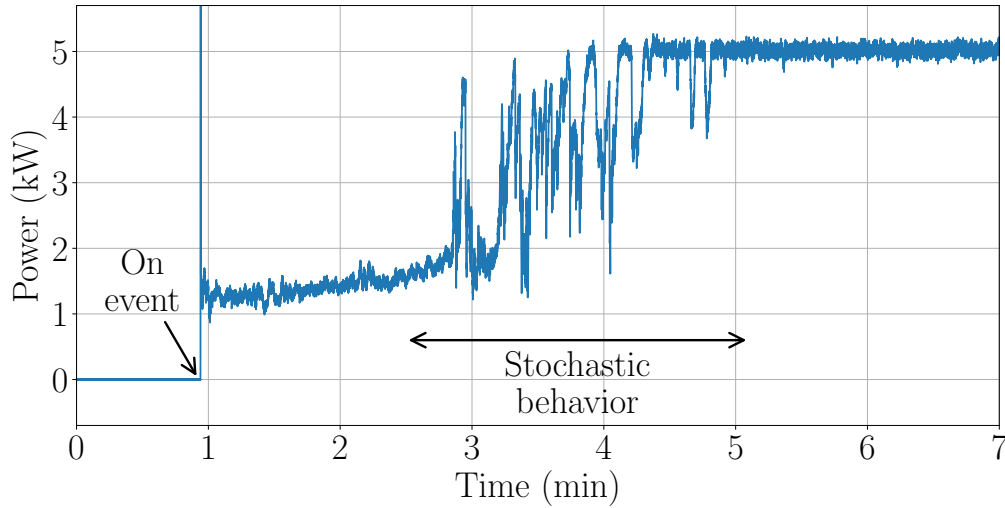


Figure 5-3: Power stream of a shipboard bilge and ballast pump with stochastic behavior in the transition between steady states.

the original power stream in the windows of interest can be numerically integrated to estimate the energy of each load. In addition, loads can exhibit stochastic behavior in the transition between steady states. By identifying the difference in steady-state power consumption for the load exhibiting stochastic behavior, its steady-state power can be updated in real time. As an example, consider the shipboard bilge and ballast pump shown in Fig. 5-3, with identified stochastic behavior labelled. The steady-state real power levels before and after the stochastic behavior are significantly different. This new steady-state value can be tracked and assigned to this load so that an incorrect steady-state value is not used in energy estimation.

The proposed algorithm to process the power stream into windowed events is given in Algorithm 5.1. This algorithm calls the function described in Algorithm 5.2 in order to disaggregate the energy for each load in the stream. A rolling window of user-defined length is run through the power stream, and geometric and statistical event detectors are run. The geometric event detector is responsible for identifying load turn-on and turn-off events, as well as transitions between discrete states. Here, it is assumed a geometric event classifier has already been trained. Although not in the scope of this chapter, many geometric event detectors are applicable. The statistical event classifier, described in Section 5.1, is responsible for identifying regions with

Algorithm 5.1 Algorithm for processing power stream into windowed events.

Input: power_stream

Input: load_list

Input: window_length

Input: geometric_length

Input: statistical_increment

```
1:  $n \leftarrow \text{window\_length}$ 
2: while  $n \leq \text{length}(\text{power\_stream})$  do
3:   window  $\leftarrow \text{power\_stream}[n - \text{window\_length}:n]$ 
4:   if CheckForGeometricEvent(window) then
5:     event_load  $\leftarrow \text{GeometricClassifier}(\text{window})$ 
6:     Update event_load.ss
7:      $i \leftarrow \text{geometric\_length}$ 
8:   else if CheckForStatisticalEvent(window) then
9:     event_load  $\leftarrow \text{StatisticalClassifier}(\text{window})$ 
10:    Update event_load.ss
11:     $i \leftarrow \text{statistical\_increment} \cdot \text{window\_length}$ 
12:   end if
13:    $i \leftarrow 1$ 
14:   EnergyEstimation(event_load,  $i$ ) ▷ Run Algorithm 5.2
15:    $n += i$ 
16: end while
```

Algorithm 5.2 Energy estimation algorithm.

Input: event_load (can be None)

Input: i , amount to increment window

```
1: if event_load is None then
2:   for load in loads do
3:     if only load is operating then
4:       load.energy += Trap(power_stream,  $i$ )
5:     else
6:       load.energy += Rect(load.ss,  $i$ )
7:     end if
8:   end for
9: else
10:  for load in loads except event_load do
11:    load.energy += Rect(load.ss,  $i$ )
12:    event_load.energy -= Rect(event_load.ss,  $i$ )
13:  end for
14:  event_load.energy += Trap(power_stream,  $i$ )
15: end if
```

large load power fluctuations. Throughout the stream, the algorithm keeps track of the current steady-state real power of each load (denoted as `load.ss` in Algorithms 5.1 and 5.2). This value is updated for a given load whenever a geometric or statistical event from that load is detected. For geometric events, the steady state after an on-event is calculated as the change in mean value of a user-defined length window (e.g., 0.2 seconds) at the end and start of the geometric window. Turn-off events detected by the geometric event detector are assumed to change the load's steady-state power to zero, rather than using the difference computed across the window. An assumption is made that the stochastic behavior occurs in superposition with the steady-state power consumption of all other energized loads and that the steady-state power of the other loads has remained constant. That is, the steady state after a statistical event is the aggregate power at the end of the window (i.e., the mean of last 0.2 seconds of the window) minus the stored steady-state values of the other energized loads. The statistical event detector will only classify and update the steady state if the classified load does not currently have a steady-state power of zero.

The energy estimation algorithm uses a combination of rectangular and trapezoidal integration [137], denoted in Algorithms 5.1 and 5.2 as *Rect()* and *Trap()*, respectively. In this chapter, only real power is integrated, so that the result has physical meaning as useful work done by the system. However, reactive and apparent power could also be integrated with the same process. When there is only one load operating, trapezoidal integration is performed, and the resulting energy is added to the operating load's total energy. When a statistical window is identified as a specific load, all other loads' energies are calculated with rectangular integration of their steady-state real power (i.e., `load.ss`). Trapezoidal integration is performed and the result is added to the load's total energy. The sum of the energy calculated for the other loads with rectangular integration is subtracted from the identified load's energy total, to remove the contribution of the other loads added in the trapezoidal step. In all other cases, rectangular integration uses each load's stored steady-state real power. An instantaneous estimate of disaggregated power can be computed similarly. For each window, each load's power is assumed to be its current steady-state

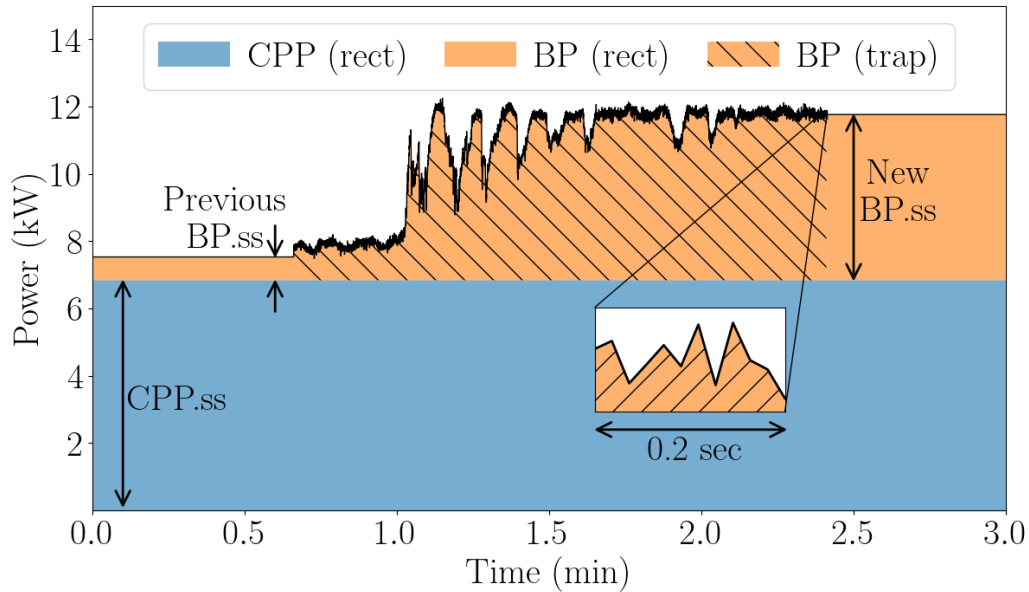


Figure 5-4: Demonstration of tracking steady state after statistical events in energy estimation algorithm.

power, unless it was responsible for a geometric or statistical event in the window. In this case, its power over the window is assumed to be the power stream over the window minus the sum of the current steady-state powers of the other loads. The total duration of statistical event operation for each load can be tracked using the identified statistical event windows.

An example of this process is illustrated in Fig. 5-4, in which there are two loads operating, a controllable pitch propeller (CPP) pump and a bilge and ballast pump (BP). In the figure, the CPP is the base load with a steady state indicated as “CPP.ss.” Since there are multiple loads energized and no CPP geometric or statistical events in this window, the CPP steady state is assumed constant and its energy is calculated using rectangular integration. For the BP, the area indicated by the shaded region has been identified as statistical event windows. The steady state of the BP prior to the statistical event region is indicated by “Previous BP.ss.” The steady state of the BP after the statistical event region is indicated by “New BP.ss” and was calculated by taking the mean of the last 0.2 seconds of the statistical region, as highlighted with the zoomed-in inset, and subtracting CPP.ss. Energy estimation of the BP statistical

region is calculated using trapezoidal integration.

5.3 Machining Equipment

Monitoring the energy consumption of machining equipment is useful for efficiency, energy reduction, and condition-based maintenance in the manufacturing industry [83, 138]. This section uses a CNC router and an industrial bench grinder to demonstrate the utility of the proposed statistical event detector and energy disaggregation method. CNC machining is a manufacturing process where automated machines remove raw material with cutting tools. The power draw of CNC cutting machine operation varies based on cutting conditions such as the workpiece material, cutting speed, feed rate, and depth of cut [83]. Industrial grinders are used for sharpening cutting tools and shaping objects. The power draw of a grinder depends on the required load of the grinding operation.

The CNC machine uses a Bosch 2.25 hp router motor, set to a fixed speed of 21,500 rpm. A CNC router was used to make a straight line cut through a piece of wood of various heights, as shown in Fig. 3-6. The major factors contributing to the total energy consumption are labelled on the power stream. There is an inrush up to 1700 W as the spindle accelerates (not fully shown). It then reaches a steady state of approximately 300 W while air-cutting. In this example, as the router cutting tool engages the wood, the power increases up to 500 W. A geometric event detector would likely only identify the spindle acceleration and deceleration as events and calculate the steady-state power for a short window before and after the event. However, this steady state would only correspond to air cutting, and not actual wood-cutting. The wood-cutting events and the large energy consumed during them would go undetected.

The industrial bench grinder used for testing was a Dayton 0.5 hp grinder with a six-inch grinding wheel, set to 1,800 rpm. An example power stream of machine operation is shown in Fig. 5-5. In this example, the startup and idle (baseline) operation is shaded blue, while the extra power required during grinding operation is shaded in gray. The base load of the unloaded grinding wheel (analogous to air-cutting

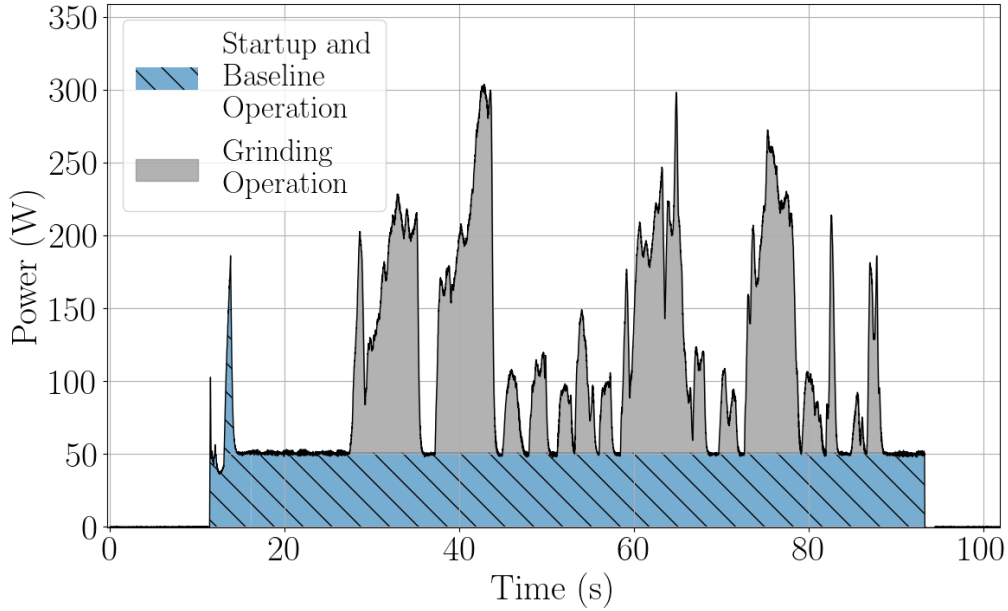


Figure 5-5: Power stream of industrial bench grinder.

of the CNC) is approximately 50 W. Grinding operations imitated burr removal and edge beveling in preparation for welding of mild steel flat stock. During grinding operation the power increased up to 300 W in this example. Integration of the shaded areas in Fig. 5-5 revealed that more than half of the total energy consumed in the example grinder stream is contained in the large fluctuations during loaded grinding operation (i.e., the area shaded gray). However, only the start and end of the base load operation would likely be identified as events by a geometric edge detector. A majority of the energy consumed would be “invisible” using purely geometric event-based disaggregation.

To demonstrate the statistical classifier, the CNC and grinder were both run individually with several runs of normal cutting and grinding operation, respectively. Statistical windows were detected from the data streams using a rolling window and median filter both with length of 20 seconds. The rolling window to generate training and testing data had an 80% overlap. The standard deviation threshold was set to $\sigma = 20$ W. In total, the dataset consisted of 21 and 68 windows of statistical activity for the CNC machine and grinder, respectively. The ECDFs for the dataset are shown in Fig. 5-6, for the $P_{r,n}$, $Q_{r,n}$, $P'_{r,n}$, and $Q'_{r,n}$ streams. The lines represent the

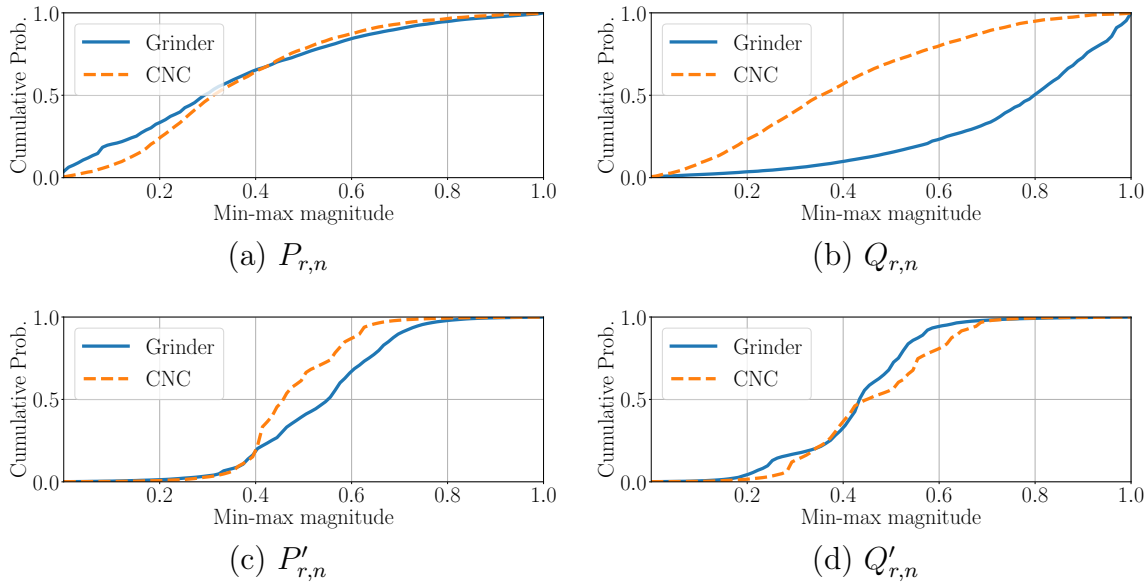


Figure 5-6: ECDF curves of the normalized CNC and grinder streams. The lines represent the average of all statistical windows in the dataset.

average of all the statistical windows in the dataset. These windows were randomly split into 60% training and 40% testing with data stratification to allocate samples evenly based on sample class. A k -NN classifier was trained with $k = 3$ without weighting. The dataset split and training was run 10 times for verification, with the results for the testing sets averaged and shown in Table 5.1. The results are presented as the average F_1 , precision, and recall scores, with σ_{F_1} , σ_{Pr} , and σ_{Re} showing the standard deviation of the runs. Precision, recall, and F_1 scores of 1 indicate perfect performance in identifying a specific class. The high scores for the grinder and CNC can largely be explained by the behavior of the $Q_{r,n}$ stream, as evident with the large separation of the average grinder and CNC ECDFs in Fig. 5-6b. The grinder appears capacitive and has negative reactive power during grinding operation. The CNC router appears inductive and has a positive reactive power which increases during cutting. This difference is because the two loads have different motor types. As a result, the histograms of the normalized reactive power stream will skew right for the grinder and skew left for the CNC.

To demonstrate the statistical classifier incorporated with the energy estimation algorithm, the CNC and grinder were run in an aggregately monitored environment.

Table 5.1: Machining equipment statistical event classification.

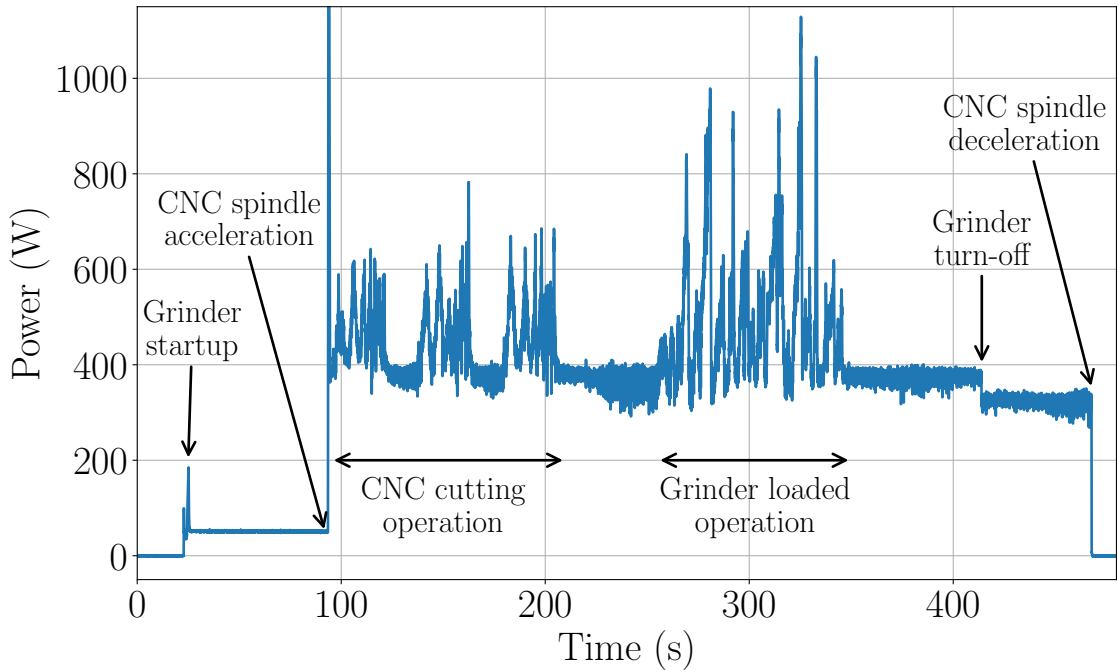
Equipment	# of Windows	F₁ (σ_{F1})	Pr (σ_{Pr})	Re (σ_{Re})
CNC router	9	0.97 (0.03)	1.00 (0)	0.94 (0.06)
Grinder	27	0.99 (0.01)	0.98 (0.01)	1.00 (0)

Table 5.2: Machining equipment energy estimation.

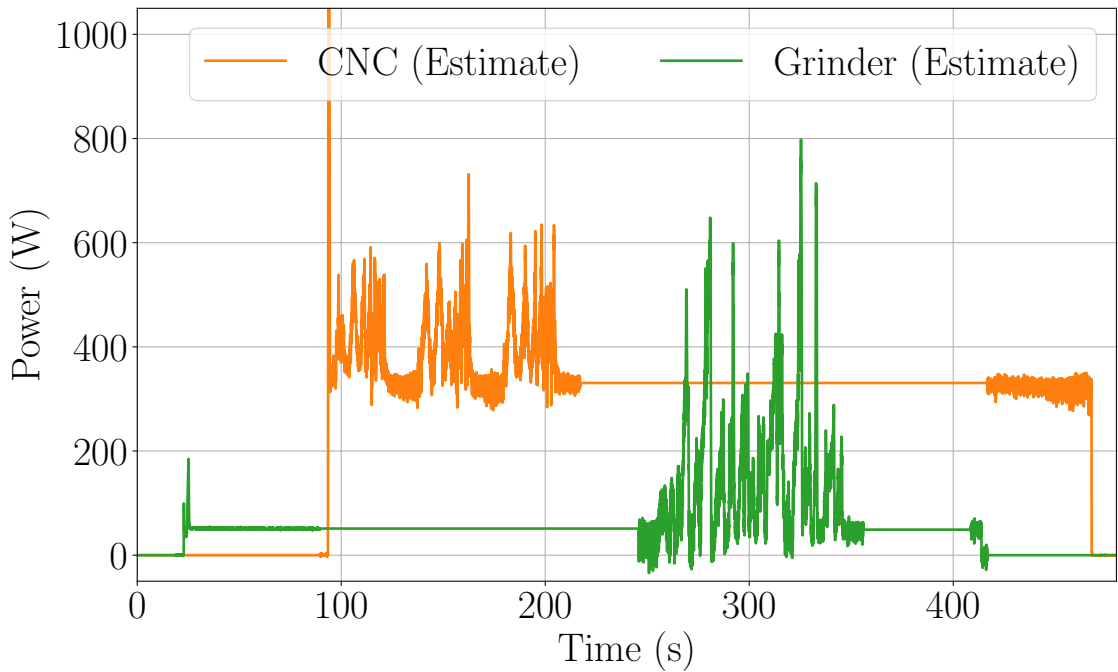
Method	Estimated (Wh)			Total Error	
	CNC	Grinder	Total	Percent	RMSE
Proposed	36.519	8.259	44.777	0.33%	5.64 W
Rectangular	37.232	5.481	42.713	-4.30%	80.51 W

The overlap for rolling statistical windows was set to 75%. The aggregate power stream is shown in Fig. 5-7a, with labels indicating main geometric and statistical events. Fig. 5-7b shows the estimated disaggregated power streams of the CNC and grinder. Table 5.2 shows the resulting energy estimates for the individual loads. The sum of the estimated individual loads’ energy is used to estimate the total energy. The results are compared to rectangular integration, in which the detected steady state at the load on-event is the assumed steady state for the entire duration of load operation. The error in total energy is shown as both percent error and root-mean-square error (RMSE) compared to the ground truth aggregate stream. For percent error, the estimated energy is compared to the ground truth energy computed with trapezoidal integration. RMSE was calculated as the root-mean-square difference between the ground truth total power stream and the sum of the estimated power streams of each load. The proposed method closely matches the measured ground truth, whereas the rectangular integration method results in an underestimate of total energy. This underestimate is due to the large underestimate of the grinder’s energy during grinding operation.

The tracked durations of grinder and CNC statistical events were 95 and 90 seconds, respectively. This corresponds to the amounts of time that the grinding wheel was being loaded and the CNC router bit was cutting. The methods described in this chapter enable NILM systems to perform cumulative cutting time-based tool condition monitoring (TCM) of various machines from an aggregate point. Cumulative



(a) Aggregate power stream.



(b) Disaggregated power streams of the CNC router and industrial benchtop grinder.

Figure 5-7: Machine shop energy estimation.

time TCM methods are commonly used to estimate cutting tool health and remaining useful life [139]. Non-uniform tool-life and loading conditions present challenges

for cumulative time TCM. However, with the techniques presented, a NILM could “weight” cutting time by equipment power consumption and account for non-uniform wear or use power consumption as a proxy for tool condition.

5.4 Shipboard Loads

On the monitored subpanels of USCGC Spencer, there are two loads that have stochastic behavior: the bilge and ballast pump (BP) and controllable pitch propeller (CPP) pump. The BP is used for emptying machinery space bilges of excess water in an emergency and for taking on ballast water for stability purposes [27]. When pumping bilges and ballast tanks, operators try to get the tanks and bilges to the lowest level possible, and as a result, the pump takes in a mixture of air and water. After the pump is turned off and suction is shifted to a new tank, the air remains in the system, resulting in a prolonged start sequence in which the pump draws a variable amount of power. It was shown in Section 4.3 that the initial steady state can be as small as one-fifth of the expected steady-state level. As shown in Fig. 5-3, the pump typically reaches the expected steady state, but it may take time on the order of minutes. This large discrepancy complicates energy disaggregation using purely geometric methods. As previously described, the monitored CPP pump is an electric hydraulic pump that supplements a separate gear driven pump in order to provide pressurized hydraulic oil to the CPP system and maintain hydraulic control pressure at the propeller. Hydraulic control valves maintain system operating pressure based on demand. As shown in Fig. 1-9, there are “surges” in power during operation of the CPP pump, as highlighted in the zoomed-in window. These surges are a result of the CPP pump compensating for the extra pressure required during ship maneuvering.

These two loads are on the starboard subpanel. For these loads, the median filter length and window length were both set to 30 seconds. For training, instances of individual operation of the CPP pump and BP were used. Rolling windows with 80% overlap were used with a standard deviation of $\sigma = 200$ W as the threshold for identifying windows with statistical activity for training and testing. In total, the

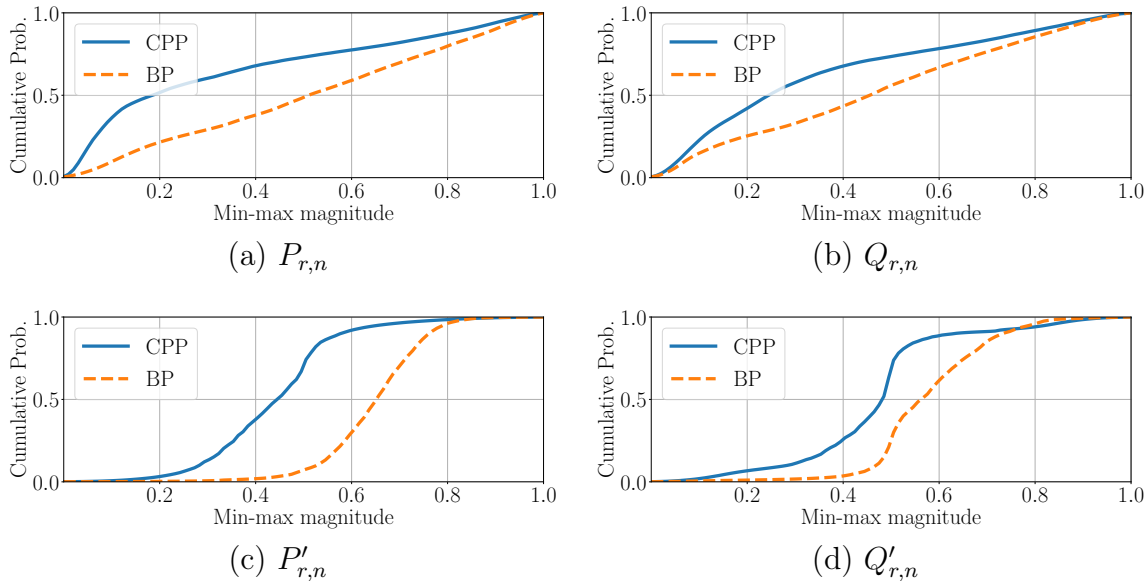


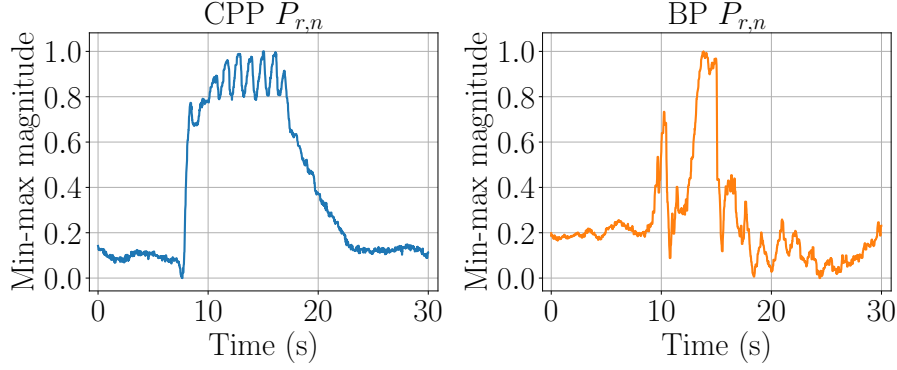
Figure 5-8: ECDF curves of the normalized CPP and BP streams. The lines represent the average of all statistical windows in the dataset.

Table 5.3: Shipboard loads statistical event classification.

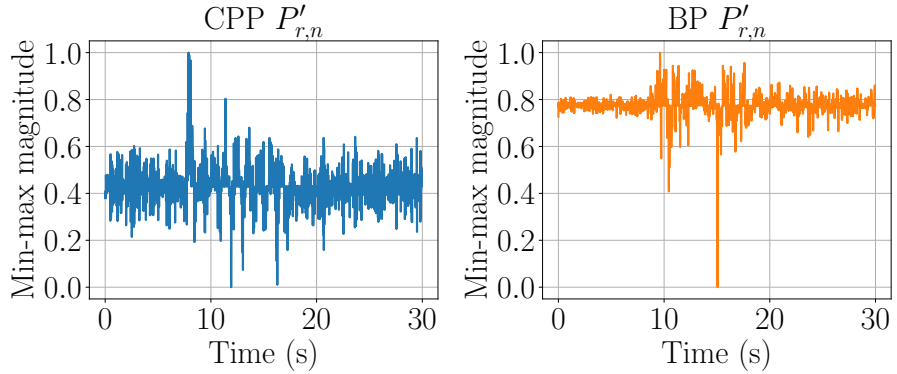
Equipment	# of Windows	F_1 (σ_{F_1})	Pr (σ_{Pr})	Re (σ_{Re})
CPP	184	0.97 (0.006)	0.97 (0.006)	0.97 (0.009)
BP	235	0.98 (0.004)	0.98 (0.007)	0.98 (0.005)

dataset has 460 windows of the CPP and 586 windows of the BP. The ECDFs for the dataset are shown in Fig. 5-8 for the $P_{r,n}$, $Q_{r,n}$, $P'_{r,n}$, and $Q'_{r,n}$ streams. The lines represent the average of all the statistical windows in the dataset. The data was randomly split into 60% training data and 40% testing data with data stratification. A k -NN classifier was trained with $k = 3$ without weighting. The dataset split and training was run 10 times for verification, with the results for the testing sets averaged and shown in Table 5.3.

The uniqueness of stochastic behavior can be explained by the physical mechanisms. For instance, for the $P'_{r,n}$ streams, as shown in Fig. 5-8c, the mean value of the BP ECDF (represented by the orange dashed line) corresponds to larger min-max magnitudes than the mean value of the CPP ECDF (the blue solid line). Fig. 5-9 shows example time-domain windows for both the CPP and BP. Fig. 5-9a shows



(a) Normalized residual streams ($P_{r,n}$).

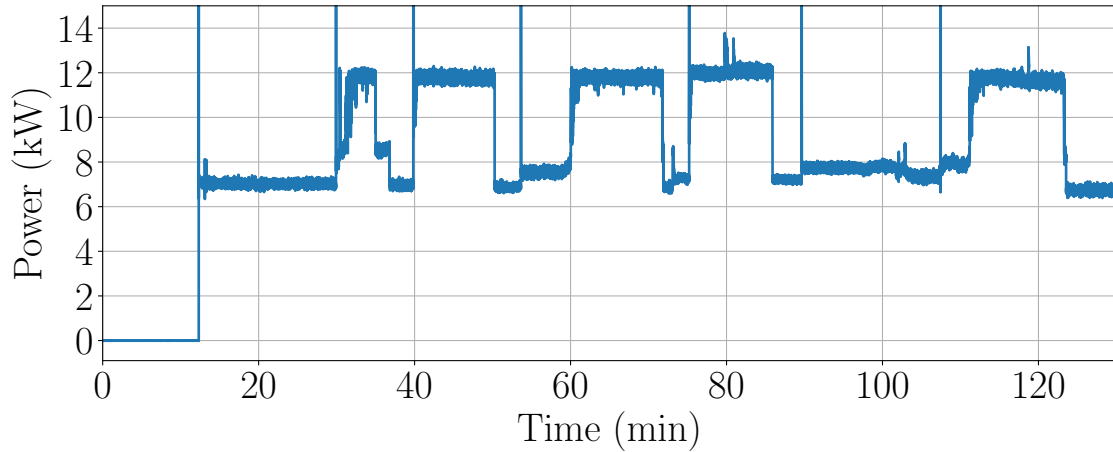


(b) Normalized first-order difference of residual streams ($P'_{r,n}$).

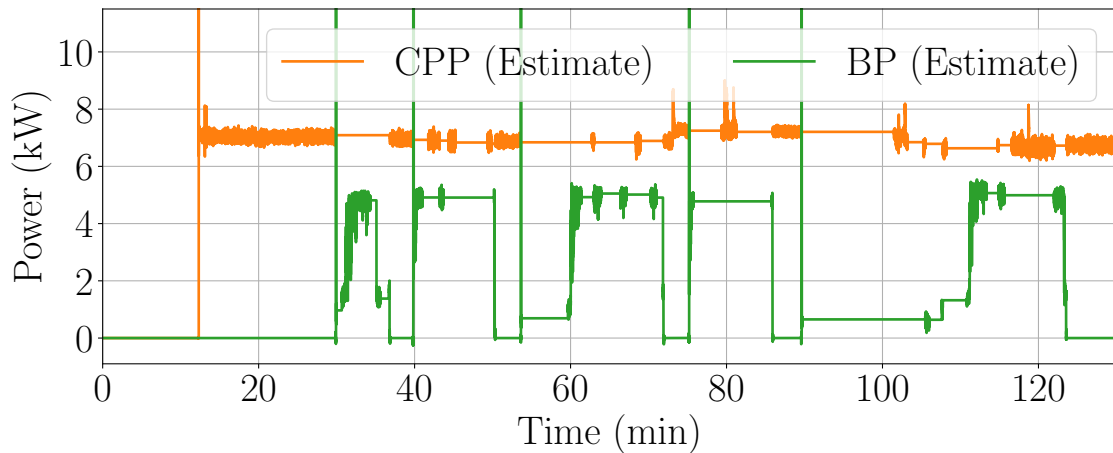
Figure 5-9: Normalized CPP and BP streams.

the normalized residual stream ($P_{r,n}$) and Fig. 5-9b shows the normalized first-order difference of the residual stream ($P'_{r,n}$). The CPP surge events have a sudden large increase in power but a slower decrease back to steady state. That is, most of the large first difference values are positive. Due to these large positive “spikes,” the average after min-max normalization will be less than 0.5. In contrast, the rapid fluctuations of the BP generally results in larger “spikes” in the negative direction, indicating many large magnitude negative first difference values. The average after min-max normalization is generally greater than 0.5.

The statistical classifier and energy estimation algorithm were run for a two-hour window of the starboard subpanel aggregate NILM stream in which the CPP and BP were both energized. The overlap for rolling statistical windows was set to 75% with threshold set to $\sigma = 100$ W. The aggregate power stream is shown in Fig. 5-10a. Fig. 5-10b shows the estimated disaggregated power streams of the CPP and BP.



(a) Aggregate power stream.



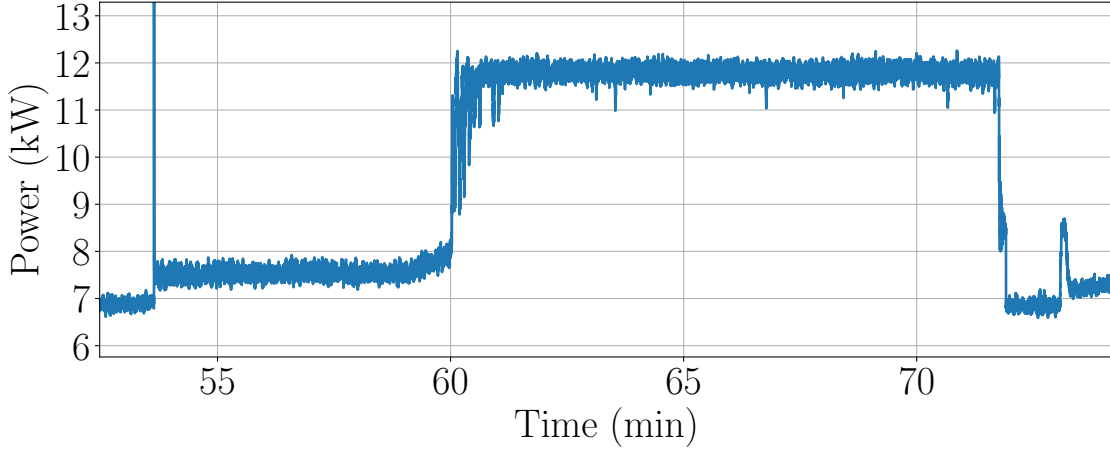
(b) Disaggregated power streams of the CPP and BP.

Figure 5-10: Shipboard load energy estimation.

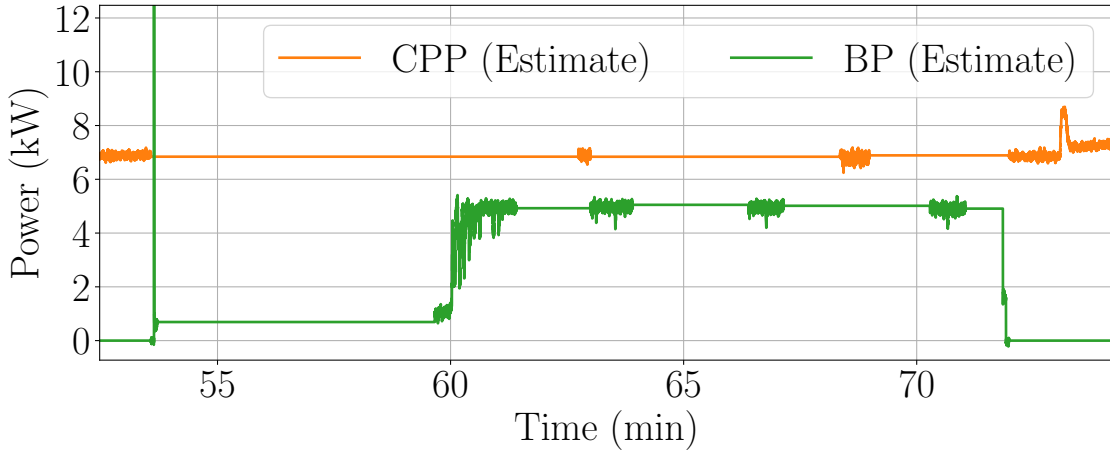
Table 5.4 shows the resulting energy estimates for the individual loads and the total energy. Again, the results are compared to rectangular integration. The proposed algorithm has excellent agreement with the aggregate ground truth for total energy. In contrast, there is a large underestimation of total energy when using rectangular integration. This is due to the large underestimation of the energy of the BP, since with the rectangular method the steady-state power is not updated after statistical events. This is better illustrated with a zoomed-in 30 minute view shown in Fig. 5-11. Here, it is clear to see that when the BP turns on, it is initially at a low steady-state value of approximately 1 kW. Then, about five minutes later it reaches a steady state of 5 kW after a period of stochastic behavior. This behavior was accurately identified

Table 5.4: Shipboard loads energy estimation.

Method	Estimated (kWh)			Total Error	
	CPP	BP	Total	Percent	RMSE
Proposed	13.600	4.392	17.992	0.13%	0.206 kW
Rectangular	13.987	1.839	15.827	-11.92 %	1.846 kW



(a) Aggregate power stream.



(b) Disaggregated power streams of the CPP and BP.

Figure 5-11: Shipboard load energy estimation zoomed-in view.

with the new statistical techniques.

The tracked durations of CPP and BP statistical events were 13.00 and 8.63 minutes, respectively. The duration of the CPP pump statistical events represents the working time of the relief valves in the hydraulic manifold. The relief valves regulate system pressure based on demand. Throttle commands that alter propeller blade

pitch require greater system pressure and places a greater demand on the system. During periods of low demand with no changes in pitch, hydraulic oil interacts with and opens the low pressure relief valve. During periods of high demand, the hydraulic oil alternatively interacts with and opens the higher pressure relief valve; corresponding to the observed statistical events. Estimates of valve working time could aid in maintenance decisions. Currently, CPP hydraulic system relief valves are tested on a fixed five-year cycle. BP statistical event detection could provide insight into pump health. With time, pump performance will inevitably deteriorate, for example, due to impeller wear. Decreased pump performance will cause changes in the time required to empty and fill storage tanks, and prime and clear air from the pump and associated piping. These changes will likely correspond to longer BP statistical events.

5.5 Chapter Summary

The results presented in this chapter demonstrate the ability to disaggregate stochastic power behavior using statistical features in real time. Statistical events are distinct from the main load on or off events and provide indication of changing load demand. Tracking statistical events can create an “automatic logbook” of power system behavior that was previously invisible using conventional nonintrusive monitoring techniques. The presented case studies showcase the applicability of the proposed method in different industrial sectors. Using both geometric and statistical classification techniques together can enhance load disaggregation abilities.

Chapter 6

Fault Detection and Diagnostics

Once individual loads have been identified from the aggregate stream, load behavior can be analyzed for fault detection and condition monitoring. Industrial loads and processes rely on materials that are susceptible to aging, corrosion, and mechanical failure [140, 141]. Since many systems operate under closed-loop control, subtle problems often go undetected. These problems, however, can increase energy consumption, impose excessive wear on electromechanical systems, and eventually result in complete failure. Some faults result in changes in run-time and run-frequency metrics, such as the graywater pump faults presented in Section 1.2. Sometimes, the power characteristics of a load can also change. The detection and tracking of drifting on-events was presented in Chapter 4. However, slow changes in power are difficult to track using the aggregate power stream if the load is continuously on. For example, gradual clogging or leaking in a ventilation system results in slow changes in power; however, these systems are often on for extended periods. With the additional information provided by slot harmonic tracking, the changes can be tracked and provide an indication of system health, as demonstrated in this chapter. Other times, the power characteristics of a load will change abruptly, such as the MPDE JW heater presented in Section 1.2. Abrupt changes may make it difficult to classify an event as the correct load, even with the adaptive tracking presented in Chapter 4. Additional features can inform classification. This chapter presents examples of fault scenarios in three-phase systems that can be characterized based on an understanding

of imbalances in the system.

6.1 Ventilation System Diagnostics

This section uses the custom nonintrusive power monitoring hardware presented in Section 2.3 and new signal processing techniques to disaggregate the slot harmonics of a collection of motors powered from a common electrical service. Signal processing identifies the harmonics associated with particular machines. Speed estimates developed from the observed slot harmonics can be used, with appropriate fan models, to estimate air flow in real time and identify blockages and leaks in air handling systems. This permits a single electrical monitor to evaluate the diagnostic condition of a collection of fans on a common electrical service. Field results are presented from USCGC Marlin. The work in this section was in collaboration with Joseph O’Connell and is presented in [142].

Ventilation systems provide fresh air and remove stagnant air for both habitable spaces and also machine processes like combustion cycles [143–145]. Efficient ventilation improves air quality and reduces disease transmission. Close environments require timely and effective maintenance. Degradation of airflow quantity or quality can occur for many reasons, including filter clogging, duct leakage, failed control devices like dampers or variable air volume plenums, and ingestion of foreign objects [143, 146]. Faults in ventilation systems not only degrade airflow quality, but can also lead to wasted energy. For example, the study in [147] found that 42% of the examined HVAC units had blockages or other interferences with airflow. Correcting the airflow brought 10% energy savings. Mission critical air handling systems may require sensors for pressure and flow not only for control but also for diagnostics. However, FDD techniques can be expensive due to the required number of sensors, and may still have difficulty in identifying root causes of faults [148].

Ventilation fans have performance curves that relate quantities like pressure and flow. System curves relate pressure and flow for proposed ventilation duct systems. During design, these curves inform decisions for selecting components. For example,

a fan’s operating point can be determined by the intersection between the system curve and fan pressure curve [149]. Many fault identification and detection schemes therefore employ rule-based methods that tabulate steady-state values for pressures and temperatures that indicate faults [147,150]. The electrical consumption of motors in an air handler can be observed and evaluated as a proxy that replaces many sensors for FDD applications. Rotor slot harmonics from motors in air handling systems can be used to track rotor speeds [47, 50].

6.1.1 Fan Physics

A fan curve relates shaft power to volume air flow. Therefore, a fan curve can be used to characterize power consumption changes that indicate faults like leaks or clogs. Because fans are often driven by induction motors, rotor slot harmonics can serve as indicators of fan blade speed. Since an induction motor’s torque-speed curve determines shaft power at any particular operating speed, fan speed is an effective practical metric for determining ventilation system health, and can be correlated with blockages and leaks through examination of a fan curve. In the typical slip or speed operating regime of an induction motor, power consumption and speed are monotonically correlated. That is, for an induction machine, low slip (high speed) operation consumes less power than high slip (lower speed) operation in the typical operating range of an induction motor. Motor or fan speed can therefore be used to identify a variety of system air flow faults. This interconnection between a fan and a motor and its utility for diagnostics are examined below.

6.1.1.1 Mechanical Characteristics

Ventilation systems are designed to supply specific air flow volume and pressure. System losses generate a system curve which relates pressure to airflow. The intersection between the system curve and a pressure-flow curve for the fan (“fan curve”) indicates an equilibrium operating point. The Air Movement and Control Association (AMCA) standard 210 describes how to generate the pressure-flow and power curves

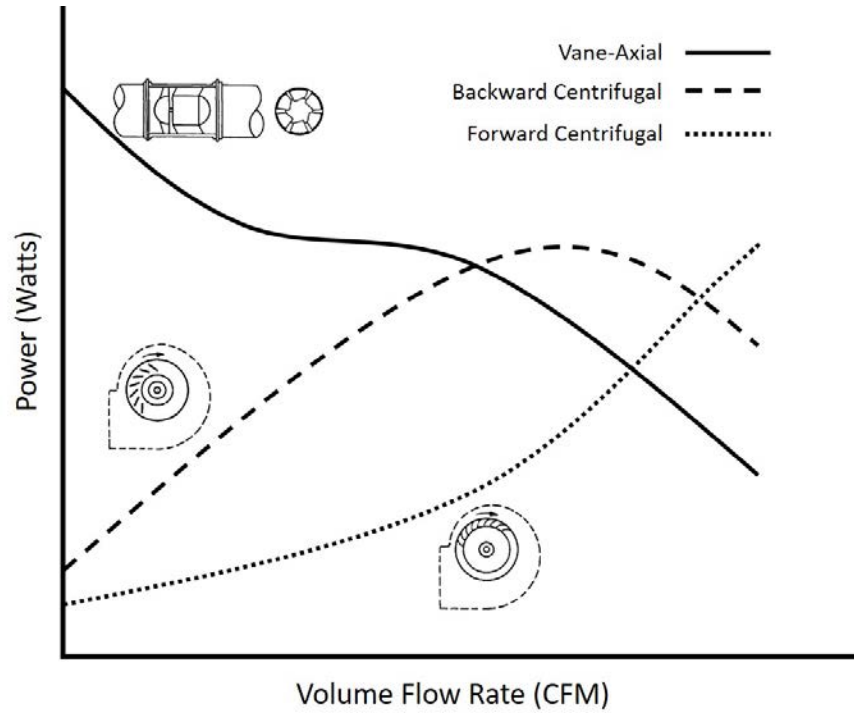


Figure 6-1: Characteristic fan power curves for various fans.

for a fan [151]. An analytical relationship links power H and air flow:

$$H = QP_tK \tag{6.1}$$

where Q is the volumetric flow rate, P_t is total pressure, and K is a thermodynamic inefficiency [91]. Flow Q and pressure P_t are non-linearly related. The simplicity of Eq. (6.1) hides the complex theoretical and real fan behaviors. Practical fan operation is understood not only with Eq. (6.1) but also using fan curves. Different types of fans have characteristic power curves, whose shapes are determined by the physical process employed to generate flow [152]. Three example power curves are shown in Fig. 6-1. There are two major categories of fans, centrifugal and axial, which are then divided further by blade type and shrouding. This section focuses on forward centrifugal and tube axial fans because they are common in commercial and industrial settings, including onboard USCGC Marlin.

For different fans operating at various points along their respective fan curves, leaky ducting or filter clogs have physically different effects. The impact of these

Table 6.1: Fault characteristics.

Fan Type	Normal Operating Point	Leaky Duct Power Impact	Clogged Filter Power Impact
Backward centrifugal	Right of power peak	Decrease	Increase
	At power peak	Decrease	Decrease
	Left of power peak	Increase	Decrease
Forward centrifugal	Any power	Increase	Decrease
Tube and vane axial	Any power (outside of stall region)	Decrease	Increase

faults on observed performance depends on the type of fan. Fan shaft power does not necessarily vary monotonically with flow rate. Leaky ducts tend to increase volume flow, and clogged filters tend to impede volume flow. The effects of these faults on shaft power can be determined from the fan power curves. Table 6.1 summarizes expected power changes for induction motors based on the fan curves for axial and centrifugal fans under flow changes.

Axial fans move air parallel to the axis of rotation. Due to their smaller sizes and lower costs, axial fans are frequently installed in small HVAC systems. Flow analysis reveals that the pressure across the impeller of these fans is driven by the decrease in relative velocity and a rise in absolute velocity [149]. The three main types of axial fans are propeller, tube axial, and vane axial. Some axial fans have a dip in their pressure-airflow curve, indicating a stall or surge region. This region is unstable and operation in this region can lead to failure [149]. Table 6.1 assumes that the axial fan is not operating in the stall region.

Centrifugal fans operate with air entering axially through inlets on either side of the main rotating vane drum, usually either open to atmosphere or inside of a duct. Inlet air is accelerated by the blades and discharged radially. Compared to axial fans, centrifugal fans are typically more efficient and produce high static pressure, however they are also more expensive. Centrifugal fans have different fan curves based on the blade type. For forward-curved blades, air flow and power rise in lockstep, with maximum power corresponding to maximum airflow. For backward-curved blades and airfoil-type centrifugal fans, the pressure curve peaks at the point of highest

efficiency, with a resulting peak in the power curve.

6.1.1.2 Electrical Tracking

The electrical characteristics of a motor driving a fan can be measured and used as an indicator of a fan's operating point and system health. For example, for induction machine drives without speed control, slip is usually restricted to a relatively small operating range. In this range, the shaft power level varies directly with slip for most induction motors. That is, a slower induction motor shaft speed usually corresponds to a higher shaft power and vice versa. Since shaft power varies directly with slip, motor speed indicates both shaft power and therefore volume flow as indicated by the fan curve for a fan driven by the motor.

Motor speed can be determined from observed electrical waveforms by tracking slot harmonics [47]. With the custom data acquisition hardware presented in Section 2.3, it is possible to track the slot harmonics of several different motors on a single aggregate electrical service. As was described in Section 2.3, slot harmonics are caused by rotor slots that create magnetic harmonics that effect phase currents, voltages and machine fluxes. The DAQ can be used to find these harmonics in observed current waveforms even though the harmonics are small compared to the base utility frequency currents feeding the motor or motors.

The use of both streams expands the possibilities for nonintrusively monitoring a collection of fan motors on a single electrical service. Power measurements or power changes may be due to a change in fan load and a fault. However, it may be difficult to associate a particular power change with a specific motor in a collection of machines. If the machines are different in other ways, for example, number of rotor bars, eccentricity, or nominal slip, then tracking slot harmonics can prove to be a valuable measurement for detecting a motor's speed uniquely in a collection of motors. The two approaches, power monitoring and speed tracking, offer some complimentary and some distinct advantages depending on the monitoring situation. The next two subsections examine the utility of power monitoring and slot harmonic tracking for fan and motor fault detection.

6.1.2 Fault Detection Through Power Consumption

Leaky ducting and clogged filters result in changes in power consumption. As a demonstration of the example predictions in Table 6.1 based on the example fan curves, a variety of fans were tested under different system conditions while monitoring power consumption and average air speed. All fans tested had an anemometer at their outlet to measure airflow. This section demonstrates these results and the utility of understanding fan curves and expected power changes for diagnostic monitoring. An axial fan was tested in a duct ventilation system which introduced a system curve or loss, while a forward-curved centrifugal fan was tested as a standalone air mover. To emulate blockages in the air system, filters were introduced on the intake side of the system. The filter elements consisted of varying stacks of a 1/2 inch polyester plastic filter, rated to MERV 5, or 3 micron permeability. Complete obstruction of the fan inlet was also emulated.

An SHT-30 ventilator, consisting of a tube axial fan and induction motor, was selected for its similarity in construction to the engine room exhaust and supply fans on USCGC Marlin. After completing gradual blockage with incremental additions of up to ten filters per intake, the fan was subjected to a complete obstruction. Fig. 6-2 shows average observed fan electrical power versus the average air speed for two minutes of fan operation. Error bars at each filter level show two standard deviations in both the increasing and decreasing directions. As expected for an axial fan, power increases as airflow decreases.

A second test fan was made on a forward centrifugal fan, chosen because it closely resembles USCGC Marlin's installed primary ventilation fan. The fan was operated for ten minutes at varying filter levels after which the fan was subjected to a complete obstruction. Fig. 6-3 plots the average steady state power versus the average air speed at each blockage level. As expected for a forward centrifugal fan, as airflow decreases the power also decreases. It is worthwhile to note that the two standard deviation error bars on the centrifugal fan are much smaller than the axial fan, due to lower levels of turbulence.

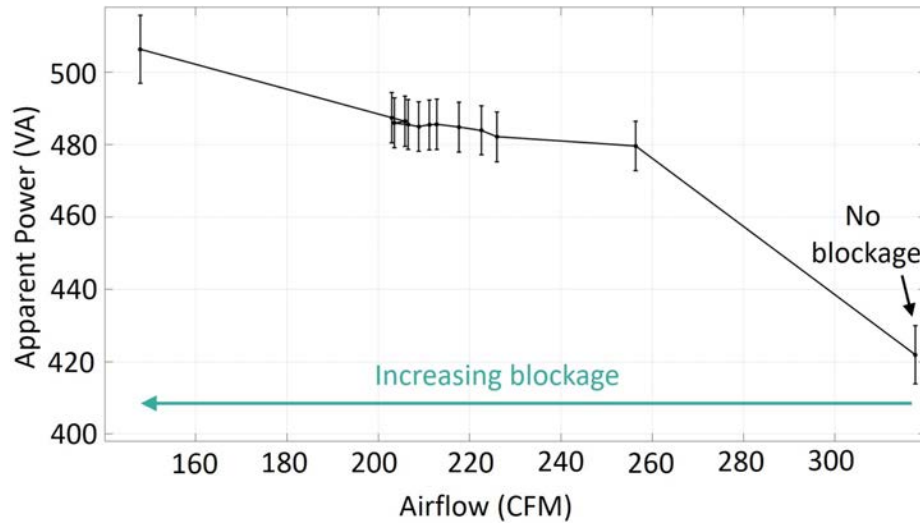


Figure 6-2: Axial fan power curve.

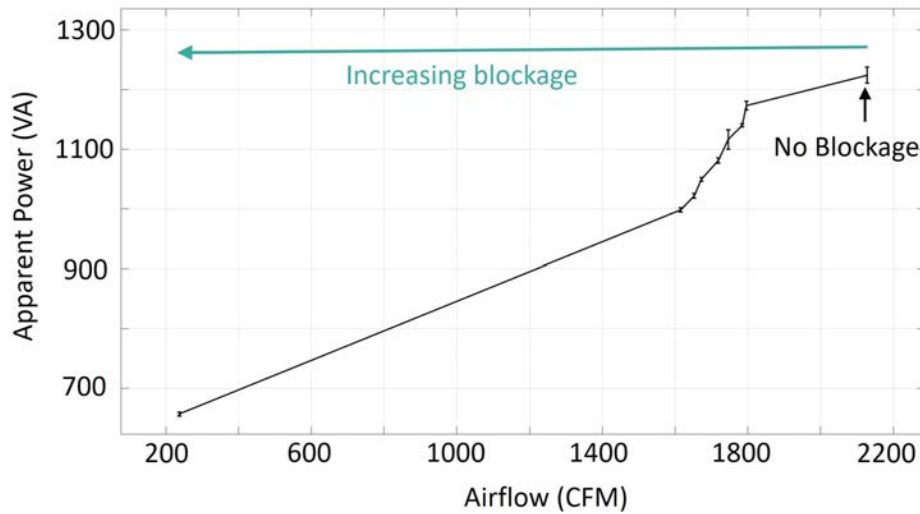


Figure 6-3: Centrifugal fan power curve.

6.1.3 Fault Detection Using Slot Harmonic Tracking

The slip for an induction machine is monotonically related to shaft power in the typical region of operation for a healthy motor operating from a utility with fixed frequency and voltage amplitude. Shaft speed is therefore a proxy for shaft power in these situations. Observed rotor slot harmonics in the current fed to a motor indicates operating speed, and can therefore serve as an indicator of the operating state of a shaft connected fan. In situations where several motors are present on an aggregate electrical service, motors and their mechanical state can be tracked if slot harmonics

can be identified that are unique to the machine of interest. Two challenges must be handled to enable nonintrusive slot harmonic tracking: compensation for variations in utility frequency and reliable identification of slot harmonics for a machine of interest in a “forest” of slot harmonics created by other machines on the utility service. Shifts in slot harmonics can be used to track changes in the mechanical fan system load, and also can be used to identify problems in the induction motor drive including broken rotor bars and other mechanical damage to the motor [153, 154]. This section presents a slot harmonic tracking algorithm applicable even when monitoring several motors. Then, experimental results are presented using an Agilent 6384B power supply. Using this controlled ac power supply permitted flexible experimentation with utility frequency variations.

6.1.3.1 Speed Estimation

The PSH frequency moves in a band limited by the minimum and maximum operating values for motor slip. For larger, efficient induction motors, slip is typically within a practical range of 5 to 10 percent [49]. Lower power machines, particularly in a fan application with plenty of air flow, may tolerate larger operating slips. Prior knowledge of the slip operating range is helpful to characterize an appropriate frequency window for finding relevant slot harmonics [8]. Various methods for rotor slot harmonic based speed estimation are reviewed in [155]. These include a frequency demodulation method [44], a maximum covariance method [45], and a short time least square Prony’s method [46]. However, all of these methods only consider a single induction machine, and are not applied in a multi-machine environment. Nonintrusive monitoring scenarios are likely to involve several motors operating from a single aggregate electrical service.

The DAQ developed for this work rejects large utility frequency components of current and permits the full resolution of an ADC fed by a high-gain amplifier to search for slot harmonic currents. A fast Fourier transform (FFT) analysis of observed electrical currents can be used to search for slot harmonics. However, because the resolution of the frequency discrimination and resulting speed estimation depends on

the size of the FFT bins, the accuracy of frequency estimation grows as the length of an observation window increases. This poses a dilemma. Long recording times provide extended data records that improve FFT frequency resolution. Shorter recording times are more likely to catch a quasi-static period of fan operation when the fan speed is essentially constant and free of mechanical disturbances. The method presented in [47] and [49] provides a balance of time and frequency resolution by finding a best-fit sinusoid within the FFT magnitude spectrum of relatively short windows of observed current. This section builds on this method for applicability in multi-machine environments.

This approach minimizes the sum of squared errors (SSE) between the FFT magnitude spectrum of a candidate sinewave and the observed FFT magnitude spectrum of the notch-filtered motor current in an observation window. Let the candidate sinewave be defined as $s(t) = \sin(2\pi f_n t)$ with frequency f_n and unity amplitude. The independent variable is the frequency f_n , which ranges from f_1 to f_2 . For a frequency window of length N_f , the vector of candidate frequencies is defined as:

$$f_n = f_1 + \frac{n}{N_f} (f_2 - f_1). \quad (6.2)$$

Increasing N_f increases the frequency resolution for finding the best-fit sinusoid. An exhaustive search is performed over all f_n and the frequency that minimizes the SSE is the best estimate in a squared-error sense for the location of the actual observed slot harmonic, and therefore the associated rotor speed. That is, the estimated principal slot harmonic frequency \hat{f}_{psh} (or more generally the slot harmonic frequency \hat{f}_h) is where the SSE is minimized:

$$\hat{f}_{psh} = \arg \min_{f_n} \text{SSE}(f_n). \quad (6.3)$$

For f_n ranging from f_1 to f_2 and corresponding frequency indices k ranging from k_1 to k_2 , the SSE is defined as:

$$\text{SSE}(f_n) = \sum_{k=k_1}^{k_2} |C(f_n) \cdot |X_{s,k}(f_n)| - |X_{i,k}||^2 \quad (6.4)$$

where $X_{i,k}$ is the FFT spectrum of the notch-filtered motor current, $X_{s,k}$ is the FFT spectrum of a unity amplitude sine wave candidate of frequency f_n , and C is a normalizing scale factor given as:

$$C(f_n) = \sqrt{\frac{\sum_k |X_{i,k}|^2}{\sum_k |X_{s,k}(f_n)|^2}}. \quad (6.5)$$

So far, this method has assumed that the slot harmonic of interest is the largest peak in the window. This assumption may not hold in a multi-motor environment. Because the power information from a NILM detects when each and every motor in an installation is energized, the NILM is aware of the operation of multiple motors. In this situation, a multi-peak detection algorithm is proposed that builds on the SSE method in Eqs. (6.2)-(6.5). First, the minimum SSE sinusoid with a candidate sinewave as described above is determined with frequency f_m . This is the location of the slot harmonic of the machine with the largest peak in the window. Then, for a two-machine environment, to find the location of the slot harmonic of the second machine, a candidate signal is defined as the sum of two sinusoids,

$$s(t) = A\sin(2\pi f_n t) + \sin(2\pi f_m t), \quad A \leq 1, \quad (6.6)$$

where A is a scale factor that is less than one. The location of the slot harmonic of the second motor is defined as the frequency that minimizes the SSE, with $X_{s,k}$ from Eq. (6.4) and Eq. (6.5) being the FFT spectrum of Eq. (6.6). If a window contains the slot harmonics of more than two motors, this process can be repeated with candidate signals defined as the sum of three (or more) sinusoids, as necessary. This enhanced method is demonstrated in field testing in Section 6.1.4. From the estimated PSH frequency \hat{f}_{psh} the rotor speed and slip can be estimated. The estimated rotor speed in rpm is:

$$\hat{\omega}_{rpm} = \frac{60}{R} (\hat{f}_{psh} - f_s). \quad (6.7)$$

The estimated slip (also described in Section 2.3) is:

$$\hat{s} = 1 - \frac{p}{R} \left(\frac{\hat{f}_{psh} - f_s}{f_s} \right). \quad (6.8)$$

6.1.3.2 Speed Estimation with Filter Blockages

To demonstrate fan speed estimation using motor slot harmonics, the centrifugal fan system was observed with various filter blockages when operating from a nominal 60 Hz utility frequency. The Agilent 6384B power supply was configured for 120 V rms output and 10 A rms current limit. The motor powering the centrifugal fan is a single-phase permanent split-capacitor (PSC) motor [156, 157]. PSC motors are commonly used in residential and commercial air handling unit blowers [158]. Speed selection is implemented with a tapped armature winding method, in which the stator winding serves as an autotransformer [157]. The lowest speed operation was selected to provide the greatest slip for the machine and therefore the greatest challenge to the speed estimation window algorithm. In this low-speed setting, the line voltage is applied to the full number of turns on the motor armature, producing the lowest machine flux and operating speed. When running at “low” speed the fan operates at roughly 30% slip.

Strictly for cross-validation and initial setting of the slot harmonic search window location in frequency, the fan speed was measured with a tachometer. Using the measured rotational speed, slip was calculated using Eq. (2.24). Slot harmonics for different values of v from Eq. (2.23) differ by $2f_s$, making $2f_s$ a good window size for locating the slot harmonic of interest [47]. Because of the high slip of this machine, filter blockages have a large effect on the motor speed. For these high slip machines, a window of length $2f_s$ that is centered on the nominal slot harmonic was used to begin the slot harmonic search:

$$f_1 = f_{prev} - f_s \leq f \leq f_{prev} + f_s = f_2 \quad (6.9)$$

where f_{prev} is the frequency of the initial or last detected slot harmonic location.

Table 6.2: Centrifugal fan speed at 60 Hz.

Filters	Measured		Estimated			Error
	ω_{rpm}	s	\hat{f}_{psh} (Hz)	$\hat{\omega}_{rpm}$	\hat{s}	ω_{rpm} (%)
0	1196.4	0.335	1016.8	1196.0	0.336	0.03
1	1262.4	0.299	1064.3	1255.4	0.303	0.56
2	1292.6	0.281	1094.3	1292.9	0.282	0.02
3	1319.4	0.267	1115.5	1319.4	0.267	0.00
4	1351.6	0.249	1138.5	1348.1	0.251	0.26
5	1353.2	0.248	1142.3	1352.9	0.248	0.02
6	1386.6	0.230	1170.6	1388.3	0.229	0.12
7	1408.6	0.217	1186.3	1407.9	0.218	0.05

This adjustable window and Eqs. (6.2)-(6.5) were used to track slot harmonics on one-second length windows with $N_f=1200$ (0.1 Hz frequency resolution). The initial window center for the test motor was 1020 Hz, which was the approximate observed slot harmonic location. Then Eqs. (6.7) and (6.8) were used to calculate the estimated speed and slip, respectively, with results presented in Table 6.2. The estimated speed was then compared with the measured tachometer speed for cross-validation, as shown in Table 6.2 as absolute percentage error. The estimated speed and estimated slip closely match those calculated from the measured tachometer speed, demonstrating that slot harmonics were accurately tracked and fan speed can be reasonably measured using slot harmonics. For this centrifugal fan, the fan speed increases as expected as the system experiences progressive vent blockages, and the slot harmonic approach can clearly detect faults such as vent blockage and duct leaks.

6.1.3.3 Speed Estimation With Varying Supply Frequencies

Particularly on microgrids with limited generation capacity and finite inertia, as are found in marine environments, there are potentially significant fluctuations in supply frequency. While these fluctuations may not affect the overall operation of the microgrid, the slot harmonic Eq. (2.23) effectively applies a “gain” to the utility base frequency as a function of the slip and construction of the induction motor. Relatively small variations in utility base frequency will therefore shift the location of slot

Table 6.3: Centrifugal fan speed with variable supply frequency.

f_s	Measured		Estimated			Error
	ω_{rpm}	s	\hat{f}_{psh} (Hz)	$\hat{\omega}_{rpm}$	\hat{s}	ω_{rpm} (%)
59.5	1214.6	0.319	1031.2	1214.0	0.320	0.05
59.6	1212.8	0.322	1027.8	1209.8	0.323	0.25
59.7	1211.8	0.324	1027.3	1209.1	0.325	0.22
59.8	1208.6	0.326	1025.2	1206.5	0.327	0.17
59.9	1206.6	0.329	1025.2	1206.5	0.329	0.01
60.0	1205.2	0.330	1023.3	1204.1	0.331	0.09
60.1	1204.4	0.332	1022.7	1203.4	0.333	0.09
60.2	1202.0	0.334	1019.3	1199.1	0.336	0.24
60.3	1200.4	0.336	1019.8	1199.8	0.337	0.05
60.4	1199.6	0.338	1017.7	1197.1	0.340	0.21
60.5	1196.2	0.341	1015.6	1194.5	0.342	0.14

harmonics, and complicate efforts to track motor speed. These frequency shifts, if not accounted for can be misinterpreted as faults. It is therefore essential to track supply frequency and incorporate the real time utility frequency into the slot harmonic calculations.

To explore the impacts of slight variations in supply frequency, the Agilent power supply energizing the fan motor was varied from 59.5 to 60.5 Hz in 0.1 Hz steps. The NILM operating with the enhanced DAQ uses its observations of voltage to provide a running estimate of utility frequency on a cycle-to-cycle basis. This estimate is used to compensate the slot harmonic search. The slot harmonics were tracked on one-second length windows with a frequency window of length $2f_s$ centered at 1020 Hz and $N_f=1200$ (0.1 Hz frequency resolution). Results from these experiments are shown in Table 6.3, with measured tachometer motor speed, slip, observed PSH, estimated speed, estimated slip, and absolute percentage error of speed. The results indicate that NILM slot harmonic tracking can effectively monitor fan rotor speed at different supply frequencies. Additional information, such as power consumption and equipment status from the NILM, can also be used to evaluate if a ventilation system is degraded. Since a fan's power consumption is relatively independent of small changes in the variable supply frequency, and as demonstrated in Section 6.1.2

can be used to detect ventilation blockages, this information can be examined in tandem with slot harmonic shifts to determine HVAC health.

6.1.3.4 Multi-Motor Frequency Space Speed Estimation

Nonintrusive monitoring scenarios are likely to involve several motors operating from a single aggregate electrical service. This situation is common, for example, with microgrids on ships. Searching for slot harmonics has proven to be of great utility in distinguishing the operation of different motors on a common electrical service. To illustrate, the previously discussed centrifugal fan was operated on the same electrical service with a three-phase axial fan and a secondary single phase centrifugal fan (bathroom fan). All three machines were powered from the Agilent ac power supply, and the observed slot harmonics were used to approximate rotor speed on all operated machines. These comparison tests were conducted at 59.5, 60.0 and 60.5 Hz to emulate the anticipated changes in supply frequency onboard a marine microgrid.

The motors inject a family of harmonics on to the line according to the slot harmonic equations and the inevitable imbalances in the machines that excite eccentricity harmonics. For this demonstration, the three motors were tracked with harmonics intentionally chosen relatively close to one another to reflect a potential “worst-case” scenario, in which two machines produce similar harmonics, increasing the potential for incorrect identification. With knowledge of the machine constructions, the machines can be tracked and their operation distinguished from the background loads. This can be seen in Fig. 6-4, which shows the harmonics of the three fans shifting in frequency from 59.5 Hz to 60.0 Hz to 60.5 Hz. In this scenario the harmonics maintain sufficient frequency separation such that no overlap occurs.

Fig. 6-4 also shows how different types of induction machines may react differently to changes in supply frequency. The centrifugal fan slows down as supply frequency increases while the secondary centrifugal fan and three-phase fan both increase speed. The centrifugal fan is powered by a single-phase capacitor run induction machine, and the shift in supply frequency alters the torque-speed curve, such that the constant torque applied by the fan blades counteracts the shift in the new endpoint. Despite

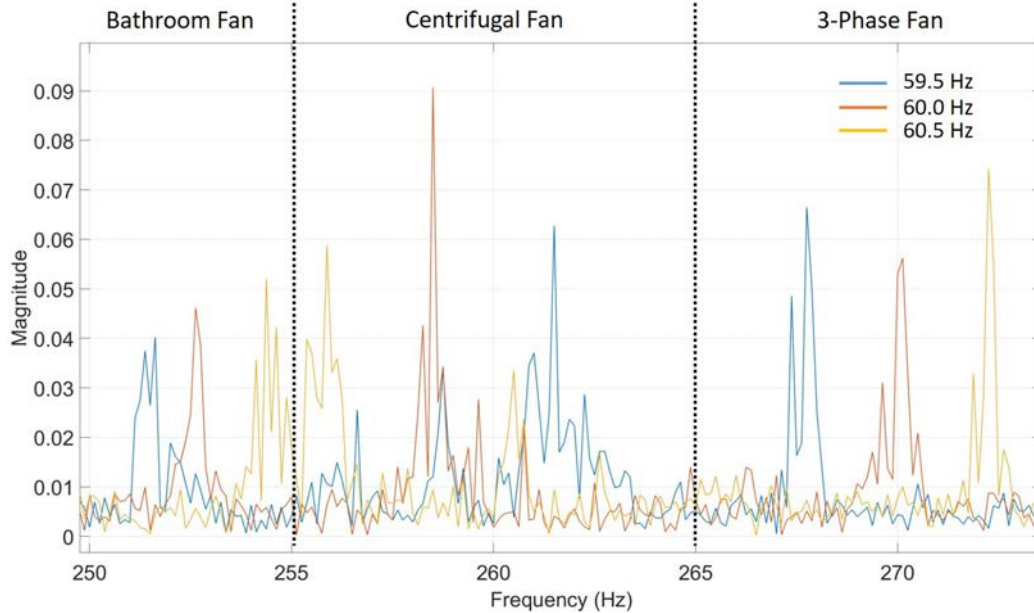


Figure 6-4: Frequency space with three induction machines at various supply frequencies.

variations in the supply frequency and differences in the torque-speed behaviors, the observed slot harmonics for each machine remain in distinct windows that can be reliably used to track motor operation. Specific knowledge of slot harmonic location and physical attributes of the motor can be leveraged to differentiate between harmonics, even as a power grid experiences frequency shifts or induction machines change speed due to mechanical operating conditions.

It is important to note that slot harmonic tracking in a multi-machine environment relies on the slot harmonic current signals being present in the NILM current measurements. Since the NILM is remote from any particular load, the current signals measured by a NILM depend on connected loads and the source impedance. It is possible for a parallel load to provide a low-impedance path at the slot harmonic frequency, attenuating the slot harmonic signal measured by a NILM. For example, passive single-tuned and double-tuned filters are commonly used for reducing specific harmonic currents. These filters are typically placed close to the harmonic producing loads, such as power-electronic loads. Thus, passive filters can shunt the slot harmonics of interest away from a NILM. It is important to be aware when such loads

are online, so that correct conclusions can be drawn.

6.1.3.5 Rotor Bar Damage

Finally, we observe that slot harmonic information can also be used to diagnose faults in the motor itself, distinct from changes in the fan system. For example, rotor bar cracking [159] can occur due to thermal cycling of a motor or mechanical stress under starting [160]. It is important to distinguish gradual shifts in slot harmonics that are due to a gradually leaking duct or clogging filter versus a change due to a broken rotor bar. To demonstrate the utility of slot harmonics observations in distinguishing different faults, two identical fans were tested, one healthy and one with a single cracked rotor bar. The tachometer-measured motor speed of the healthy motor was 1069.2 rpm, which gives a slip of 0.406. With a broken rotor bar, the expected location of the slot harmonic for the motor moves, and can be predicted by:

$$f_{brb} = f_s \left[(kR + n_d) \frac{1-s}{p} + v - 2s \right], \quad (6.10)$$

which indicates a predicted slot harmonic at $2f_s s$ lower than the healthy slot harmonic location [159]. Using Eq. (6.10) and the slip, the predicted shift in the slot harmonic is 48.7 Hz, essentially identical to the observed shift of 48.5 Hz shown in Fig. 6-5. A nonintrusive monitor with the enhanced DAQ for tracking slot harmonics, aware of the activity of a motor from power traces, can detect the slot harmonic shift due to a broken rotor bar.

6.1.4 Field Testing on USCGC Marlin

This section demonstrates speed estimation and diagnostics onboard USCGC Marlin. There are two different scenarios in terms of frequency stability: in-port (connected to land based shore-power) and at sea (on generator power). The frequency for a typical day in-port and a typical day at sea are compared in Fig. 6-6. In this comparison, the in-port and at sea frequencies had ranges of 0.15 Hz and 2.42 Hz, respectively.

USCGC Marlin has two primary ventilation systems installed, one for the engine

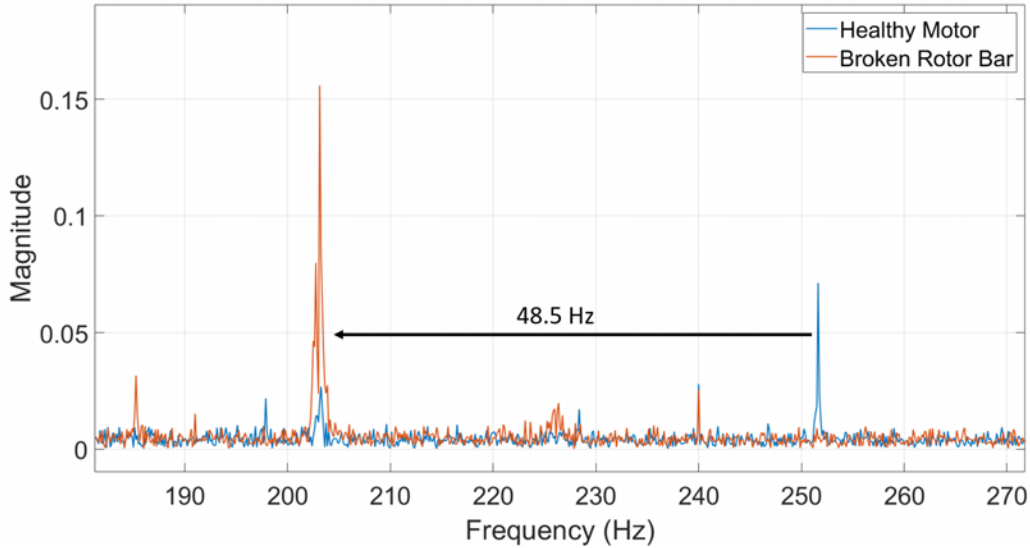


Figure 6-5: Slot harmonic location shift due to rotor bar damage.

Table 6.4: Marlin supply fan speed estimates at various blockage levels

Blockage	\hat{f}_{psh} (Hz)	$\hat{\omega}_{rpm}$	\hat{s}
No Blockage	1354.5	3530.46	0.0193
1 Filter	1354.5	3530.46	0.0193
2 Filters	1354.6	3530.73	0.0192
3 Filters	1354.7	3531.00	0.0192
1 Inlet Blocked	1355.6	3533.46	0.0185
2 Inlets Blocked	1358.6	3541.64	0.0162

room and the other for living spaces. Testing was conducted on the supply system for the primary habitable space. This fan system uses a belted centrifugal fan drawing air from both the weatherdeck and the internal superstructure. Fig. 6-7 shows the supply fan on USCGC Marlin. Speed estimation during ventilation blockage and a multi-motor environment are demonstrated while the ship is connected to shore power. Then, speed estimation is demonstrated while the ship is at sea with a changing supply frequency.

6.1.4.1 Primary Supply Fan

The primary supply fan employs a two-pole induction motor with 22 rotor bars rated for 3530 rpm at 60 Hz. To test the primary supply fan, filters were introduced simul-

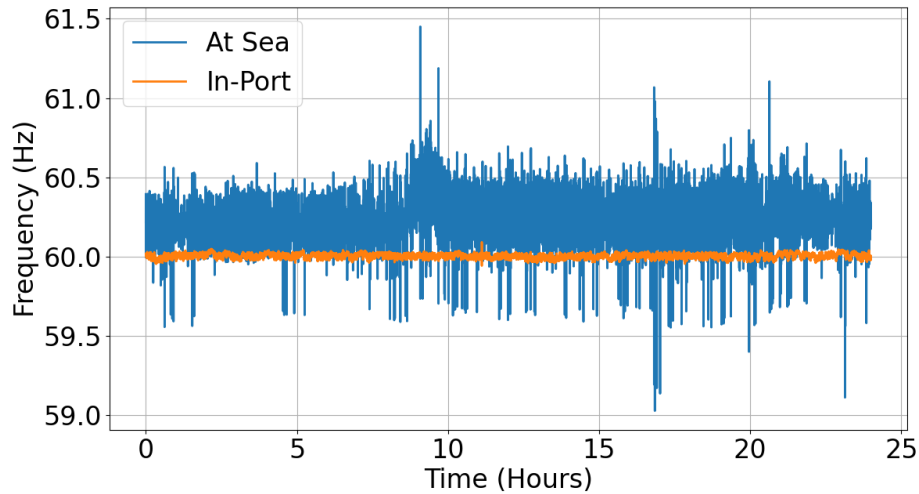


Figure 6-6: Supply frequency for typical day at sea on generator power and in-port on land-based utility power.

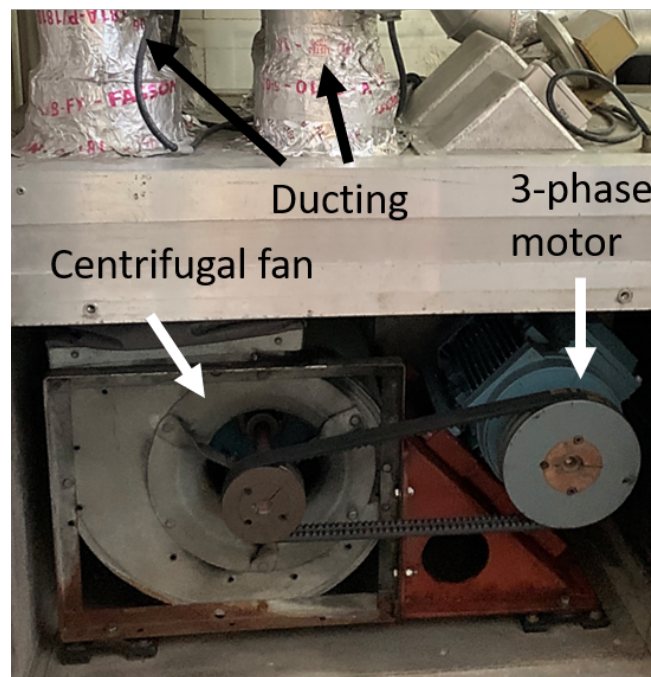


Figure 6-7: Supply fan on USCGC Marlin.

taneously on both the weatherdeck “fresh” air intake and the internal re-circulation intake. Full blockage on the system caused the steady-state power of the motor to drop noticeably, as expected from Table 6.1. The motor PSH shifts as well, indicating a motor speed increase. This is similar to the observed behavior of the lab tested centrifugal fan. The SSE slot harmonic detection method in Eqs. (6.2)-(6.5)

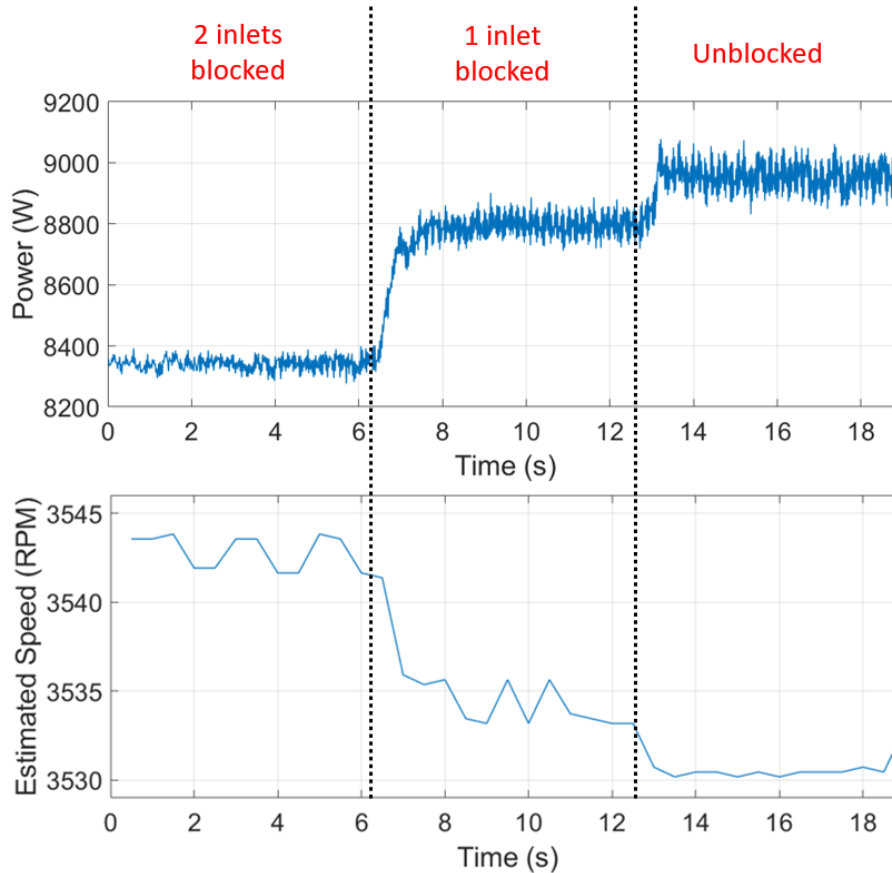


Figure 6-8: Measured power and estimated speed for the supply fan with various blockage levels.

were used to nonintrusively track motor speed. Table 6.4 shows the estimated PSH location and motor speed and slip from the slot harmonic estimation with one-second long windows and search parameters $f_1 = 1335$ Hz, $f_2 = 1375$ Hz and $N_f=300$ (0.1 Hz frequency resolution).

Fig. 6-8 shows the aggregate three-phase real power stream and the estimated speed as blockages are subsequently removed. Under normal shipboard operations this supply fan is infrequently energized/secured and any clogs or leaks would happen as the fan is operating. Although the inlet blocked conditions are visible in the aggregate power stream, it would be difficult to attribute the changes in power to the supply fan when other loads are energized and consuming power. Tracking the motor PSH unambiguously identifies clogging on the ship.

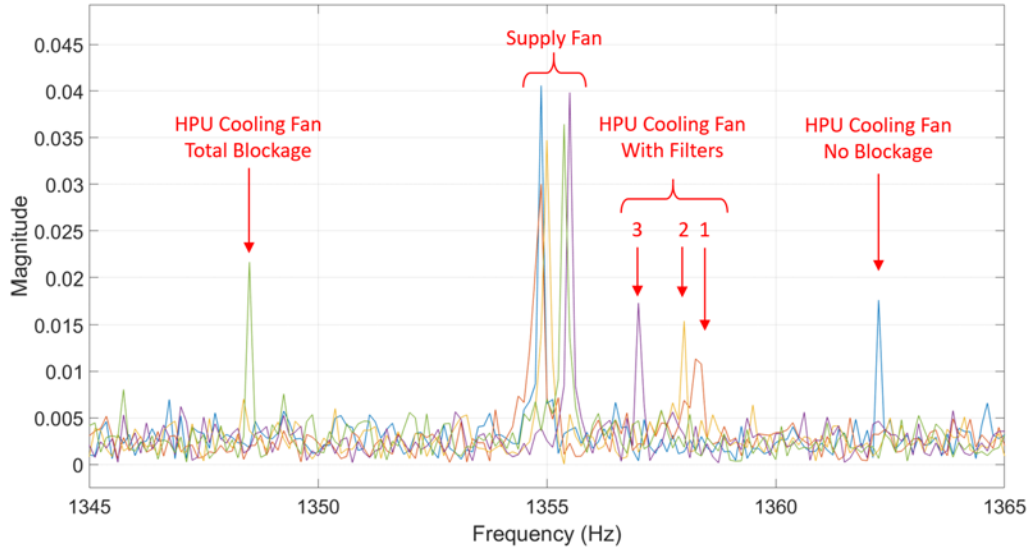


Figure 6-9: Marlin multi-motor environment.

6.1.4.2 Marlin’s Multi-Motor Environment

In addition to the primary supply fan on Marlin, there is a hydraulic power unit (HPU) cooling fan, which draws power from the same monitored electrical grid on the ship. This cooling fan is a three-phase axial fan, similar to the three-phase axial fan lab-tested in Section 6.1.3. No data was acquired about the construction of the HPU cooling fan, e.g., number of rotor bars. Instead, the slot harmonics for the cooling fan were located and tracked by running the HPU cooling fan while introducing filter blockages to slow down the fan, demonstrating that the slot harmonic tracking can be configured *ad hoc* on a field NILM. In contrast to the earlier observed centrifugal fans, this axial fan slowed down as expected as filters were introduced. This is due to an increase in torque from larger pressure differentials across the blades [149]. Up to three layers of 3 micron permeability filters were added, as well as total blockage. To “train” the NILM to recognize these slot harmonics with no other information about the machine, the HPU was energized during a period when the ship grid had other motors “off.” This ability to energize, secure and physically decelerate or accelerate the fan of interest can serve as a replacement for knowledge regarding the motor’s internal construction. This method illustrates an important second method for identifying harmonics, when limited information is available about the machine of interest.

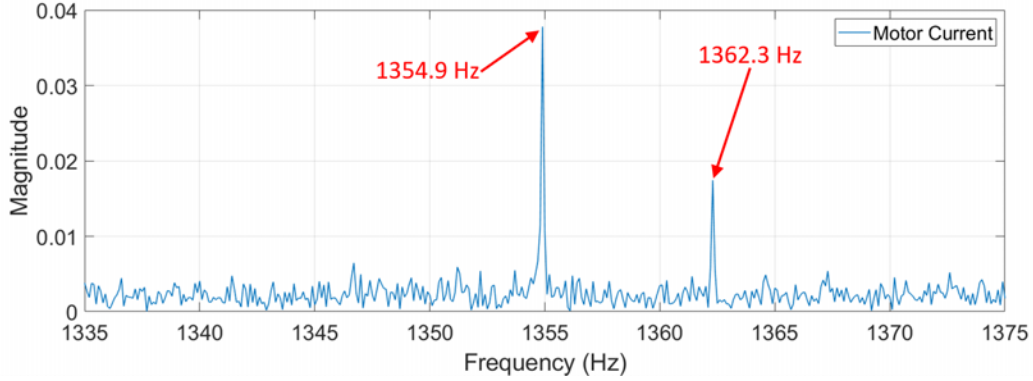
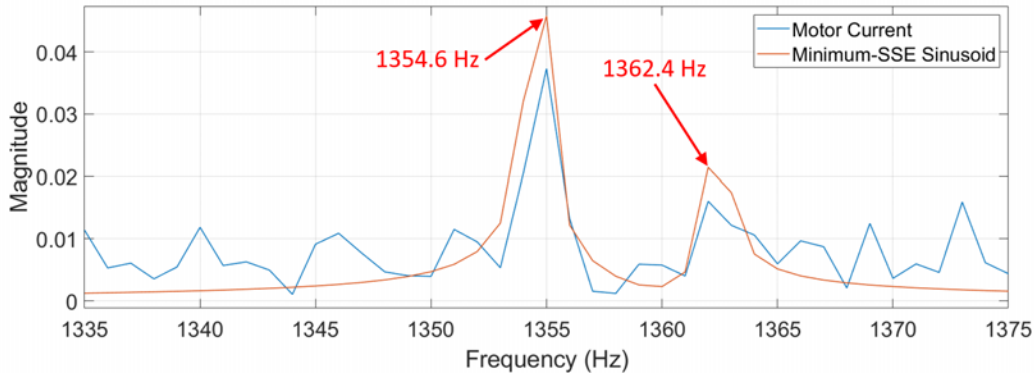
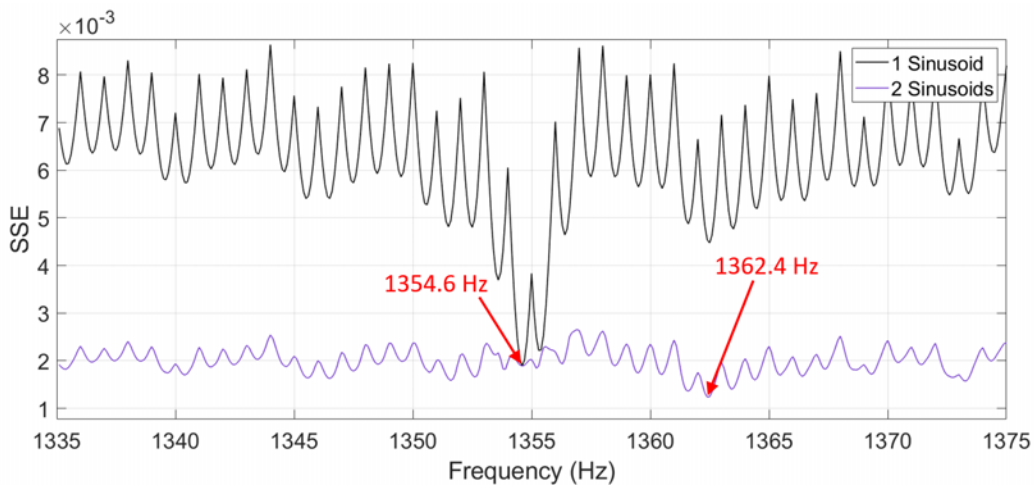


Figure 6-10: FFT current spectrum for the motors for a 10 second window.

Once the slot harmonic window for the HPU cooling fan was identified, a multi-motor environment was tested, in which both the supply fan and HPU cooling fan were energized. The supply fan was kept at “healthy” operation, while filter blockages were introduced for the HPU cooling fan. The slow down of the HPU cooling fan as filters are introduced causes the slot harmonic to move towards the supply fan’s slot harmonic. Eventually, the fully blocked axial fan’s slot harmonic travels past the supply fan’s harmonic location. The frequency spectrum of this multi-motor environment in various operating conditions is shown in Fig. 6-9. In order to track the respective slot harmonics of each fan, the multi-peak detection SSE method from Eq. (6.6) was essential. Here, $f_1 = 1335$ Hz, $f_2 = 1375$ Hz and $N_f=300$ (0.1 Hz frequency resolution). For Eq. (6.6), A is set to amplitudes ranging from 0.4 to 1 in 0.1 increments. Fig. 6-10 shows the spectrum for a ten-second window, with the peak values. In comparison, Fig. 6-11a shows the spectrum for a one-second window and the minimum-SSE sinusoid. These detected frequency peaks from the minimum-SSE sinusoid method match closely with the estimates from the longer ten-second window, which offered higher frequency resolution. Fig. 6-11b shows the minimum SSE from using a single sinusoid, versus the multi-peak method with two sinusoids. These results show that even in a multi-motor environment, slot harmonics for machines of interest can be detected and tracked. Using transient detection in the NILM power stream to determine when a load is operating can inform the slot harmonic detection algorithm as to when multi-motor slot detection is necessary.



(a) FFT current spectrum for the motors and minimum-SSE sinusoids.



(b) SSEs according to one and two sinusoids.

Figure 6-11: Current spectrum and SSEs for a 1 second window.

6.1.4.3 USCGC Marlin At Sea

USCGC Marlin was observed during an underway period at sea. During this time, the installed NILM monitored supply frequency and both filtered and unfiltered current data. The supply fan was energized during the entire underway period, presenting the fan with various environmental conditions and supply frequencies. The NILM used the presented algorithms to track the supply fan slot harmonics. Supply frequency changes can impact the rotor speed as well as slot harmonic location. Fig. 6-12 shows the estimated speed from the tracked slot harmonic as the supply frequency fluctuates from a maximum of 60.5 Hz to a minimum of 59.9 Hz, using the same method as described in Section 6.1.4.1. As observed, the rotor speed decreases slightly from an

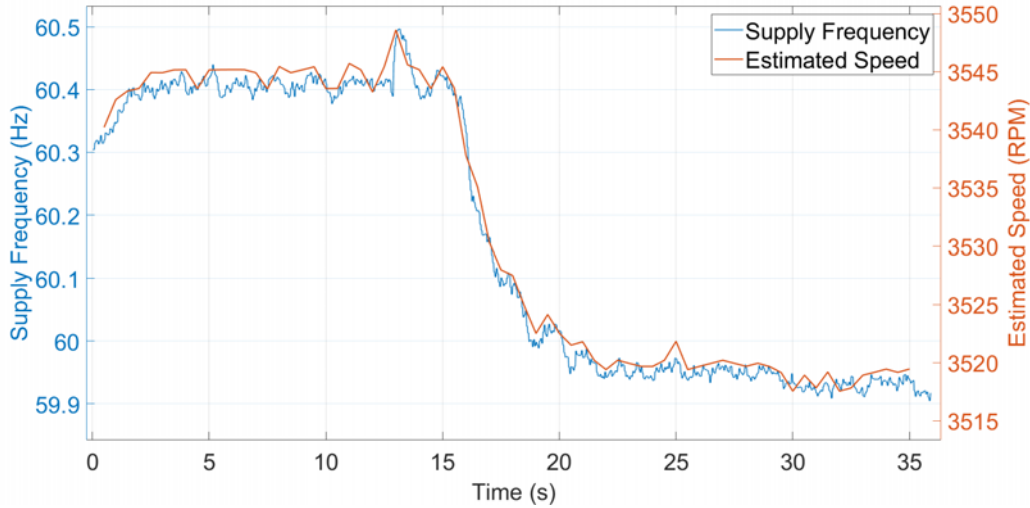


Figure 6-12: USCGC Marlin estimated speed with shifting supply frequency.

estimated 3549 rpm to 3518 rpm as the ship’s grid frequency decreases. Analysis of this data supports the conclusions drawn in Section 6.1.3, which indicate that slot harmonics can be tracked with a varying utility frequency when the NILM tracks utility frequency from the system voltage.

6.2 Three-phase Equipment Condition Monitoring

This section considers additional challenges that arise for load identification and diagnostics in three-phase systems, in part due to potential imbalances in the system. Features particularly useful for FDD in these systems are proposed. With correct interpretation, data from a NILM can identify healthy versus faulty behavior and supplement existing protection schemes. Machine learning systems essentially seek correlations between input data and training classifications or categories [161]. The presentation of input data characteristics, i.e., the “feature space” used to describe a problem or data set, can greatly affect training effort, operating speed, and identification accuracy [162]. The training data for a nonintrusive monitoring classifier is unlikely to contain instances of possible fault conditions. Given a limited training dataset, it is crucial for the feature space to be physically-informed. An understanding of the physical mechanisms for possible faults can be used to inform a feature

space useful for nonintrusive load FDD.

A three-phase system can have an unbalanced generation system (i.e., unbalanced supply voltages) and unbalanced loads (i.e., loads are not evenly distributed among the phases). Unbalanced voltages can have detrimental impacts on connected equipment and result in deviations in expected load behavior [163]. An unbalanced load system is also a power quality concern; however, load imbalance does not necessarily indicate improper individual load operation. A healthy load is one that is operating according to design specifications. A healthy load does not imply a balanced load system, and a balanced load is not necessarily a healthy load. This section presents analysis for attributing load transient behavior, such as when a load energizes or changes state, to healthy or faulty load operation. Changes in system balances for currents, voltages, and harmonic content, particularly during load turn on and turn off transients, form a particularly useful feature space for FDD. Changes in balances that occur during transients tend to be associated with the particular load change under observation and can be distinguished from an arbitrary background imbalance in the power system. Examples are illustrated with measurements made on several USCG marine microgrids. Ungrounded delta systems are a common configuration on shipboard microgrids and other microgrids in which maintaining a high degree of reliability and continuity is necessary [164, 165].

There are various power system configurations, including single-phase, split-phase (single-phase three wire), and three-phase. Three-phase systems can be three-wire, four-wire, or five-wire, depending on the neutral and ground connections. There are also configurations such as a high-leg delta. A NILM can monitor any of these systems. However, measurements are also limited by physical access and convenience. An example that illustrates this is a three-wire delta system. For instance, measurements of phase currents (i.e., the current through the load) of delta-connected loads are unlikely to be made nonintrusively. It is more likely to have nonintrusive access to line currents (i.e., the currents in the line conductors). For FDD it is useful to have all three line currents, e.g., if there are current imbalances or a bonding fault. However, there is valuable reexamination to be done for considering the feature space that is

best suited for connecting measured power and harmonic consumption to physical performance. The ideas discussed in this section can be generalized to other utility configurations with a properly configured NILM with relevant nonintrusive electrical measurements.

6.2.1 Load Imbalance Feature Space

Pattern classifiers need to disaggregate both healthy and faulted loads from the aggregate power stream. An edge detector [85] is used to find load events, such as load on and off events and transitions between discrete states. At each event, a set of extracted features is used for load identification [6]. The preprocessing in Section 2.1 computes the power and higher-order current harmonics for each phase (i.e., a, b, c), based on the line current measurements. Thus, features extracted from the per phase power and higher-order current harmonics, such as steady-state and inrush characteristics, would be an intuitive choice of features for load identification [6]. However, when a load experiences gradual degradation or a sudden fault, the following questions should be considered when assessing the usefulness of a feature space for load diagnostics:

- How do the changes manifest in the feature space?
- Are the changes in the feature space physically interpretable?
- Is the load still uniquely identifiable?

These questions are especially important in a three-phase system because of the inherent imbalances of certain load and fault types. Without a physics-informed feature space, it is difficult to correctly attribute imbalances to the correct load. In the examples presented in this section, the loads are connected line-to-line. Thus, the power consumption on each delta phase (i.e., ab, bc, ca) may be more intuitive for interpreting load changes. Computing the power per delta phase requires knowing the phase currents. However, from the line currents of a delta-connected system, and without making any further assumptions, it is not possible to calculate the true delta phase

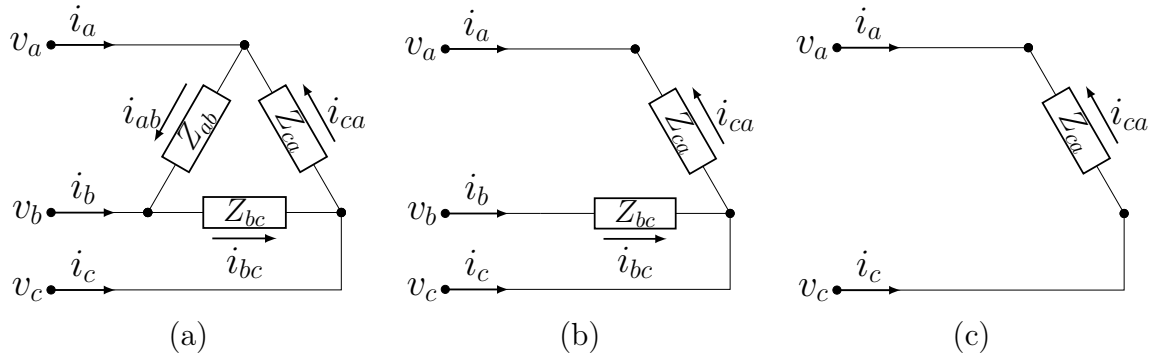
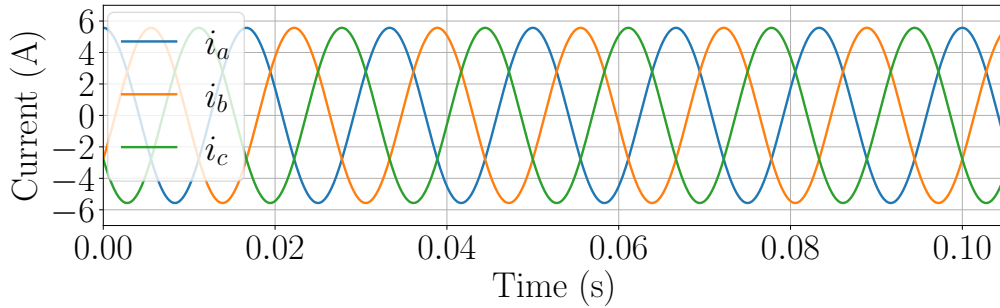


Figure 6-13: Conceptual diagrams for three possible load connection types in a delta-configured system: (a) three delta phases, (b) two delta phases, and (c) single delta phase, with line currents i_a , i_b , i_c and phase currents, i_{ab} , i_{bc} , and i_{ca} .

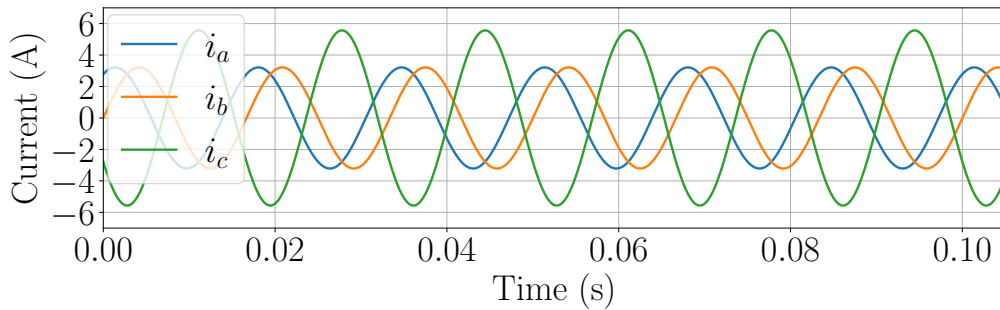
currents. The line currents are governed by the following relationships, where the matrix has rank two, indicating it is not invertible:

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} i_{ab} \\ i_{bc} \\ i_{ca} \end{bmatrix}. \quad (6.11)$$

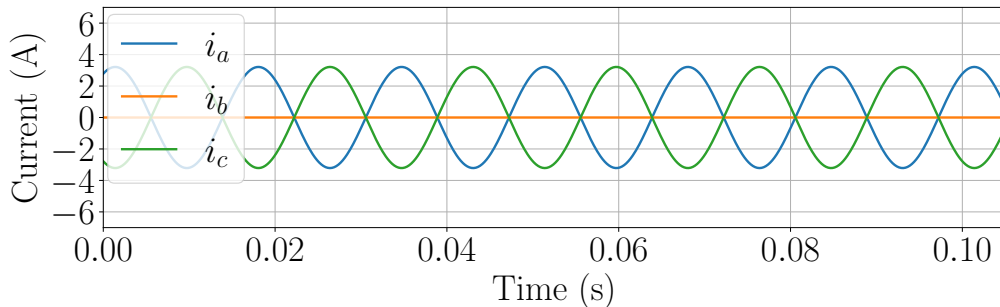
To resolve the non-uniqueness of possible phase currents, this work proposes a feature set based on an assumption that loads are connected in one of three ways in the monitored delta-configured system: three delta phases, two delta phases, or a single delta phase. Every step change in power can first be evaluated to determine if the relative balance or imbalance is consistent with that of the expected load connection types. In this work, a transient that results in imbalances consistent with that of a load connected across two delta phases is considered a fault condition of three-phase load, such that one of the delta phases open-circuited. A load identified as connected on a single delta phase can either be a normally operating single-phase load or a fault condition of a three-phase load. Conceptual diagrams for examples of these three connection types are shown in Fig. 6-13. Example line currents are shown in Fig. 6-14 for resistances of 193.6Ω (corresponding to 1 kW per delta phase on a 440/254 V system). The three-phase load has three equal magnitude line currents with a 120° phase offset between each. The two-phase load has one unaffected line current and



(a) Three delta phases.



(b) Two delta phases.



(c) Single delta phase.

Figure 6-14: Line currents for three different load connections in a delta-connected system.

two line currents with reduced magnitudes and phase shifts. The single-phase load has two reduced magnitude line currents which are 180° out of phase from each other and one line current that is zero.

With this assumption, the ratios of fundamental line current spectral envelope magnitudes can be used to determine the load connection type, and subsequently the estimated phase currents. The magnitudes of the spectral envelopes are computed

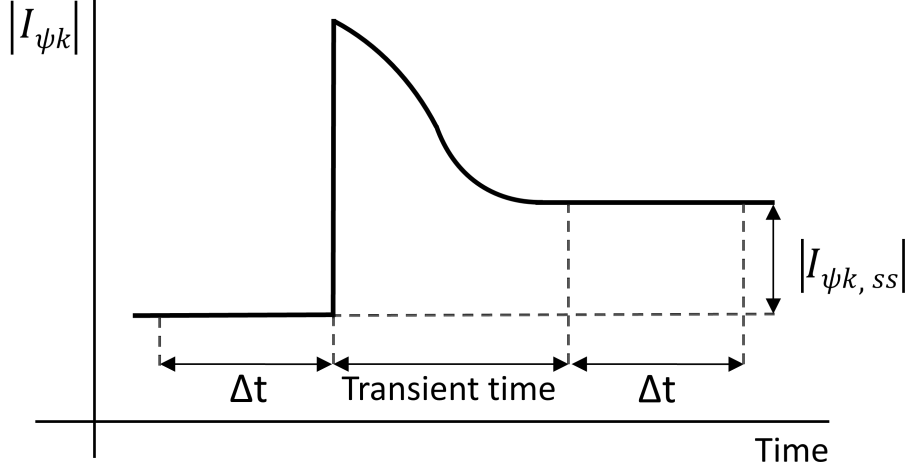


Figure 6-15: Conceptual diagram of steady-state calculation.

as:

$$|I_{\psi k}| = \sqrt{(I_{\psi k}^I)^2 + (I_{\psi k}^Q)^2}. \quad (6.12)$$

At every detected event, the change in steady-state magnitude, $I_{\psi 1,ss}$, is calculated for each phase as the difference in medians (or alternatively, the means) of Δt windows before and after the event, as illustrated in Fig. 6-15. The relative magnitudes of the three line currents can now be computed:

$$\left[\left| \frac{I_{a1,ss}}{I_{b1,ss}} \right|, \left| \frac{I_{b1,ss}}{I_{c1,ss}} \right|, \left| \frac{I_{c1,ss}}{I_{a1,ss}} \right| \right]. \quad (6.13)$$

Alternatively, the ratios can be computed using the apparent power streams (i.e., $S_{\psi} = \sqrt{P_{\psi}^2 + Q_{\psi}^2}$) since P_{ψ} and Q_{ψ} are scaled versions of $I_{\psi 1}^I$ and $I_{\psi 1}^Q$, respectively (using Equation (2.7)). These ratios present a measure of load balance or imbalance. The next two subsections demonstrate how these ratios, in conjunction with power consumption, can help distinguish healthy versus faulty behavior.

6.2.2 Balanced Three-Phase Loads

A balanced three phase load is one in which the load currents have equal magnitudes and phase angles, i.e., $A_{ab,k} = A_{bc,k} = A_{ca,k}$ and $\delta_{ab,k} = \delta_{bc,k} = \delta_{ca,k}$ for every k (where the currents are given in Equation (2.3)). The impedances are $Z_{ab} = Z_{bc} = Z_{ca} =$

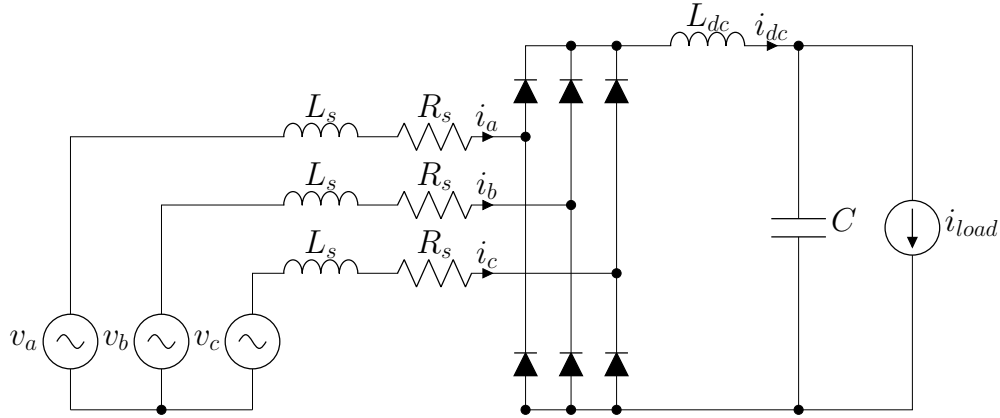


Figure 6-16: Three-phase uncontrolled diode bridge rectifier with inverter-motor combination modeled as current source.

$Z e^{j\phi}$, where ϕ is the angle of the voltage with respect to the corresponding current (i.e., $\phi = \theta - \delta$). For a balanced load, the relative line current magnitudes are [1,1,1]. There is a one-to-one mapping between the power per phase and power per delta phase.

6.2.2.1 Balanced Non-Linear Loads

If the load is nonlinear, i.e., has non-sinusoidal current waveforms, the higher-order current harmonics and their relative magnitudes are also of significance. In a balanced delta system, the triplen (e.g., third, sixth, and ninth order) components are zero-sequence. The zero-sequence line currents are equal to $\frac{1}{3}(i_a + i_b + i_c)$ [166]. Given KCL which requires $i_a + i_b + i_c = 0$, the zero-sequence line currents must be zero. That is, they circulate around the delta but do not flow through the lines. The fifth-order harmonics are typically negative-sequence, while the seventh-order harmonics are typically positive-sequence. Thus, balanced fifth-order and seventh-order line current harmonics are expected.

For example, a variable frequency drive (VFD) is a common nonlinear load used to optimize and control induction motors. A VFD consists of a rectifier, a dc-bus link, and an inverter. The front-end rectifier draws currents rich with harmonic content. A VFD was modeled as shown in Fig. 6-16, using a constant current source i_{load} to

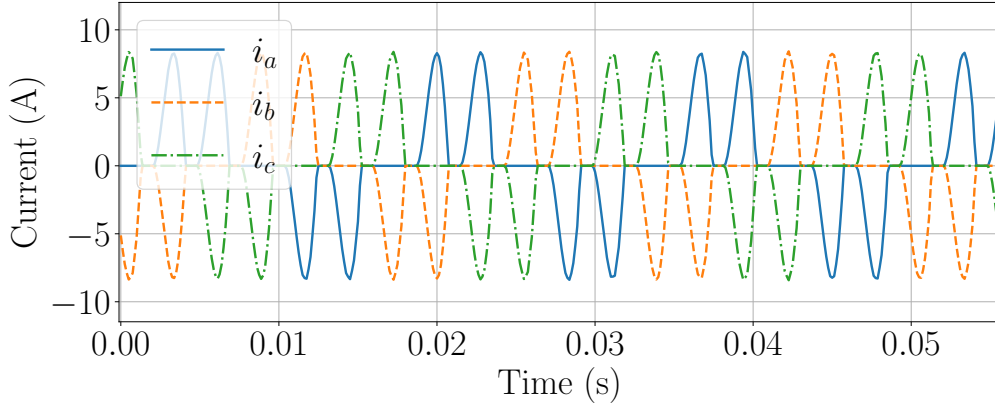


Figure 6-17: Line currents of a simulated 1 kW load controlled by a VFD.

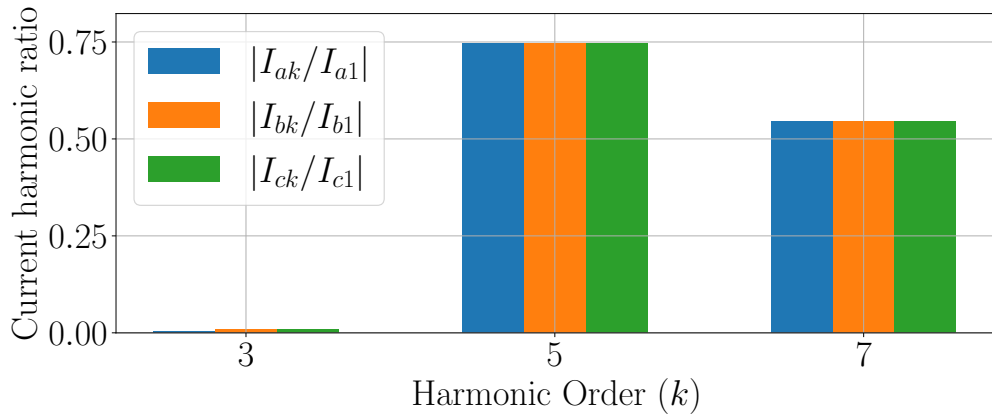
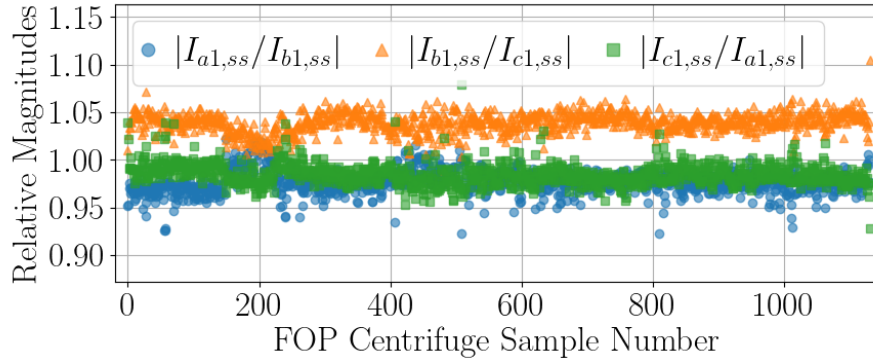
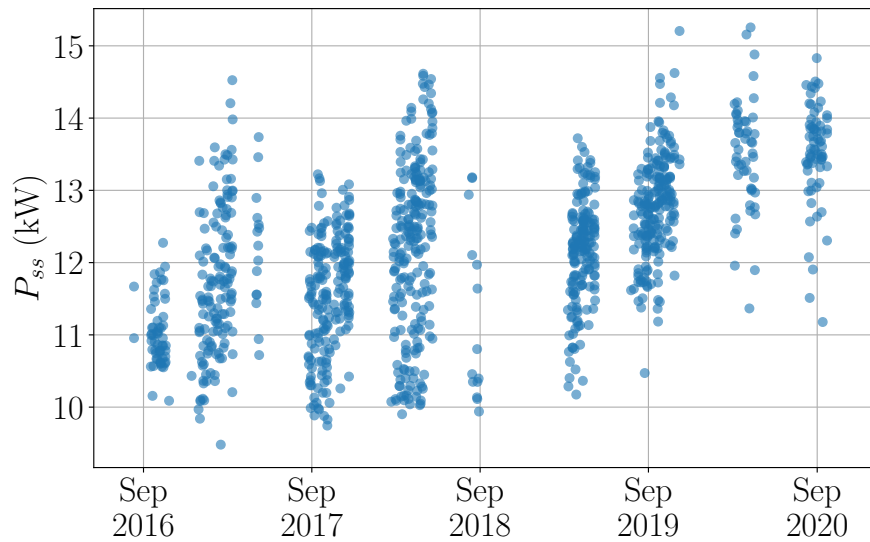


Figure 6-18: Magnitude of third, fifth, and seventh harmonic current envelopes normalized by the fundamental of a motor controlled by a VFD.

model the inverter-motor combination [28]. The line currents of a simulated 1 kW load controlled by a VFD with a balanced source voltage of $V_{pk} = 170$ V are shown Fig. 6-17. The current is characterized by the double pulses of equal magnitude at each positive and negative voltage peak. Fig. 6-18 shows the mean magnitude for the five second simulation of the third, fifth, and seventh harmonic current envelopes, normalized by the magnitude of the fundamental (i.e., $|I_{\psi k}/I_{\psi 1}|$). As expected, the third harmonics are negligible. There are significant fifth and seventh harmonics, which are balanced among the phases. The relative line current magnitudes can also be calculated for each relevant harmonic, k , which in this example is $[1,1,1]$ for $k = 5$ and 7.



(a) Relative line current magnitudes.



(b) Total steady steady real power (P_{ss}).

Figure 6-19: Three-phase fuel oil purifier (FOP) centrifuge over time.

6.2.2.2 Gradual Drift of Balanced Loads

A three-phase load can remain balanced while experiencing slow changes in total power due to changing environmental conditions, changing operating conditions, or a degradation in condition due to a “soft fault.” Soft faults, such as loss of refrigerant, slipping belts, and clogged ventilation systems, can leave the system operating at reduced efficiency and induce wear on the system. Once the load has been identified as being balanced, then the total power can be examined for other subtle changes in load operation [3]. For instance, a NILM on USCGC Spencer revealed the slow change in steady-state power of a fuel oil purifier (FOP) centrifuge. Fig. 6-19a shows

Table 6.5: Phase currents based on relative line current magnitudes.

	Relative line current magnitudes			Phase currents		
	$\frac{I_{a1,ss}}{I_{b1,ss}}$	$\frac{I_{b1,ss}}{I_{c1,ss}}$	$\frac{I_{c1,ss}}{I_{a1,ss}}$	i_{ab}	i_{bc}	i_{ca}
Two delta phases	$\sqrt{3}$	1	$\frac{1}{\sqrt{3}}$	$-i_b$	0	i_c
	$\frac{1}{\sqrt{3}}$	$\sqrt{3}$	1	i_a	$-i_c$	0
	1	$\frac{1}{\sqrt{3}}$	$\sqrt{3}$	0	i_b	$-i_a$
Single delta phase	0	1	∞	0	i_b	0
	1	∞	0	i_a	0	0
	∞	0	1	0	0	i_c

the relative magnitudes of the FOP centrifuge over a four-year period, where each data point represents a single on-event with $\Delta t = 0.5$ seconds for steady-state calculation. The relative magnitudes stay approximately constant and balanced for the entire duration. Fig. 6-19b shows the total steady-state power (P_{ss}) of the FOP centrifuge at each on-event, which varies from approximately 10 kW to 15 kW. This increase in power over time can likely be attributed to load aging and wear.

6.2.3 Imbalances in Three-Phase Systems

Changes in balance that occur during transients need to be correctly attributed to healthy or faulty behavior. For a two-phase or single-phase load, the relative current ratios are predictable, as shown in Table 6.5 for the ideal scenario. Fig. 6-20 shows the relative magnitudes over time for a single-phase ship service diesel generator (SSDG) lube oil (LO) heater on USCGC Spencer. Each data point represents a single on-event with $\Delta t = 0.5$ seconds for steady-state calculation. Here, the relative magnitudes for $\left| \frac{I_{a1,ss}}{I_{b1,ss}} \right|$, $\left| \frac{I_{b1,ss}}{I_{c1,ss}} \right|$, and $\left| \frac{I_{c1,ss}}{I_{a1,ss}} \right|$ are generally less than .001, $1 \pm .05$, and greater than 1000, respectively. Using a tolerance for allowable divergence from expected values, this matches that of a single phase load on phase bc . If an event is identified as being connected on two delta phases or a single delta phase, then the phase currents can be designated as given in Table 6.5. The rms spectral envelopes for i_{ab} , i_{bc} , and i_{ca} can be computed with the same process as described in Section 2.1, but with phase rotations

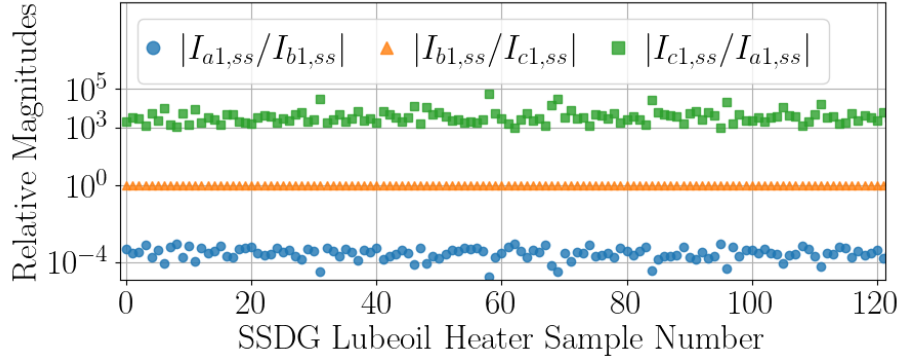


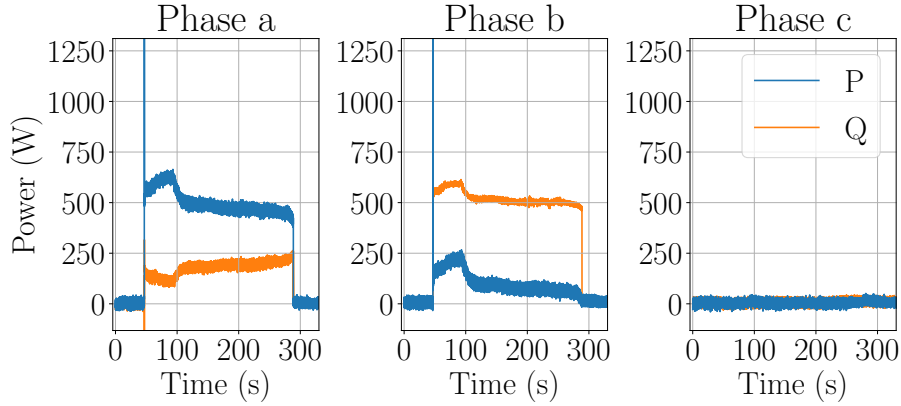
Figure 6-20: Relative magnitude of single-phase ship service diesel generator (SSDG) lube oil (LO) heater load on-events over time.

of 0° , 120° , and 240° . The power per delta phase can be calculated by multiplying by a constant $\sqrt{3}V$ or by the rms spectral envelopes of the line-to-line voltages. If a step change is detected that does not match the expected relative magnitudes of either a three-phase, two-phase, or single-phase load, then the phase currents cannot be estimated. This provides indication of a possible fault condition. This section presents analysis of both normal and faulty imbalances.

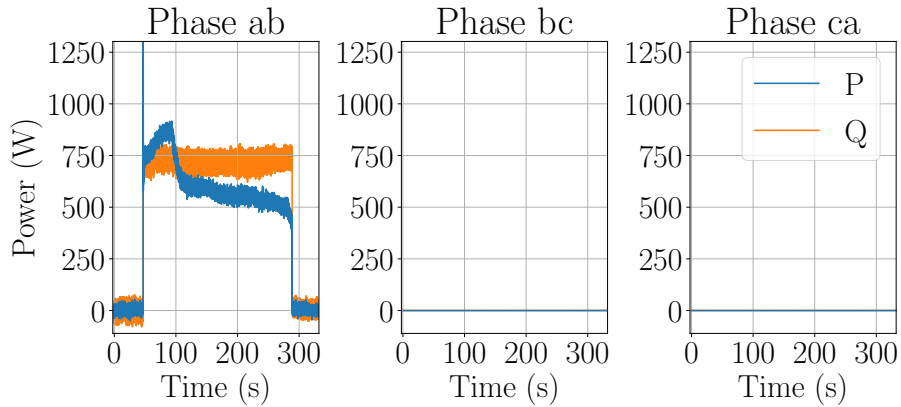
6.2.3.1 Single-Phase Loads

A single phase load inherently imparts an imbalance on the system. For example, the power per phase of a single-phase induction motor from NILM field data is shown in Fig. 6-21a. There is an unbalanced amount of P and Q on phases a and b . However, the imbalance here is normal load behavior. The relative line current magnitudes for the detected on-event are $[1.008, 119.7, 0.008]$. These closely match the expected values for a phase ab single phase load. The phase currents were reconstructed and power per delta phase was computed, as shown in Fig. 6-21b. From the delta phase power it is now possible to track subtle changes over time for this load.

The delta phase power is useful for diagnostics. However, the per phase power may still prove useful for load identification. The question arises if the imbalances have any physical interpretation? The relationship between the power per delta phase and the power per phase can be determined mathematically. The relationships are shown in Table 6.6 for a single-phase load on phase ca with impedance $Z_{ca} = Ze^{j\phi}$.



(a) Measured power per phase of a single-phase inductive load.



(b) Reconstructed power per delta phase.

Figure 6-21: Power of a single-phase inductive load.

Here, V is the rms line-to-neutral voltage. Similar relationships can be derived for single-phase loads on phase ab or bc . The proportion of real and reactive power on phases a and c depends on ϕ . For example, consider a resistive (i.e., $\phi = 0$) load with power on phase ca equal to the load's nominal power P_r , such that $P_{ca} = P_r$ and $Q_{ca} = 0$. The real power on each phase is: $P_a = \frac{1}{2}P_{ca}$, $P_b = 0$, and $P_c = \frac{1}{2}P_{ca}$. The reactive power on each phase is: $Q_a = \frac{1}{2\sqrt{3}}P_{ca}$, $Q_b = 0$, and $Q_c = -\frac{1}{2\sqrt{3}}P_{ca}$. That is, due to the imbalance imparted on the system, there is equal magnitude but opposite sign “reactive” power on two of the phases.

This “reactive” power for a resistive load can be related to the fact that standard definitions of reactive and apparent power are unable to simultaneously characterize the efficiency of equipment, efficiency of power transmission, and oscillatory character of power transfer in unbalanced and polyphase systems [167, 168]. Definitions which

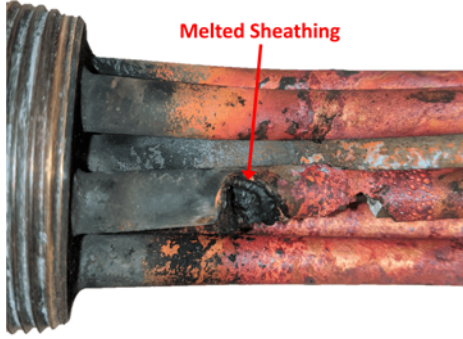
Table 6.6: Line-to-line single-phase load relationships

	Delta phase	Per phase
Currents	$i_{ab} = 0$ $i_{bc} = 0$ $i_{ca} = \sqrt{3} \frac{V}{Z} e^{j(\frac{5\pi}{6} - \phi)}$	$i_a = -i_{ca}$ $i_b = 0$ $i_c = i_{ca}$
Real Power	$P_{ab} = 0$ $P_{bc} = 0$ $P_{ca} = 3 \frac{V^2}{Z} \cos \phi$	$P_a = \sqrt{3} \frac{V^2}{Z} \cos(\phi + \frac{\pi}{6})$ $P_b = 0$ $P_c = \sqrt{3} \frac{V^2}{Z} \cos(\phi - \frac{\pi}{6})$
Reactive Power	$Q_{ab} = 0$ $Q_{bc} = 0$ $Q_{ca} = 3 \frac{V^2}{Z} \sin \phi$	$Q_a = \sqrt{3} \frac{V^2}{Z} \sin(\phi + \frac{\pi}{6})$ $Q_b = 0$ $Q_c = \sqrt{3} \frac{V^2}{Z} \sin(\phi - \frac{\pi}{6})$

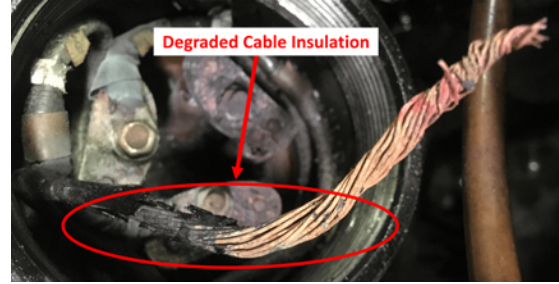
use different assumptions about the system can provide possible additional features. For instance, the arithmetic apparent power is the arithmetic sum of the apparent power of the individual phases, $S_A = \sum_{\psi} \sqrt{P_{\psi}^2 + Q_{\psi}^2}$ [168]. The vector apparent power is the magnitude of the total vector power, $S_V = \sqrt{(\sum_{\psi} P_{\psi})^2 + (\sum_{\psi} Q_{\psi})^2}$ [168]. The arithmetic and vector power factors are $PF_A = P_{total}/S_A$ and $PF_V = P_{total}/S_V$. For a resistive single-phase load, $PF_A = 0.866$ and $PF_V = 1.0$. The arithmetic power factor is less than one, which is a result of the per phase “reactive” power. The power per phase and the arithmetic power factor both take into account imbalances imparted on the system, and can provide a unique load signature for load identification. However, the delta phase power and vector power factor are more intuitive metrics for reporting load behavior. The vector power factor represents the individual load’s power factor, without consideration for any imbalances imparted on the system. That is, it physically makes sense for a resistive load to have a power factor close to one.

6.2.3.2 Three-Phase Heater Faults

A delta-connected resistive or heating load consists of three heating elements connected line-to-line. This type of load can experience numerous failures due to the degradation of materials in industrial environments. Fig. 6-22 shows two examples of degradation pertaining to a three-phase heater from two different USCG shipboard microgrids. Specifically, Fig. 6-22a shows deteriorated heating elements and Fig. 6-



(a) Significant corrosion and melted sheathing of heating elements.



(b) Loose wire connection and degraded cable insulation.

Figure 6-22: Two degraded three-phase heaters.

Table 6.7: Line-to-line open-circuit fault relationships

	Delta phase	Per phase
Currents	$i_{ab} = 0$ $i_{bc} = \sqrt{3} \frac{V}{Z} e^{j(-\frac{\pi}{2}-\phi)}$ $i_{ca} = \sqrt{3} \frac{V}{Z} e^{j(\frac{5\pi}{6}-\phi)}$	$i_a = -i_{ca}$ $i_b = i_{bc}$ $i_c = 3 \frac{V}{Z} e^{j(\frac{2\pi}{3}-\phi)}$
Real Power	$P_{ab} = 0$ $P_{bc} = 3 \frac{V^2}{Z} \cos \phi$ $P_{ca} = 3 \frac{V^2}{Z} \cos \phi$	$P_a = \sqrt{3} \frac{V^2}{Z} \cos(\phi + \frac{\pi}{6})$ $P_b = \sqrt{3} \frac{V^2}{Z} \cos(\phi - \frac{\pi}{6})$ $P_c = 3 \frac{V^2}{Z} \cos \phi$
Reactive Power	$Q_{ab} = 0$ $Q_{bc} = 3 \frac{V^2}{Z} \sin \phi$ $Q_{ca} = 3 \frac{V^2}{Z} \sin \phi$	$Q_a = \sqrt{3} \frac{V^2}{Z} \sin(\phi + \frac{\pi}{6})$ $Q_b = \sqrt{3} \frac{V^2}{Z} \sin(\phi - \frac{\pi}{6})$ $Q_c = 3 \frac{V^2}{Z} \sin \phi$

22b shows a loose connection and degraded cable insulation. These faults result in various electrical failure signatures.

The first fault condition considered is a line-to-line open-circuit. This can result from a failed heating element, for example due to damaged sheathing and corrosion. A line-to-line open-circuit manifests as a reduction in magnitude and phase shift in two of the line currents. That is, the three-phase load acts as a two-phase load. Given a phase ab open-circuit such that $Z_{ab} = \infty$, $Z_{bc} = Ze^{j\phi}$, and $Z_{ca} = Ze^{j\phi}$, Table 6.7 presents the relationship between the power per delta phase and the power per phase. Similar relationships can be derived for a bc or ca open-circuit. If there is a second line-to-line open-circuit, it would act as a single-phase load. Fig. 6-23 shows the power for a simulated 3 kW resistive load with a phase ab open-circuit. Fig. 6-23a shows the

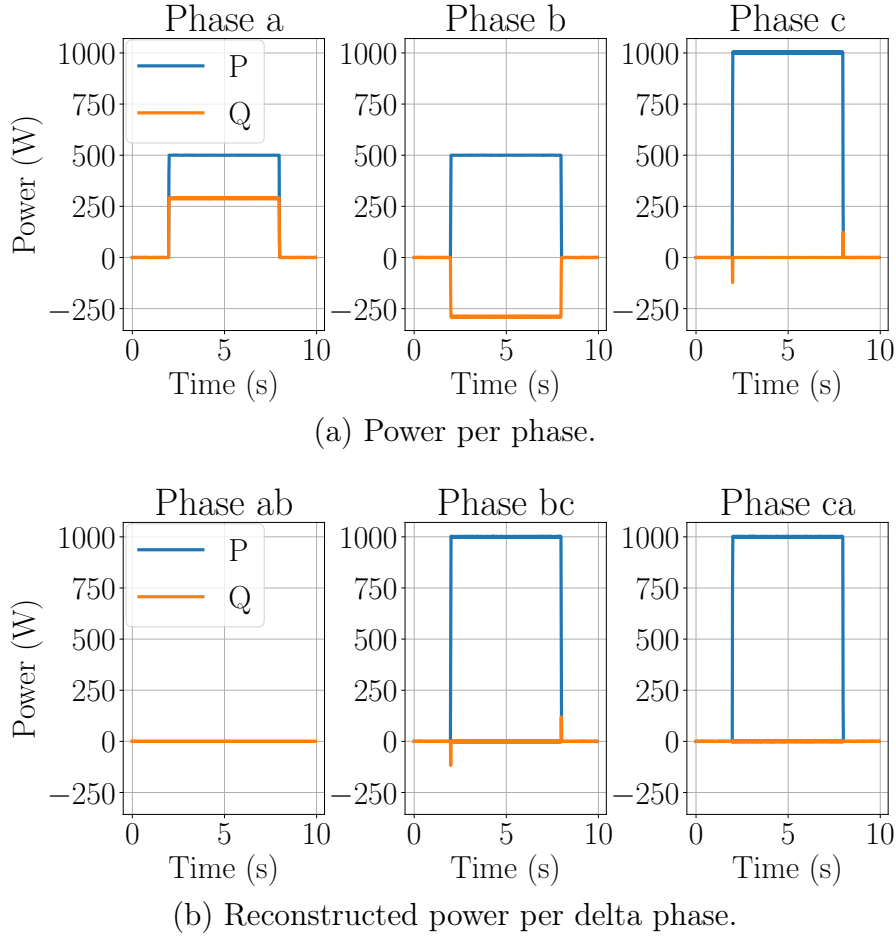


Figure 6-23: Load connected across two delta phases.

power as calculated from the line currents. The phase currents are reconstructed and power per delta phase is computed, as shown in Fig. 6-23b. The power per delta phase correctly shows 1 kW on both phases *bc* and *ca*. The next fault condition considered is a missing voltage connection to the load, essentially turning a three-phase load into a single-phase load. This can result from degraded or loose wiring in the terminal box. Fig. 6-24 shows conceptual diagrams for examples of these two fault scenarios (line-to-line open-circuit and missing voltage connection), plus the combination of both faults simultaneously. Here R is the resistance of a single heating element. Table 6.8 shows the resulting delta impedances, delta phase power, per phase power, and total power for these three fault scenarios for a heating load with nominal rating $P_r = 9\frac{V^2}{R}$. With a missing voltage connection, the heater draws one-half of its nominal power. With an open-circuited heating element, the heater draws two-thirds of its nominal

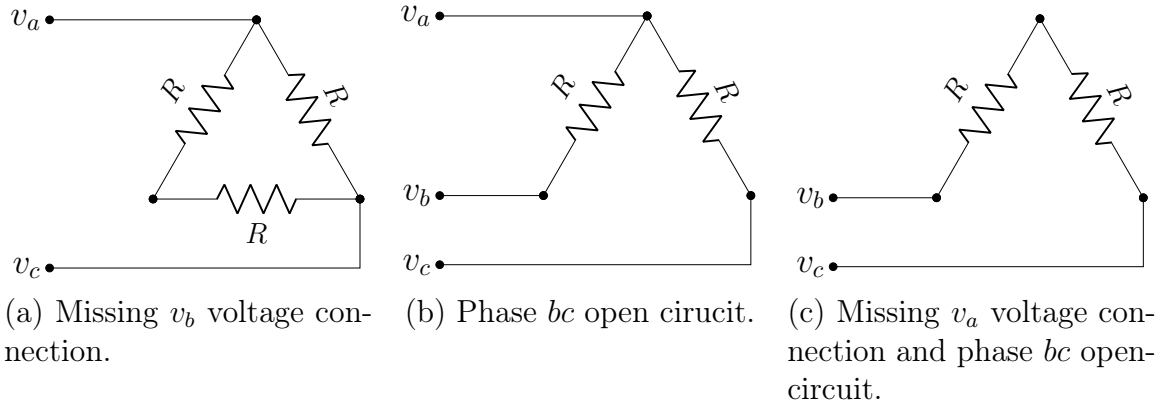


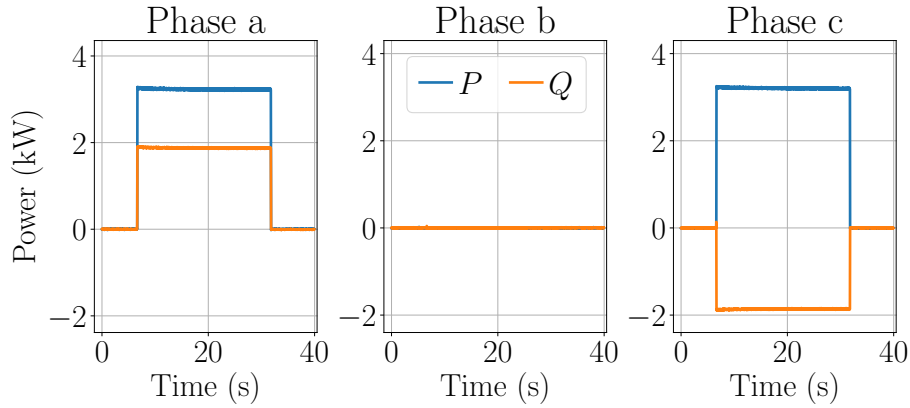
Figure 6-24: Delta-connected heater under various fault conditions.

Table 6.8: Delta-connected heating load fault conditions.

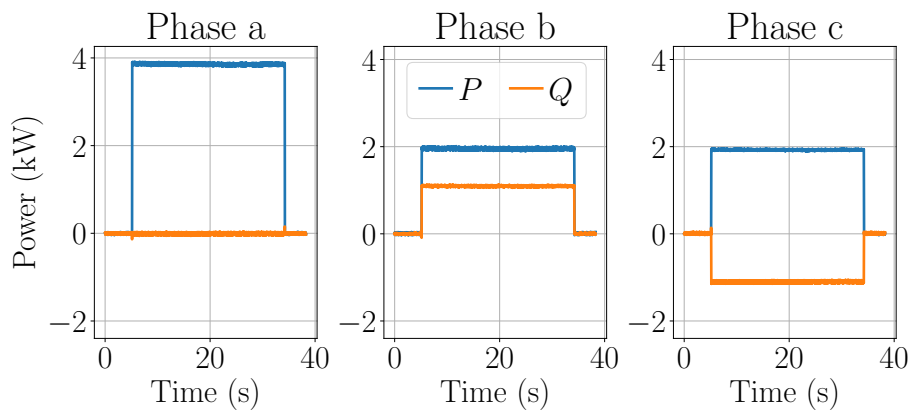
Scenario	Delta impedances	Delta phase power	Per phase power		Total power
No v_b connection	$Z_{ca} = \frac{2}{3}R$	$P_{ca} = \frac{1}{2}P_r$	$P_a = \frac{1}{4}P_r$ $P_c = \frac{1}{4}P_r$	$Q_a = \frac{1}{4\sqrt{3}}P_r$ $Q_c = -\frac{1}{4\sqrt{3}}P_r$	$\frac{1}{2}P_r$
bc open-circuit	$Z_{ab} = R$ $Z_{bc} = \infty$ $Z_{ca} = R$	$P_{ab} = \frac{1}{3}P_r$ $P_{bc} = 0$ $P_{ca} = \frac{1}{3}P_r$	$P_a = \frac{1}{3}P_r$ $P_b = \frac{1}{6}P_r$ $P_c = \frac{1}{6}P_r$	$Q_a = 0$ $Q_b = \frac{1}{6\sqrt{3}}P_r$ $Q_c = -\frac{1}{6\sqrt{3}}P_r$	$\frac{2}{3}P_r$
bc open-circuit and no v_a connection	$Z_{bc} = 2R$	$P_{bc} = \frac{1}{6}P_r$	$P_b = \frac{1}{12}P_r$ $P_c = \frac{1}{12}P_r$	$Q_b = -\frac{1}{12\sqrt{3}}P_r$ $Q_c = \frac{1}{12\sqrt{3}}P_r$	$\frac{1}{6}P_r$

power. These two faults combined, depending on which voltage is missing and which heating element is open-circuited, results in the heater drawing either one-sixth (the scenario shown in Fig. 6-24c) or one-third of its nominal power. The heater will also draw one-third of its nominal power if it has two open-circuited heating elements.

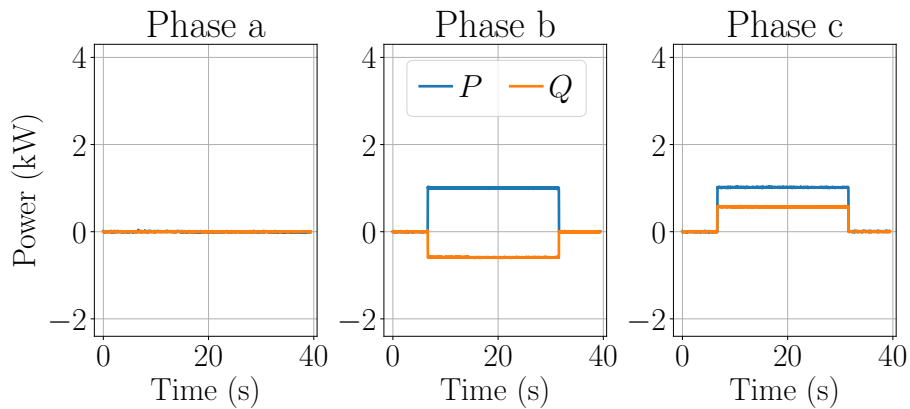
These fault scenarios have been observed with NILMs on various USCG shipboard microgrids. Fig. 6-25 shows the per-phase power streams of the nominal 12 kW main propulsion diesel engine (MPDE) jacket water (JW) heater on USCGC Thunder Bay in these various fault states. The baseline power has been subtracted out for easier visual display. The port-side MPDE JW heater in Fig. 6-25a is drawing a total of 6 kW, due to a missing v_b voltage connection. It appears as a single-phase load in this faulty state, with relative line current magnitudes, [653.54, 0.0015, 0.9913]. This



(a) Port-side heater with a missing voltage connection.



(b) Stbd-side heater with a line-to-line open-circuit.



(c) Stbd-side heater with a line-to-line open-circuit and a missing voltage connection.

Figure 6-25: Real and reactive power for USCGC Thunder Bay MPDE JW heater in various fault scenarios.

matches that of a phase ca single-phase load. The starboard-side MPDE JW heater in Fig. 6-25b is drawing 8 kW. This is due to an open-circuited phase bc heating

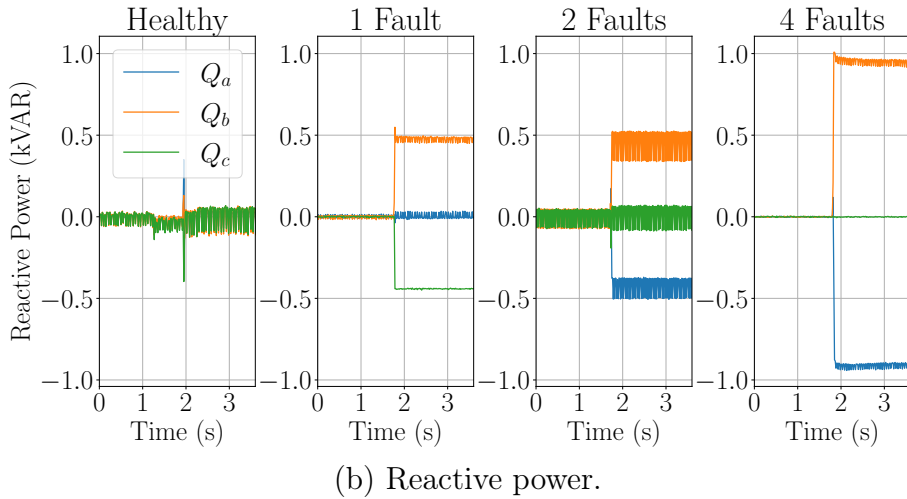
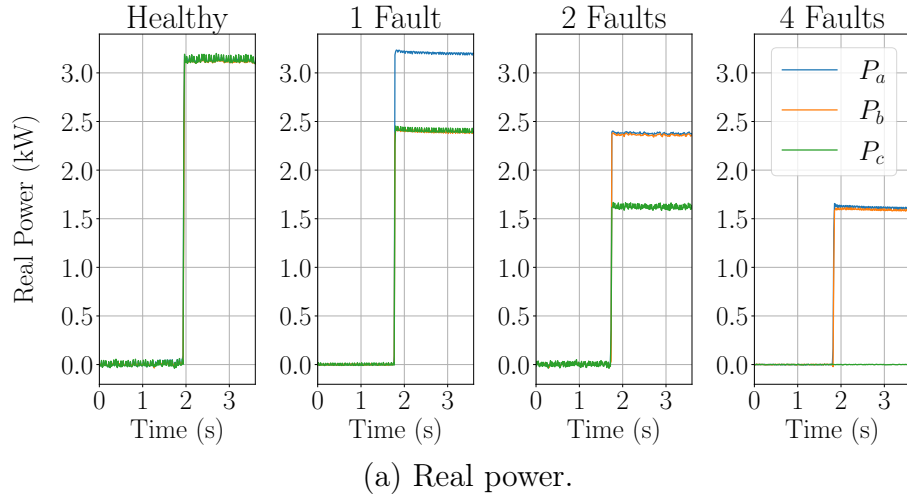


Figure 6-26: Per-phase real power and reactive power of the USCGC Spencer port-side MPDE JW heater turn-on events, starting healthy and after the heater underwent various stages of degradation.

element. The relative line current magnitudes for this load, $[1.7315, 0.9995, 0.5778]$, matches that of a two-phase load on phases ab and ca . This heater previously had no v_a voltage connection due to corroded wiring [see Fig. 6-22b]. Combined with the line-to-line open-circuit, the heater was only drawing 2 kW, as shown in Fig. 6-25c. The relative current magnitudes in this state, $[0.0004, 0.9977, 239.92]$, matches that of a phase bc single-phase load. The per-phase power for these three scenarios matches the expected values from Table 6.8.

On USCGCs Spencer and Escanaba, degradation of the 4.5 kW MPDE JW heaters

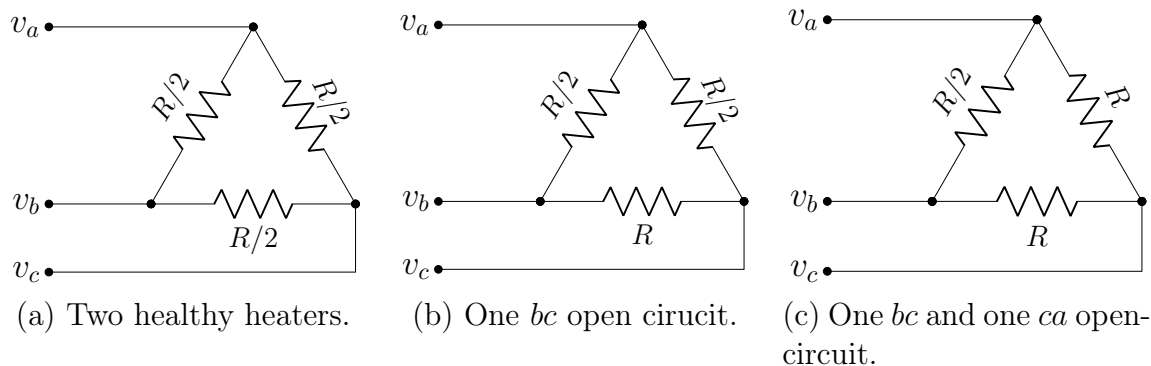


Figure 6-27: Two delta-connected heaters operating in tandem.

Table 6.9: Two delta-connected heaters operating in tandem.

Scenario	Delta	Delta	Per phase		Total power
	impedances	phase power	power		
Two healthy heaters	$Z_{ab} = \frac{1}{2}R$	$P_{ab} = \frac{2}{3}P_r$	$P_a = \frac{2}{3}P_r$	$Q_a = 0$	$2P_r$
	$Z_{bc} = \frac{1}{2}R$	$P_{bc} = \frac{2}{3}P_r$	$P_b = \frac{2}{3}P_r$	$Q_b = 0$	
	$Z_{ca} = \frac{1}{2}R$	$P_{ca} = \frac{2}{3}P_r$	$P_c = \frac{2}{3}P_r$	$Q_c = 0$	
One bc open-circuit	$Z_{ab} = \frac{1}{2}R$	$P_{ab} = \frac{2}{3}P_r$	$P_a = \frac{2}{3}P_r$	$Q_a = 0$	$\frac{5}{3}P_r$
	$Z_{bc} = R$	$P_{bc} = \frac{1}{3}P_r$	$P_b = \frac{1}{2}P_r$	$Q_b = \frac{1}{6\sqrt{3}}P_r$	
	$Z_{ca} = \frac{1}{2}R$	$P_{ca} = \frac{2}{3}P_r$	$P_c = \frac{1}{2}P_r$	$Q_c = -\frac{1}{6\sqrt{3}}P_r$	
One bc and one ca open-circuit	$Z_{ab} = \frac{1}{2}R$	$P_{ab} = \frac{2}{3}P_r$	$P_a = \frac{1}{2}P_r$	$Q_a = -\frac{1}{6\sqrt{3}}P_r$	$\frac{4}{3}P_r$
	$Z_{bc} = R$	$P_{bc} = \frac{1}{3}P_r$	$P_b = \frac{1}{2}P_r$	$Q_b = \frac{1}{6\sqrt{3}}P_r$	
	$Z_{ca} = R$	$P_{ca} = \frac{1}{3}P_r$	$P_c = \frac{1}{3}P_r$	$Q_c = 0$	

was observed, as described in Section 1.2. Each NILM monitors two heaters in tandem. Thus, the heater pair draws 9 kW under normal operation. A NILM can detect an open-circuit failure of any of the six heating elements. Open-circuiting of the heating elements has been observed on both ships, with the observed time between consecutive open-circuit failures ranging from hours to months. For example, Fig. 6-26 shows the real and reactive power on-transient for four different measured scenarios of the port-side MPDE JW heater system on USCGC Spencer. The leftmost plot is of two healthy heaters. The step-change in power when the load turns on is approxi-

Table 6.10: Relative line current magnitudes of MPDE JW heater system.

	Expected			Observed		
	$\left \frac{I_{a1,ss}}{I_{b1,ss}} \right $	$\left \frac{I_{b1,ss}}{I_{c1,ss}} \right $	$\left \frac{I_{c1,ss}}{I_{a1,ss}} \right $	$\left \frac{I_{a1,ss}}{I_{b1,ss}} \right $	$\left \frac{I_{b1,ss}}{I_{c1,ss}} \right $	$\left \frac{I_{c1,ss}}{I_{a1,ss}} \right $
Two healthy heaters	1	1	1	1.002	0.9955	1.002
One <i>bc</i> open-circuit	1.309	1	0.764	1.310	1.003	0.761
One <i>bc</i> and one <i>ca</i> open-circuit	1	1.528	0.655	1.002	1.483	0.672
Two <i>bc</i> and two <i>ca</i> open-circuits	1	∞	0	0.9980	1678.23	0.0006

mately 3 kW of real power and zero reactive power on each phase. The second plot has a single degraded *bc* heating element. The third plot has two degraded heating elements: one *bc* element and one *ca* element. The rightmost plot has four degraded heating elements: both *bc* elements and both *ca* elements. In this final state, only two of the six heating elements are operational and the heating system is only drawing one-third of the rated power.

Fig. 6-27 shows conceptual diagrams corresponding to the first three scenarios, where R is the resistance of a single heating element. Table 6.9 shows the resulting delta impedances, delta phase power, per phase power, and total power for these three scenarios, where P_r is the nominal power of a single three-phase heater (e.g., 4.5 kW for the USCGC Spencer MPDE JW heater). As shown, two healthy heaters have a total power of $2P_r$. Each failed heating element reduces the total power by $\frac{1}{3}P_r$. The expected and observed relative current magnitudes for these four scenarios are shown in Table 6.10, showing excellent agreement. The first scenario has the expected relative magnitudes for a balanced three-phase load. In the second and third scenario, the heating system is acting as an imbalanced three-phase load, so it does not match the relative current magnitudes of a balanced three-phase load. Instead, the expected relative magnitudes are determined using the apparent powers calculated from the per-phase real and reactive powers in Table 6.9. Finally, the fourth scenario matches that of a load only across phases *a* and *b*.

These events are difficult for watchstanders to detect, as the heater uses its operational heating element(s) to still warm the jacket water. For a single heater, only after three heating elements open-circuit or two loose voltage connections will the heating system have a complete system failure. For two heaters in tandem, only after six failed heating elements will the system have a complete failure. Automatic controllers alter run times to ensure jacket water temperatures are regulated. Only in the extremely degraded state of Fig. 6-25c was the heater not able to produce the heat output necessary, even when the heater was operating with an approximately 100% duty cycle. These failures indicate that the heater may be at risk for other damaging failures with potentially dangerous operating conditions. Thus, the degraded heater should be detected as early as possible. As was shown, these failure signatures are predictable. Thus, a NILM pattern classifier can be trained to recognize these failure events using simulated degraded load transients. That is, training provided by a simulation may fill gaps in systems where, for example, faults cannot be purposely introduced.

6.2.3.3 Arcing Fault

Degraded materials can lead to damaged sheathing or exposed wiring and can cause arcing, effectively a dangerous electrical short between the phases. Two arcing events were observed on USCGC Spencer. These arcing events are attributed to the degradation of the copper sheathing of a 4.5 kW MPDE JW heater, since the currents return to normal levels with the turn-off of the heater. Both events clipped at the 300 A limit of the NILM current sensors. The currents and voltages of the first event are shown in Fig. 6-28. The arcing event lasted for about six line cycles and occurred while the heater was drawing the nominal 4.5 kW. All three phases experience a large current spike. Using curve fitting on the clipped current data to estimate the true current magnitude, it was estimated that the current on phase *b* was greater than that of phases *a* and *c*. Arcing is known to cause distortions in the voltage waveform [169,170], as is evident in Fig. 6-28. The second arcing event occurred approximately nine hours later. In the interval between these arcing events, two heating

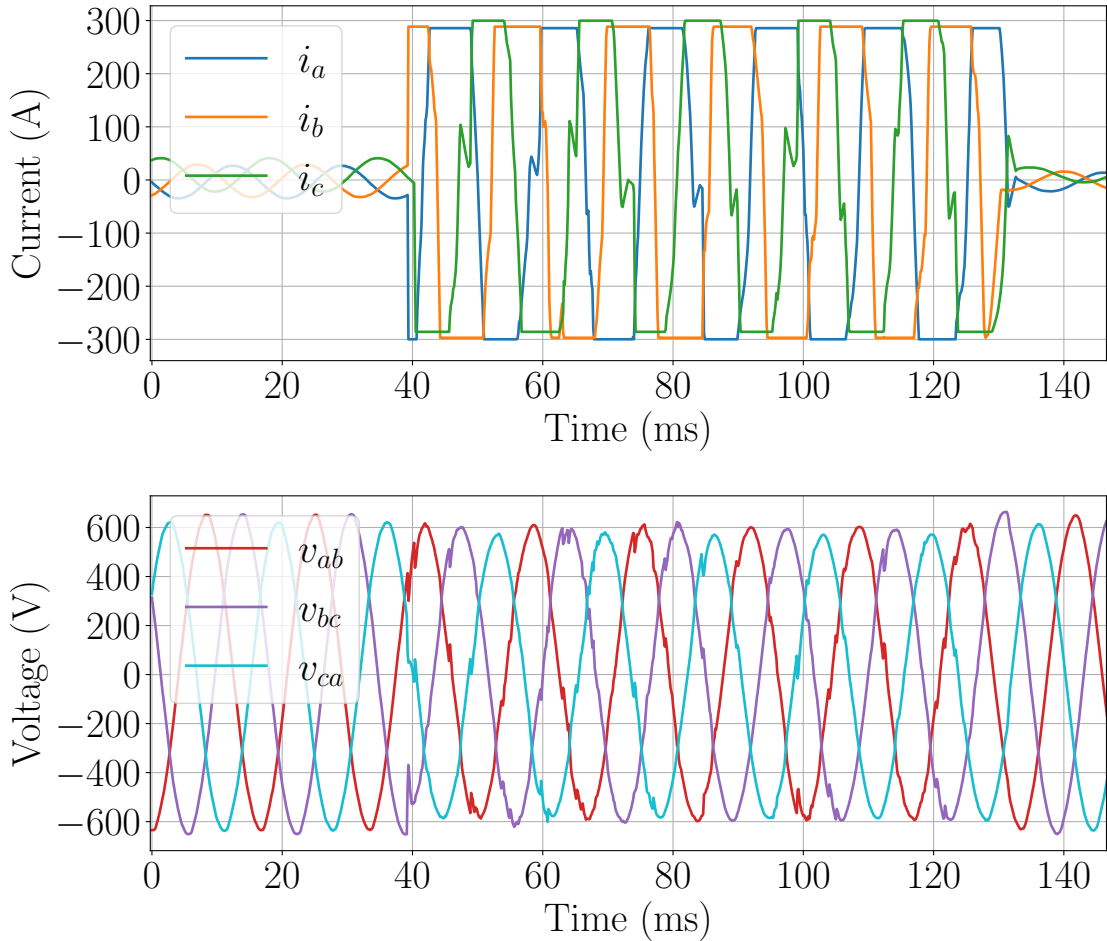


Figure 6-28: Currents and voltages during the first arcing event, in which all three line currents spike.

elements open-circuited. Thus, during the second arcing event only the ca element is operating and only phases c and a experience spikes in current. This event lasted for about 22 line cycles. Fig. 6-29 shows the current stream for the entire arc duration. Fig. 6-30 shows the current and voltage streams for the beginning of the arc event, showing the evident distortions in the voltage waveform.

The heater is supplied from a 15 A breaker with an instantaneous trip rating of 180 to 750 A rms. The trip rating depends on both the current magnitude and event duration. From curve fitting on the clipped current data, it was estimated that both events fell between the breaker's minimum and maximum clearing times. An exact magnitude cannot be determined because of the clipping and distortions around the zero-crossing, but the estimate indicates that the breaker likely did not trip during

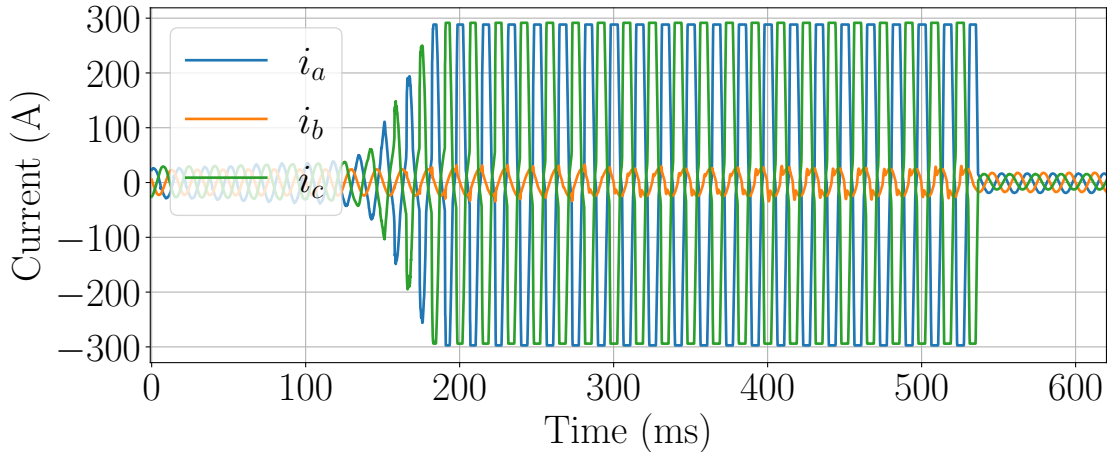


Figure 6-29: Currents during the second arcing event, in which phases a and c show current spikes.

these events. This was corroborated by speaking with watchstanders onboard the vessel and confirmed that they had no indication that these arcing events occurred or of any breakers having tripped. The standard protection and relaying on this power system failed to catch the dangerous arcing that was easily observed by the NILM. A NILM can alert watchstanders to any unusually high power events as indication of arcing. Removal of the heater after the arcing event in Fig. 6-29 revealed significant damage and melted sheathing [see Fig. 6-22a].

6.2.3.4 High Impedance Ground Fault

The sheathing of the presented JW heaters are grounded. Thus, damaged sheathing and a damaged heating element may lead to a ground fault. The detection and location of ground faults are necessary for power utilization safety. The monitored shipboard systems are nominally ungrounded systems. However, an ungrounded system is in reality capacitively grounded through the parasitic, or natural capacitances of three-phase conductors and other distribution equipment to ground (e.g., the ship’s hull). This creates an inherent capacitive current [171]. The capacitive impedances of ungrounded systems limits the ground fault current and makes them “high-impedance faults.” As a result, the normal overcurrent protection devices may

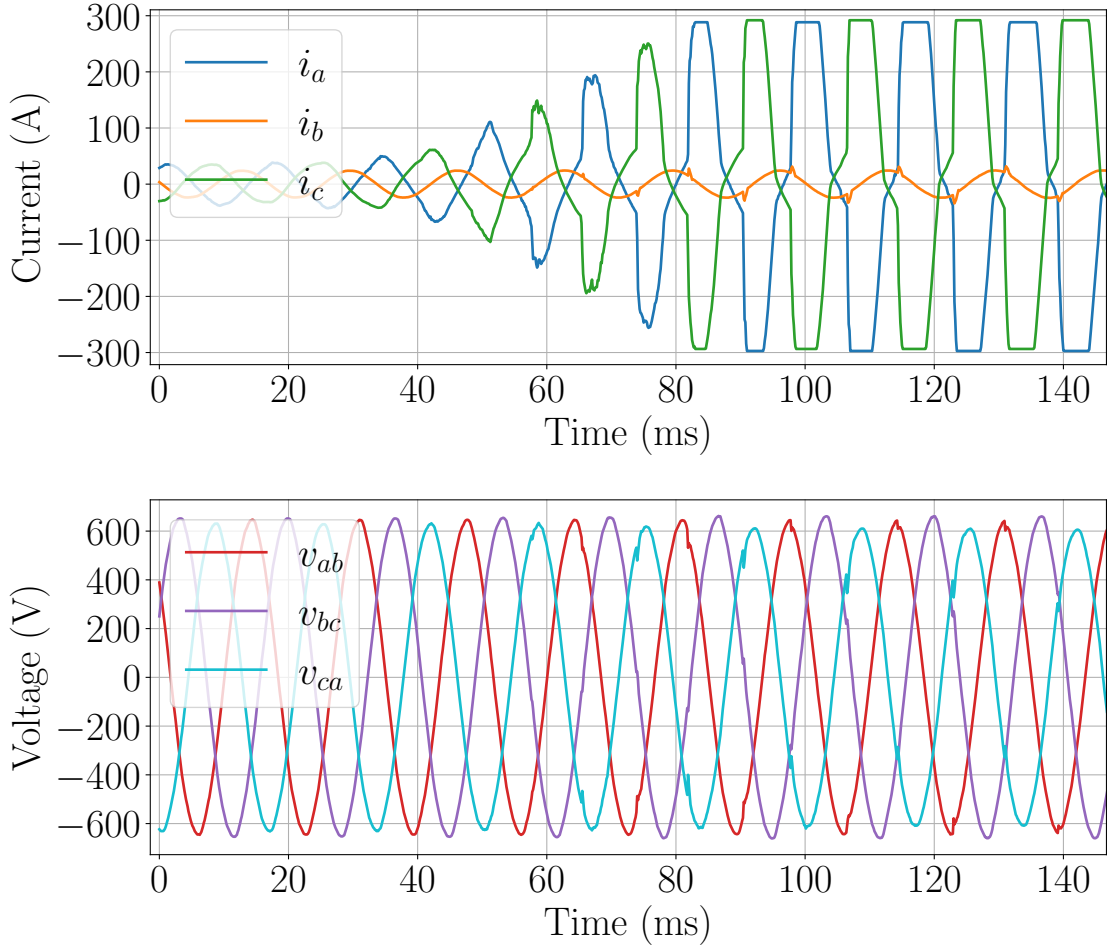


Figure 6-30: Zoomed-in view of currents and voltages during the second arcing event, in which phases a and c show current spikes.

not detect and clear these faults which may be similar in magnitude to those of normal loads [172, 173]. Instead, detection of a ground fault is often performed with a zero sequence voltage relay. Locating ground faults requires the manual sequential opening of branch circuits until the fault disappears. As will be described, a ground fault manifests as a unique power signature. Thus, a NILM can detect ground faults and supplement existing protection schemes.

Fig. 6-31 shows a diagram for an ungrounded power system with parasitic capacitances, C_0 , between each phase and ground, and a phase a fault to ground. For an unfaulted scenario and assuming balanced voltages as given in Eq. (2.2), the

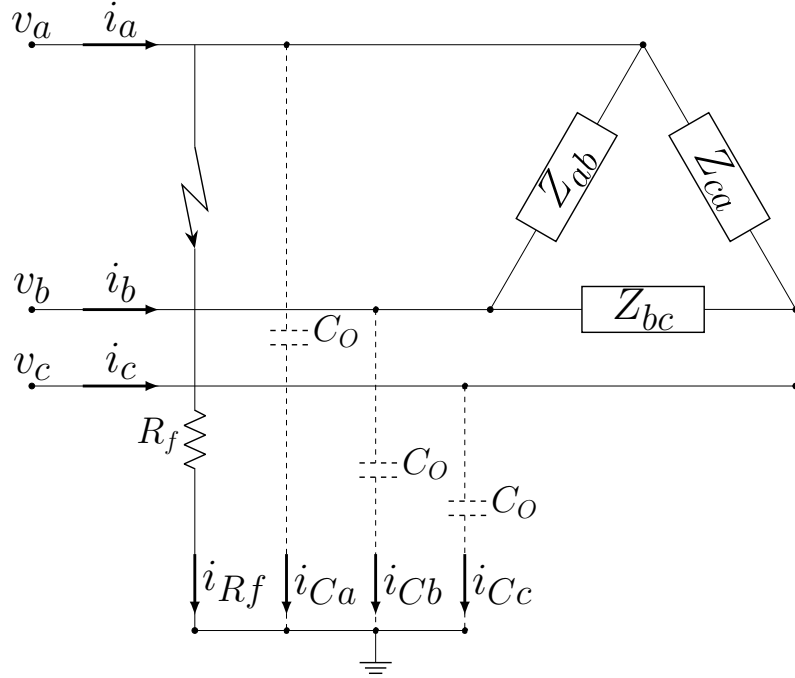


Figure 6-31: Ungrounded power system with a phase-*a* fault to ground. The system is parasitically grounded through the capacitances between the three-phase conductors and ground.

capacitive currents are:

$$\begin{bmatrix} i_{Ca} \\ i_{Cb} \\ i_{Cc} \end{bmatrix} = \begin{bmatrix} \omega C_0 V e^{j\frac{\pi}{2}} \\ \omega C_0 V e^{-j\frac{\pi}{6}} \\ \omega C_0 V e^{-j\frac{5\pi}{6}} \end{bmatrix}. \quad (6.14)$$

The parasitic capacitances are the only current return path during a ground fault. Given a phase *a* solid ground fault, the line-to-ground voltages and capacitive currents are:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{3}V e^{-j\frac{5\pi}{6}} \\ \sqrt{3}V e^{j\frac{5\pi}{6}} \end{bmatrix}, \quad \begin{bmatrix} i_{Ca} \\ i_{Cb} \\ i_{Cc} \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{3}\omega C_0 V e^{-j\frac{\pi}{3}} \\ \sqrt{3}\omega C_0 V e^{-j\frac{2\pi}{3}} \end{bmatrix}. \quad (6.15)$$

In this case, the faulted phase and ground potential are equated and the voltage of the two unfaulted phases to ground are raised from V to $\sqrt{3}V$ [171]. The line-to-line voltages do not change with a ground fault. That is, a single line-to-ground fault does not cause any service disruptions. Instead, a shifting of the neutral occurs. Using KCL, the fault current is three times the original capacitive current from each

phase to ground. Ground faults often occur with some resistance, R_f , associated with the fault, which affects the line-to-ground voltage shift and fault current magnitude. Ignoring line impedances, the line-to-ground voltage for a phase a fault to ground is:

$$v_a = V \frac{j3\omega C_0 R_f}{1 + j3\omega C_0 R_f}. \quad (6.16)$$

The total phase a fault current is:

$$i_{\text{fault}} = j3\omega C_0 V \frac{1 + j\omega C_0 R_f}{1 + j3\omega C_0 R_f}. \quad (6.17)$$

The proportion of the fault current that gets measured in the line current of the faulted phase is the current through the fault resistance, which can be solved for by current division:

$$i_{R_f} = \frac{j3\omega C_0 V}{1 + j3\omega C_0 R_f}. \quad (6.18)$$

If R_f is zero, the current as measured by the NILM is three times the original capacitive current from each phase to ground. As R_f increases, the change in line-to-ground voltage decreases and the fault current decreases. In terms of power (using the definition in Eq. (2.7) which assumes a constant voltage, V), if R_f is zero, a negative reactive power will be added to the aggregate power stream. If R_f is nonzero, the ground fault also adds a positive real power to the faulted phase. For example, with a parasitic capacitance $C_0 = 6.25 \mu\text{F}$, the real and reactive power as would be measured by a NILM on the faulted phase for fault resistances ranging from 0Ω to 200Ω is shown in Fig. 6-32.

Two line-to-ground fault events were observed on USCGC Spencer. These events were likely due to the damaged sheathing and degraded heating elements of the MPDE JW heaters. Fig. 6-33a and Fig. 6-33b show the aggregate power during the first ground fault. There is a ground fault on phase c at $t = 0.6$ min, appearing as a step change in P_c and Q_c of approximately 109 W and -437 VAR, respectively. There is no significant change in power on the other two phases. The relative line current magnitudes for this observed fault are $[2.462, 0.005, 74.060]$, which do not match any

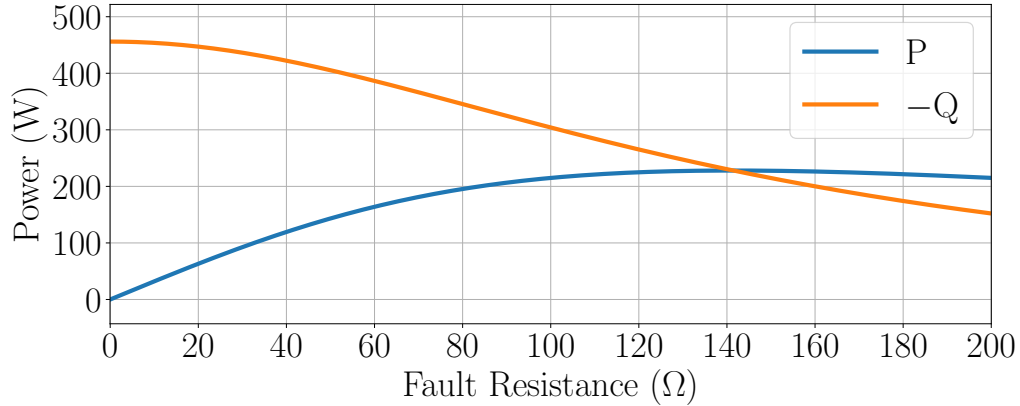
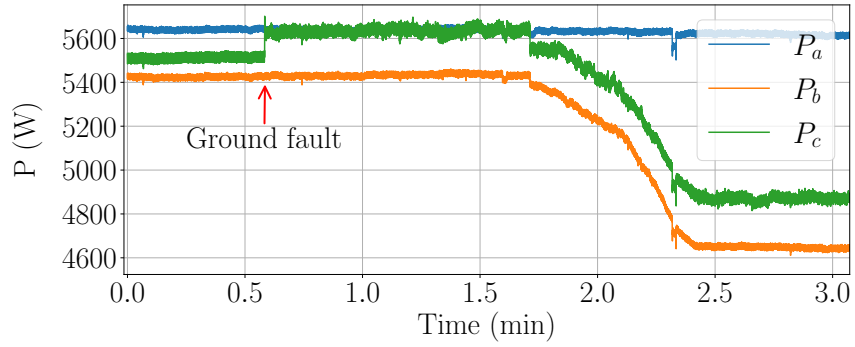


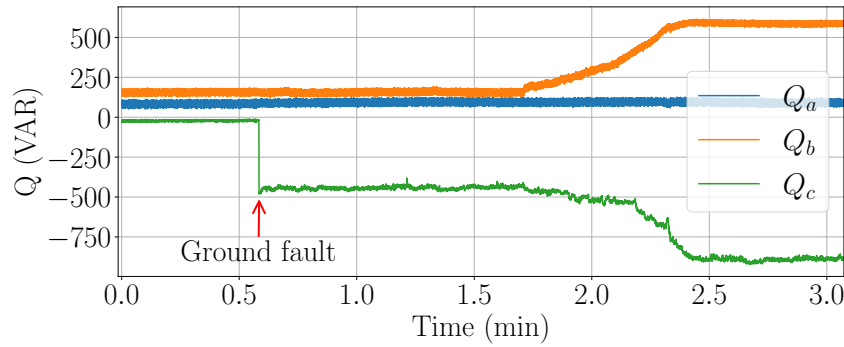
Figure 6-32: Ground fault power on the faulted phase for different fault resistances.

of the expected values for a relatively balanced three-phase load, two-phase load, or single-phase load. A similar ground fault was observed several months later on phase b , as shown in Fig. 6-34a and Fig. 6-34b. At $t = 0.35$ min, there is a step change in P_b and Q_b of approximately 88 W and -448 VAR, respectively. Similar to the first ground fault, there is no significant change in power on the two healthy phases. The relative current magnitudes of this ground fault are $[0.003, 34.334, 9.343]$. A ground fault also leads to third-harmonic line current in the faulted phase [173]. As mentioned, third-harmonic currents are typically zero-sequence components. With a ground fault, a low impedance path is provided for the zero-sequence components. Fig. 6-33c and Fig. 6-34c show the rms third-harmonic current envelopes for the two ground faults, with a 12-point rolling mean applied.

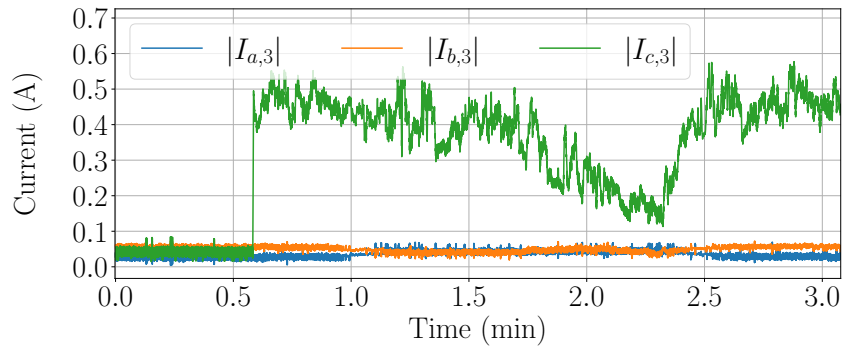
The nature of the high-impedance ground fault has meaningful implications for using a NILM to detect these faults. Even though the step change in power is similar in magnitude to those of normal loads, the way it manifests as increased line currents on only a single phase, both fundamental and third-harmonic, is likely unique. All the load connection types described will have line currents on at least two of the lines. The relative line current magnitudes can also provide indication of a ground fault. The relative line current magnitudes for a ground fault will contain a 0 and ∞ (realistically just a large value), similar to a single-phase load. The third value depends on the step changes of the line currents of the two healthy phases, which are both close to zero, so it is unlikely for the ratio to be exactly 1. If the ground fault



(a) Aggregate real power.



(b) Aggregate reactive power.

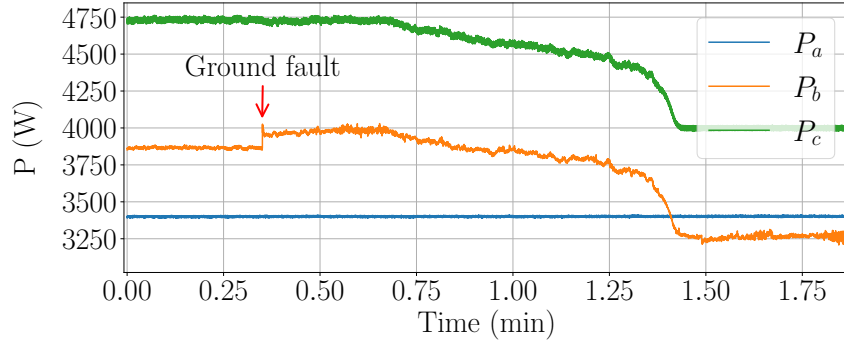


(c) Third harmonic current.

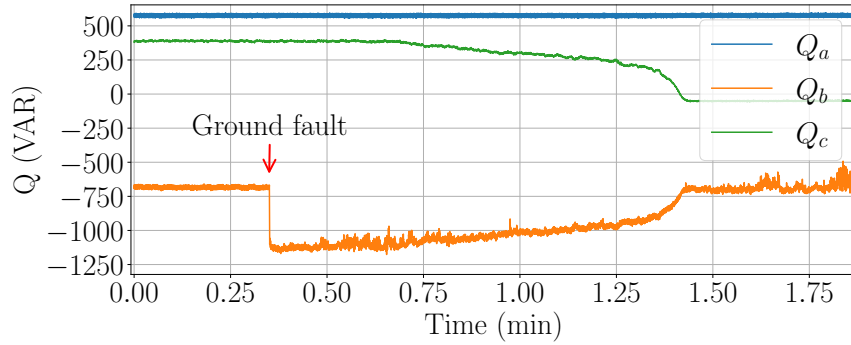
Figure 6-33: Phase c ground fault at 0.6 min immediately followed by an open-circuit of the bc heating element

is due to faulty load behavior, the location can be narrowed down to the energized equipment as identified by a NILM classifier.

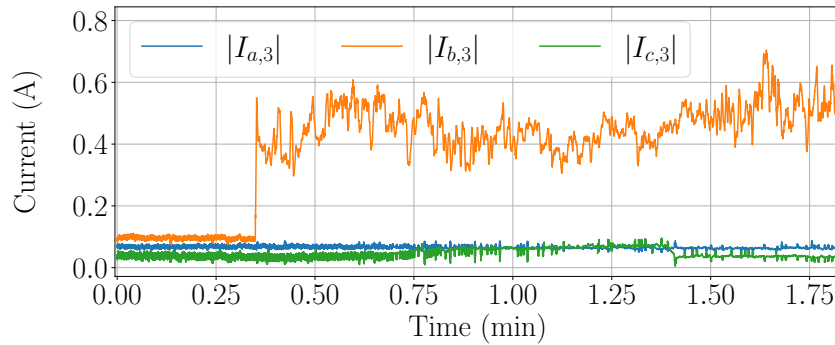
These ground faults were both immediately followed by the open-circuit of a bc heating element of a MPDE JW heater. As shown in Fig. 6-33 and Fig. 6-34 the ground faults were followed by a slow change in power until about $t = 2.4$ min and $t = 1.4$ min, respectively, as a result of the open-circuit. This heater is nominally



(a) Aggregate real power.



(b) Aggregate reactive power.



(c) Third harmonic current.

Figure 6-34: Phase b ground fault at 0.35 min immediately followed by an open-circuit of the bc heating element.

4.5 kW, so as expected, the open-circuited heating elements result in a decrease of about 750 W on phases b and c , and a split of about 433 VAR on those phases. The root-cause analysis for the MPDE JW heater faults is presented in Appendix B.

Table 6.11: Relative line current magnitudes of unbalanced VFD

	$\frac{I_{ak,ss}}{I_{bk,ss}}$	$\frac{I_{bk,ss}}{I_{ck,ss}}$	$\frac{I_{ck,ss}}{I_{ak,ss}}$
k=1	0.84	0.91	1.30
k=3	0.36	1.34	2.07
k=5	0.86	0.93	1.25
k=7	0.83	0.93	1.29

6.2.3.5 Unbalanced Nonlinear Loads

For the purposes of this section so far, ideally balanced and sinusoidal voltages have been assumed. However, unbalanced supply voltages may be especially prevalent on some systems such as on marine microgrids [174] and on grids with high penetration of distributed energy resources such as photovoltaics [175]. For a VFD, even small voltage imbalances can cause significant variation in line current waveforms [176]. As the voltage imbalance increases, the same amount of power still needs to be supplied, resulting in current peak increases. This means an increased current through the diodes and increased stress on the components. With a large enough supply voltage imbalance, the balanced double-pulse line current waveform eventually becomes a single pulse waveform. As described in Section 6.2.2, for a balanced non-linear load, triplen components are zero-sequence. More generally, the third harmonic line currents are:

$$\begin{aligned}
 i_{a3}(t) &= A_{ab3} \cos(3\omega t + \delta_{ab3}) - A_{ca3} \cos(3\omega t + \delta_{ca3}) \\
 i_{b3}(t) &= A_{bc3} \cos(3\omega t + \delta_{bc3}) - A_{ab3} \cos(3\omega t + \delta_{ab3}) \\
 i_{c3}(t) &= A_{ca3} \cos(3\omega t + \delta_{ca3}) - A_{bc3} \cos(3\omega t + \delta_{bc3}).
 \end{aligned} \tag{6.19}$$

When the third harmonic phase current magnitudes and phase angles are not equal, third harmonic line currents will be measured. That is, the uneven current distribution in the rectifier can cause uncharacteristic positive-sequence triplen harmonics in the line currents [177, 178].

A three-phase motor controlled by a VFD was tested using the laboratory grid

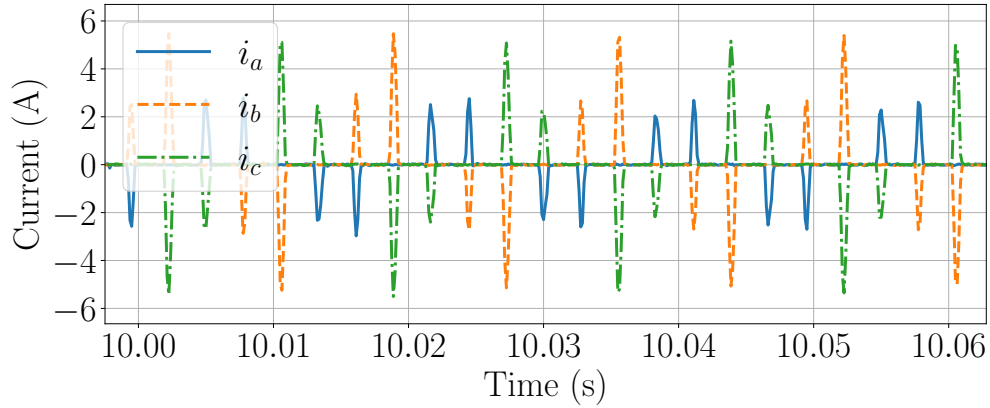


Figure 6-35: Unbalanced line currents for a motor controlled by a VFD.

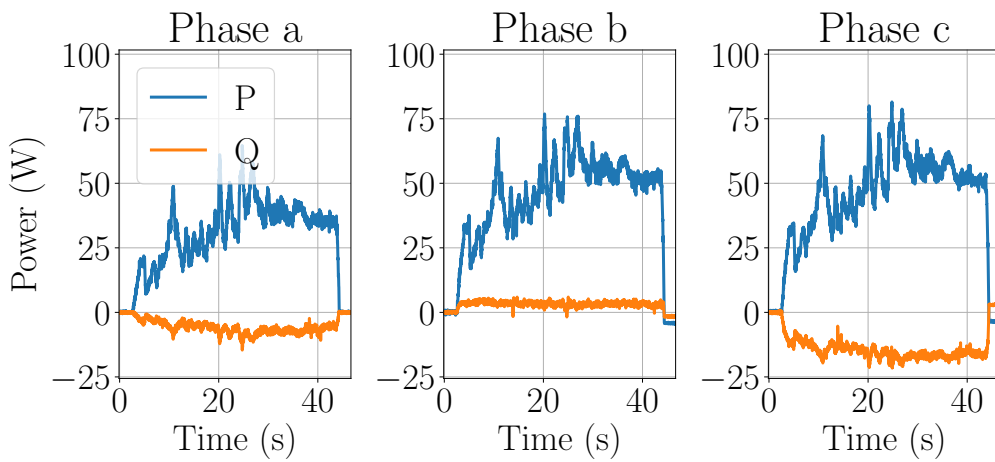


Figure 6-36: Power envelopes of a motor controlled by a VFD.

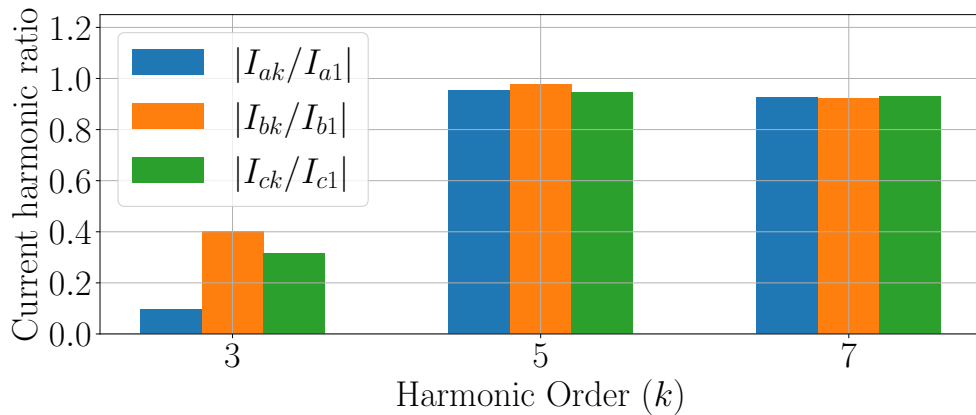


Figure 6-37: Magnitude of third, fifth, and seventh harmonic current envelopes normalized by the fundamental of a motor controlled by a VFD.

voltage. The line currents and per phase power are shown in Fig. 6-35 and Fig. 6-36, respectively. The line-to-line voltages during the testing period averaged $V_{ab} = 292.3$

V, $V_{bc} = 295.6$ V, and $V_{ca} = 295.5$ V. Even these small voltage imbalances resulted in an asymmetric current conduction. Fig. 6-37 shows the average current harmonic magnitudes normalized by the fundamental for the load duration shown in Fig. 6-36. Unlike the balanced case of Fig. 6-18, there are significant third harmonic line currents. The unbalanced line currents also result in relative line current magnitudes that deviate from the expected values of [1,1,1], as shown in Table 6.11 for the fundamental, third, fifth, and seventh harmonics. These values were calculated at the detected on-event with $\Delta t = 1$ s for steady-state calculation. Since the line currents are not balanced, the true phase currents cannot be determined. However, the imbalanced line currents and presence of third harmonic currents are a telling sign of increased stress on components and possible power quality issues.

6.3 Chapter Summary

This chapter presented example case-studies of applying nonintrusive load monitoring for fault detection and diagnostics. The first case-study involved tracking the health of an HVAC system. Ventilation systems play important roles in residential, commercial, industrial, and transportation systems. They are critical energy consumers and generate substantial electrical waste when run under faulted conditions. Combining knowledge of physical characteristics of installed fans with their respective signatures enables a NILM to effectively monitor fans and their motors, providing a method or additional method for detecting and diagnosing faults. The methods presented are especially valuable for aggregate monitoring of microgrids with highly-variable supply frequency. The value of power monitoring for detection of filter clogging and leaking was demonstrated. If a fan is infrequently energized and secured, clogs or leaks need to be detected while the fan is operating. Thus, slot harmonic tracking is employed to correlate specific individual fan behavior with power consumption. Fan speed estimation and filter clogging detection was demonstrated via slot harmonics. The presented experiments demonstrated that supply frequency needs to be tracked so that clogging or other fault conditions can be distinguished from shifts in supply frequency. The

algorithms and hardware were demonstrated in a field microgrid on USCGC Marlin, both in-port and at sea. The in-port experiments, which occurred with a relatively stable utility frequency, demonstrated how power and slot harmonic information can be used in tandem to track ventilation health in the field. Furthermore, testing on USCGC Marlin confirmed that these methods work well in a practical multi-machine environment served by a common electrical source. The motivation for and the applicability of the multi-peak SSE algorithm was demonstrated in this multi-machine environment. The results from USCGC Marlin at sea confirmed that motor speed can be tracked even with a highly variable supply frequency on an isolated microgrid. The laboratory and ship results together demonstrate that with a combination of equipment operation history, power consumption, and slot harmonic tracking, it is possible for a NILM system to identify fans and monitor HVAC systems for fan and motor health and speed.

This chapter then presented fault detection and diagnostics in three-phase systems, using ungrounded delta-configured systems as an example. A three-phase system presents additional challenges for monitoring due to potential imbalances in the system, some of which are inherent to specific loads, but are other times indicative of a fault scenario. As was demonstrated, abrupt faults are associated with changing electrical consumption and often evolve in a predictable way. Electrical analysis unveiled degraded shipboard loads, revealing corrosion and degradation of critical equipment. Monitoring these systems is important for ensuring efficient and safe operation. A practical three-phase system is never perfectly balanced. Changes in system balances for currents, voltages, and harmonic content during transients can inform load identification and diagnostics. Nonintrusive load monitoring can be used as a condition-based maintenance aid, by using power stream data to detect load or system degradation before complete failure occurs.

Chapter 7

Dashboard User Interface

For a nonintrusive load monitoring system to be an effective condition based maintenance tool, the data needs to be presented as an intuitive decision aid for users. This chapter presents diagnostic metrics that have proven useful for the shipboard case-studies. Then the NILM Dashboard and its easy-to-understand visual displays are presented.

7.1 Diagnostic Indicators

Nonintrusive load monitoring records the electrical signature and the operating schedule for a piece of equipment, allowing for a broad range of fault diagnostic methods. It is crucial to select the appropriate parameters for condition-monitoring to create a useful tool that provides actionable information for the end-user. To this end, the following six parameters were selected for equipment diagnostics:

- Power: Steady-state real power.
- Power factor: The ratio of real power to apparent power.
- Average run duration: Time between activation and shutdown.
- Total run time: Total time the equipment is online over a 24-hour period.
- Daily actuations: Number of discrete operations per day.
- Duty cycle: Total run time divided by total time period (24-hours).

These parameters work well for the equipment monitored on the marine micro-grids, but other metrics may be useful in other environments. The equipment in this work consists largely of motors, pumps, and heaters. All of these metrics may be useful for detecting material degradation of equipment, such as mechanical wear and corrosion. A change in power demand may indicate a worn motor bearing, or a change in power factor could be a sign of corroded heating elements. Many of the heaters and pumps monitored by the NILMs in this work are controlled by closed-loop automated systems such as tank-level sensors or thermostats. Thus, the average run duration, total run time, daily actuations, and duty cycle track equipment behavior and are useful for finding sensor and automation faults that might cause equipment to run too frequently or not enough. As was shown, a broken tank level indicator or failed thermostatic controller can, for example, cause equipment to activate in repeated short-cycles or run for excessively long periods. As equipment experiences material degradation, feedback control may also alter average run duration, total run time, daily actuations, and duty cycle. For instance, for the MPDE JW heater, as the heating elements successively fail, the duty cycle of the heating system increases in order to maintain the heat output. Finally, it is important to note that a single extended pump run or even a few frequent runs is not necessarily a cause for concern. These may occur during manual operation or maintenance. This is accounted for by tracking the average over 24 hours for averaged parameters, i.e., power, power factor, average run duration, and duty cycle, and the total over 24 hours for summated parameters, i.e., total run time and daily actuations. The 24-hour window serves to help prevent falsely displaying an alarm as the result of a brief anomaly. The 24-hour period can easily be adjusted for different applications where loads activate less or more frequency, or tighter controls are required.

Condition-based maintenance parameters are communicated on the NILM Dashboard via “green-yellow-red” diagnostic gauges, as demonstrated below. The green region represents healthy operation, the yellow region represents a trouble warning, and the red region is a more definitive fault alarm. Determining the proper threshold for each region on the gauges is crucial to making it a useful, actionable tool for the

ship’s crew. A variety of methods has been proposed to determine fault thresholds for industrial applications [179]. For this work, a statistical process control (SPC) method is implemented. Effective SPC attempts to differentiate between natural variations and variations that are due to process failure [180]. Deviation from the historical data for any parameter is evidence of a possible fault. SPC provides a method to determine exactly how much deviation is acceptable and when a deviation should trigger a fault warning. The SPC method consists of determining a centerline, an upper control limit (UCL) and a lower control limit (LCL). Warnings are issued when a parameter reaches the upper or lower control limits.

First consider a continuous variable, in which the variable can fall anywhere within a particular range of values, such as power, power factor, average run duration, total run time, and duty cycle. Considering the standard normal distribution, SPC uses the arithmetic mean (μ) of the parameter as the centerline [180]. The UCL and LCL are defined as,

$$UCL = \mu + n\sigma, \quad LCL = \mu - n\sigma, \quad (7.1)$$

where σ is the standard deviation and n is an integer that sets the distance of the control limits. For a parameter with a normal distribution, $n = 3$ is the accepted industry standard for a fault warning [180,181], and corresponds to the red region on the gauges. The choice of control limits affects the risks of Type I or Type II errors, where Type I errors refer to incorrectly reported faults and Type II errors refer to missed faults. By widening the control limits, the risk of Type I errors decreases; however there is an increased risk of Type II errors as more data points will fall within the control limits and be viewed as normal. Contrarily, if the control limits are narrowed, there is an increased risk of Type I errors and decreased risk of Type II errors, as more data points will fall outside the control limits and will be classified as fault conditions. The “3-sigma” rule is conservative and designed to minimize the risk from false alarms. However, analysts often suggest using two sets of limits; *action limits* at “3-sigma” and *warning limits* at “2-sigma” [180]. For this application the intermediate control at $n = 2$ corresponds to the yellow region on the gauge. Addition

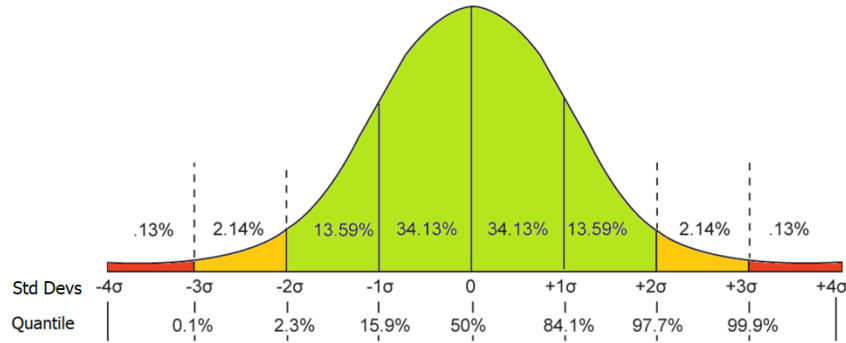


Figure 7-1: Probability density function of a normal distribution showing progressive thresholds for fault detection. Colors correspond to the red, yellow, and green regions on the dashboard gauges.

of the intermediate control limit provides more rapid detection of faults. Figure 7-1 shows how SPC maps the probability density function (PDF) of a normal distribution to the green, yellow and red regions of the gauges. The percentages in each region correspond to the likelihood that the variable falls within that particular range of values. The quantile values displayed at the bottom of Fig. 7-1 correspond to the probability that some variable, X , is less than or equal to some value, x , where x is the centerline and control limits. This can be written as a cumulative distribution function (CDF),

$$F(x) = Pr(X \leq x). \quad (7.2)$$

The inverse cumulative distribution function (ICDF), or quantile function,

$$F^{-1}(p) = x, \quad (7.3)$$

solves for the x value that would make $F(x)$ return some probability, p .

The SPC process can be adapted if the normal distribution does not properly fit the data. For example, the Weibull distribution is often used in machinery reliability applications [181]. The PDF for a two-parameter Weibull function is:

$$f(x) = \left(\frac{\beta}{\alpha}\right) \left(\frac{x}{\alpha}\right)^{\beta-1} e^{\left(\frac{-x}{\alpha}\right)^\beta} \quad (7.4)$$

where α is the scale parameter and β is the shape parameter. To create the gauge regions for a non-normal distribution, the probability quantiles (p) should match the red, yellow, and green regions of the normal distribution in Fig. 7-1. The ICDF function for a Weibull distribution is:

$$x = F^{-1}(p|\alpha, \beta) = \alpha(-\ln(1 - p))^{\frac{1}{\beta}}. \quad (7.5)$$

Therefore, the centerline can be found by setting p equal to 0.50 and solving for x . The upper and lower yellow threshold levels can be found by setting p to 0.977 and 0.023, respectively. Similarly, the upper and lower red threshold levels can be found by setting p to 0.999 and 0.001, respectively. This ensures that the probability of an alarm detection is the same regardless of the PDF selected for modeling. Each parameter monitored by the NILM Dashboard can be analyzed individually and the gauges adjusted to provide diagnostic warnings at appropriate levels.

Next consider a discrete variable, in which the variable has finite values, such as the number of *daily actuations*. Because these can only occur as integer values, the CDF is not continuous and increasing; thus the generalized inverse distribution function will be used instead of the ICDF [182]. The generalized inverse distribution function can be expressed as,

$$x = F^{-1}(p) = \inf\{x \in \mathbb{R} : F(x) \geq p\}, \quad (7.6)$$

where \inf is the infimum, or the greatest lower bound. Similar to a continuous function, centerline and upper and lower limits can be determined by setting p to the appropriate values and solving for x .

The statistical process control method was used on historical NILM data from USCGC Spencer. Histograms for each metric were created. Then, probability densities were fit to the data. For continuous functions, the histograms are modeled with multiple probability density functions (PDF), while a discrete function is modeled with probability mass functions (PMF), and the best fit function is selected. Note that the model for daily actuations is only for *if* the load ran that day. If there were

no runs for a given day, that is demonstrated with a grayed out gauge. Depending on the PDF or PMF selected, the fault detection thresholds are set as shown in Fig. 7-1 or using a quantile function as described in Equations (7.5) and (7.6). This process was repeated to determine control limits for each of the monitored loads. This information was communicated to the ship’s crew using the NILM Dashboard graphical user interface.

If a user knows that a piece of equipment should be operating, but the NILM Dashboard shows the load has not been operational, this would alert the user to a possible fault. There are several reasons the daily actuations parameter could drop to zero, even if the load should be operating. First, it could be indicative of a broken sensor, such as a tank-level sensor or temperature sensor, resulting in the load not turning on even when it should be. Second, it could be indicative of a complete failure of a piece of equipment. Alternatively, it could be due to a degraded piece of equipment, resulting in a load signature that has changed to the point where load identification accuracy is reduced.

7.2 Dashboard Overview

After disaggregating individual load events, the NILM Dashboard software indicates to watchstanders when pieces of machinery are energized or secured [10]. After additional historical data is collected the Dashboard can perform historic diagnosis and fault detection [10]. The Dashboard consists of three main views: Timeline, Metrics, and Historic.

7.2.1 Timeline View

The “Timeline” view provides a live view of equipment status, allowing the user to see loads energized and secured in real-time. The user can monitor the entire plant or hide certain equipment from view, allowing for increased attention on select loads. The time window can be adjusted to the user’s choosing, either through zoom/pan functions or by selecting a date range. The timeline tool provides the user with a

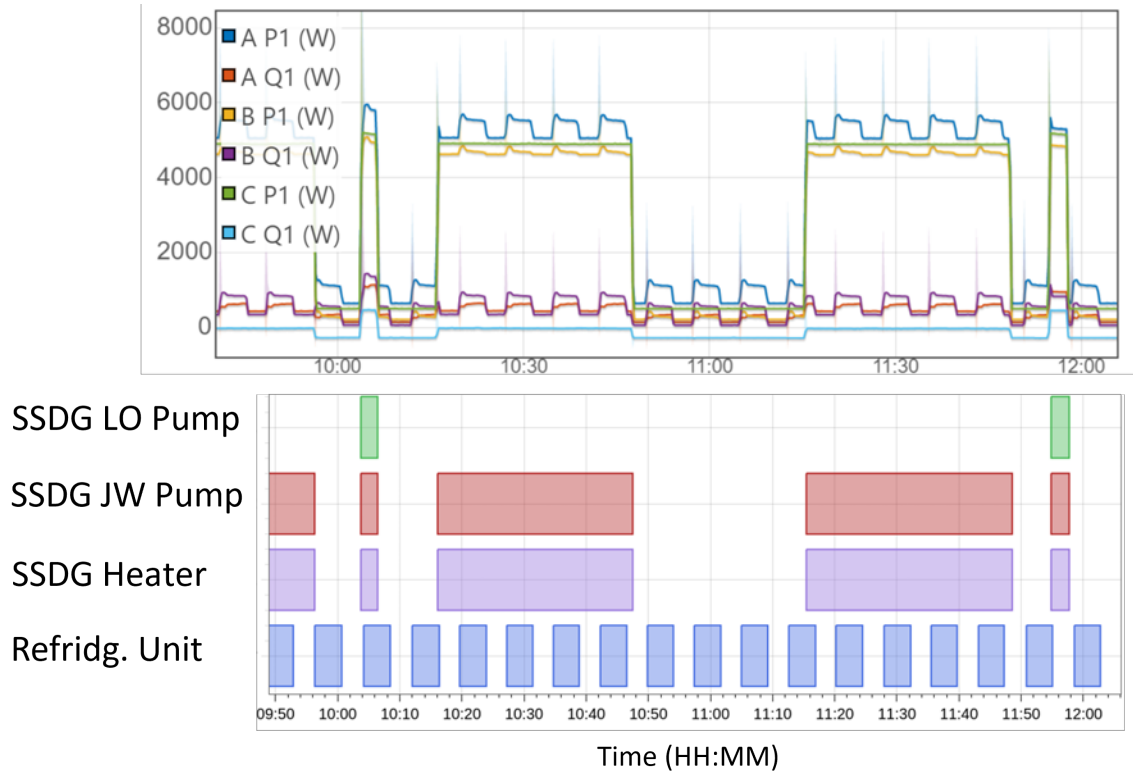


Figure 7-2: Power stream displayed above. NILM Dashboard timeline view displayed below, where colored blocks indicate periods where a load is energized.

compact picture of plant operations and the ability to easily investigate any apparent anomalies. For example, Fig. 7-2 shows an example two-hour power stream from USS Indianapolis time-aligned with the NILM Dashboard timeline view. Colored blocks indicate periods where a load is energized.

7.2.2 Metrics View

The “Metrics” view provides the user with a set of diagnostic indicators for a selected piece of equipment. These include the metrics presented in Section 7.1, including real power, power factor, total run duration, average run duration, daily actuations, and duty cycle. Each metric is displayed as a gauge with green, yellow, and red sections. The colored sections are derived from nameplate data, known usage patterns and statistics from previous normal operation. Green indicates normal operations, while yellow and red indicate increasing likelihood of a fault. The gauge needle is the

average metric value for the last 24 hours. The Metrics view provides an analysis of individual equipment health and helps direct initial troubleshooting efforts.

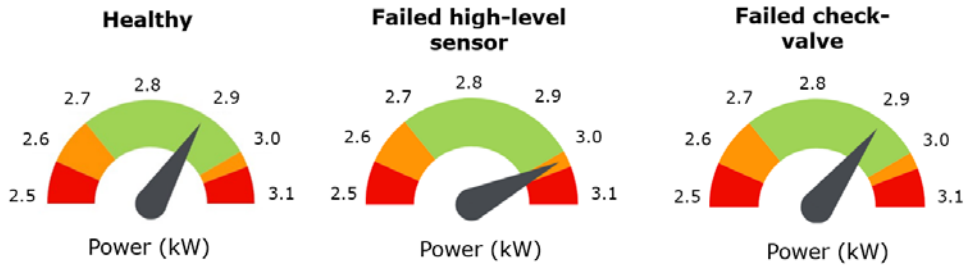
7.2.3 Historic View

The “Historic” view provides short- and long-term trend data to supplement the analysis from the Metrics view. This tool allows the user to select a single load and any of the six metrics. The Historic View is presented as a bar graph, where each bar represents one day. The Historic view is designed to give the user the ability to assess trend data, track behavior over time, and make informed decisions for condition-based maintenance.

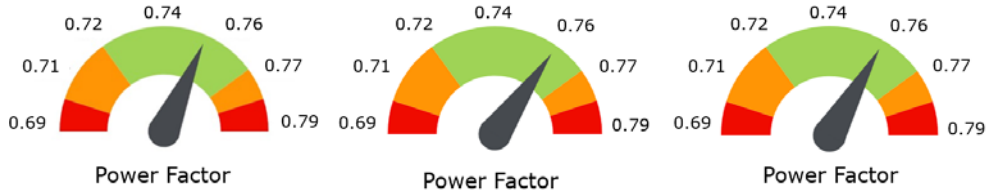
7.3 USCGC Spencer Demonstration

On USCGC Spencer, the first prototype of the NILM Dashboard graphical user interface was installed in July 2018 as was shown in Fig. 1-3c, overlooking the engine room. The crew could navigate through different loads on the port and starboard panels to plot on the Dashboard to verify operation. The two computers were connected via Ethernet, such that one Dashboard interface could display the detected events from both subpanels from a single location. The load identification was performed in real-time as data was collected on the ship using the multi-scale identification framework from Section 3.1. The Joule [52] data processing framework was utilized to perform load identification using the `/meter/`prep streams as input.

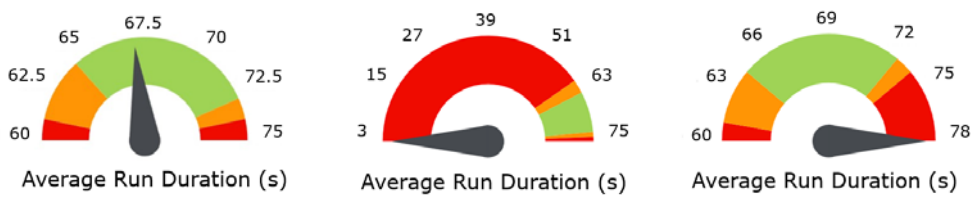
To demonstrate the ability of Dashboard metrics to provide indication of fault conditions, the metrics view is shown in Fig. 7-3 for three scenarios of the graywater pump: healthy condition, failed high-level tank sensor, and failed check valve. The graywater system and these fault conditions were introduced in Section 1.2. Each row in Fig. 7-3 shows one metric for a sample day in each of the scenarios, with healthy operation on the left, the failed high-level sensor in the middle, and the failed check valve on the right. The gauge needle is the average metric value for that day of operation. The red region of the gauge will expand accordingly if the value for



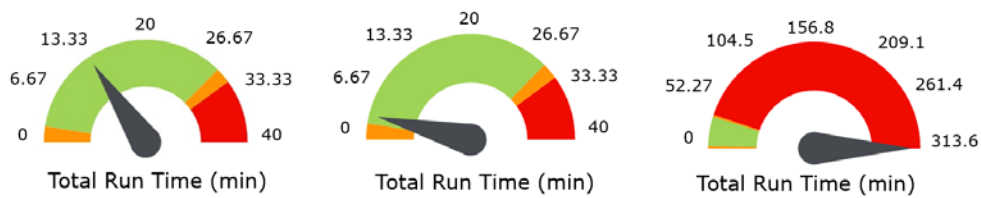
(a) Power.



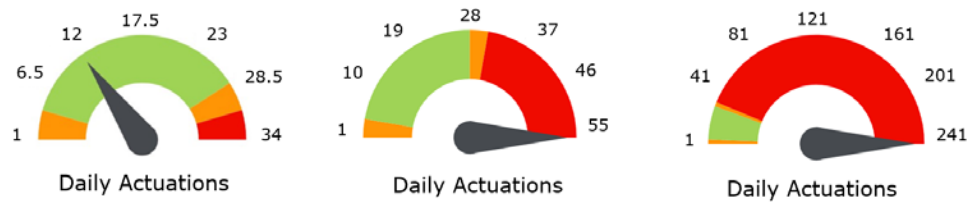
(b) Power factor.



(c) Average run duration.



(d) Total run time.



(e) Daily actuations.

Figure 7-3: Metrics for various conditions of graywater pump operation: healthy (left), failed high-level tank sensor (middle), and check valve failure (right).

that day places it outside the gauge limits visible on a “normal” day. In Fig. 7-3a and Fig. 7-3b, the power and power factors are relatively healthy for all three scenarios. This is expected, since these two fault scenarios do not have an effect on the steady-state power consumption of the load. Fig. 7-3c shows that both the high-level sensor fault and check valve fault result in the average run duration to be in the red region. However, these two faults have opposite behaviors. The high-level sensor fault results in an average run duration which is much shorter than normal and the check valve failure results in an average run duration which is higher than normal. Due to the almost continuous flow of water into the holding tank, the pump would have to run for longer as it would attempt to empty the tank. Typically the average run time is within a narrow band, so the approximately ten second increase in run duration during the check valve failure was significant. Fig. 7-3d shows that the total run time for the high-level sensor fault was still in the green region, whereas the the check valve failure was in the red region. Finally, Fig. 7-3e shows that the daily actuations for both the failed high-level tank sensor fault and check valve failure are much greater than normal. These examples demonstrate the ability of the average run duration, total run time, and daily actuations metrics to diagnose load cycling fault conditions.

7.4 USS Indianapolis Demonstration

After collecting data from USS Indianapolis and verifying individual load transients, load identification was run on historical data streams. Fig. 7-4 shows eight days of the Dashboard timeline view, with equipment listed vertically on the left side, and colored blocks representing time equipment is energized. The shown equipment include the splitter gear control oil (CO) pump, hydraulic power pack (HPP) booster pump, HPP cooling pump, splitter gear lube oil (LO) pump, ship service diesel generator (SSDG) lube oil (LO) pump, SSDG jacket water (JW) pump, and SSDG heater unit.

Along with displaying equipment operational status, metrics about load operational history can provide insight for condition-based maintenance and fault detection. Here the duty cycle is demonstrated, providing a diagnostic metric of operating

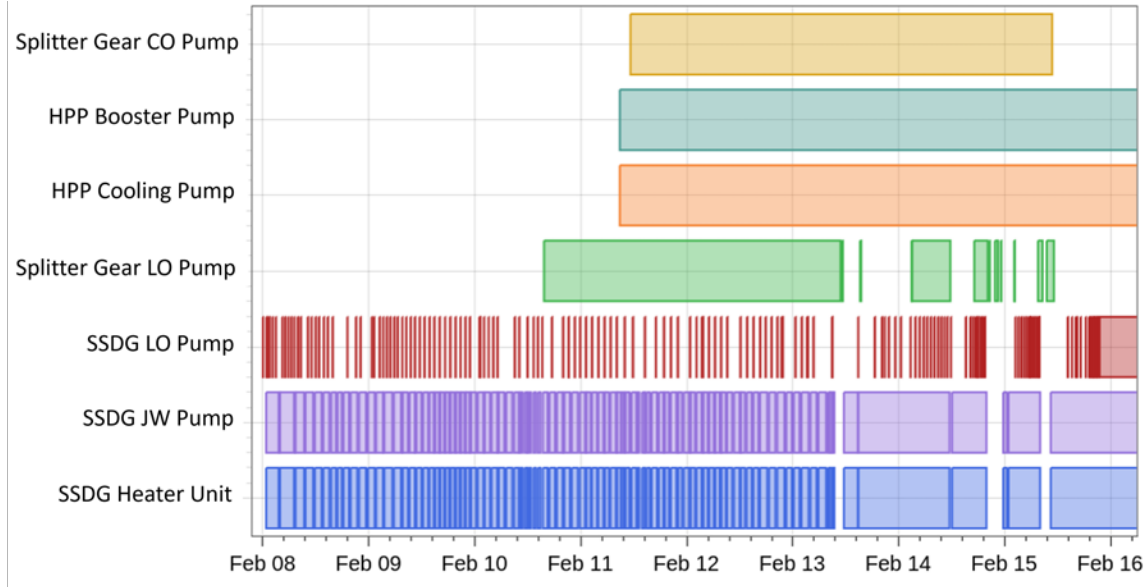


Figure 7-4: Dashboard timeline view of equipment from two sub-panels for eight days of operation.

Table 7.1: Load duty cycles.

Equipment	In-Port	Underway
SSDG Skid Heater	.4543	.6140
SSDG LO Pump	.0359	.1850
SSDG JW Pump	.4543	.6140
Splitter Gear LO Pump	.1157	.6899
Splitter Gear CO Pump	0	.8837
HPP Booster Pump	0	1.0
HPP Cooling Pump	0	1.0

time. The duty cycle is calculated as $D = T_{on}/T$, where T_{on} is the time the load is energized during the time period of interest, T . It is important to characterize the duty cycle based on the status of the ship, because ship status drives the frequency of equipment use. Table 7.1 takes seven of the loads observed on LCS and presents the calculated duty cycle for in-port (i.e., the generator is on standby) and underway (i.e., the vessel is at-sea on generator power) periods. In-port duty cycles were calculated for a 3 day period, from February 8, 2020 to February 10, 2020. Underway duty cycles for the auxiliary machine room subpanel were calculated for a 14-day period from February 11, 2020 to February 24, 2020, and duty cycles for the water jet machinery room subpanel were calculated for a 5-day period from February 11, 2020 to Febru-

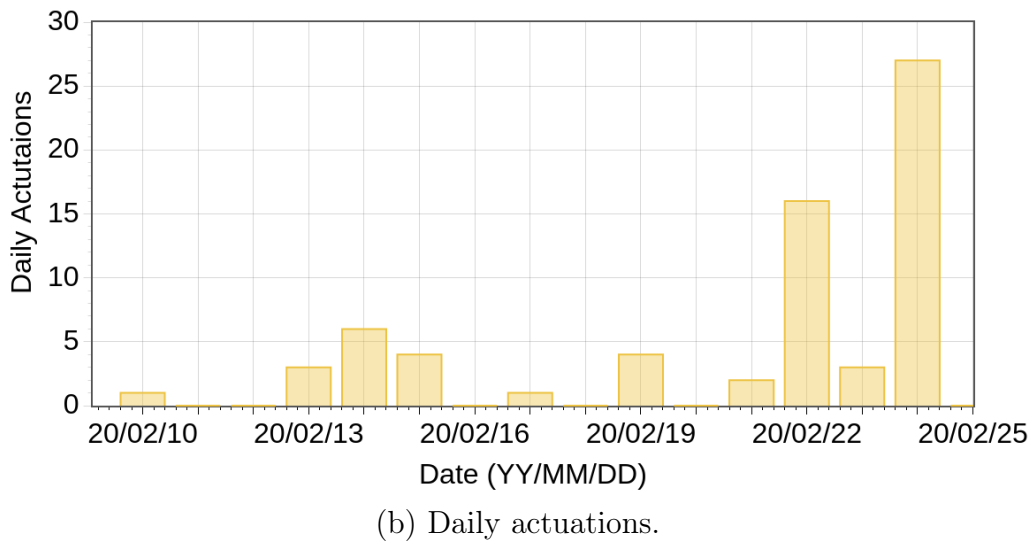
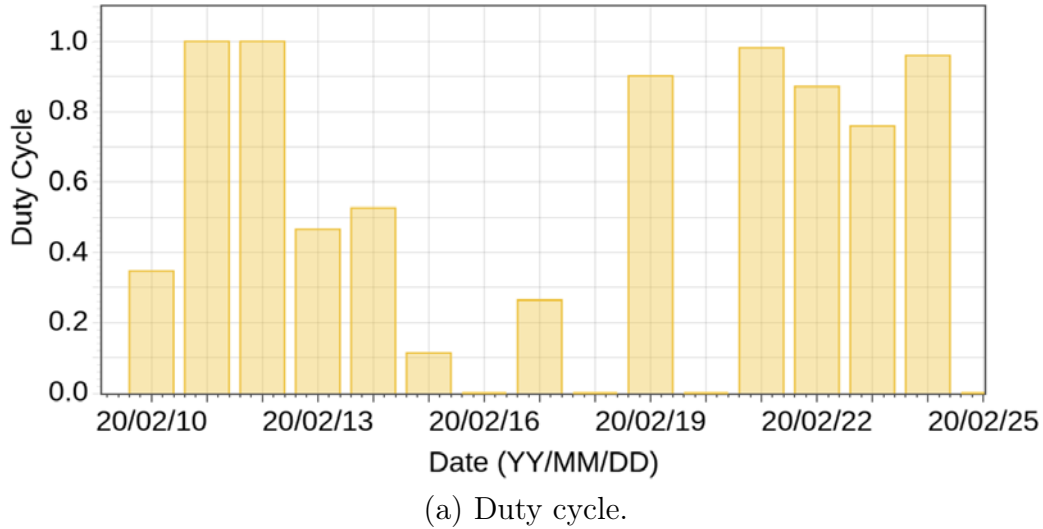


Figure 7-5: Historical view of the splitter gear LO pump.

ary 15, 2020. These duty cycles provide valuable insight into equipment operation. The splitter gear LO pump, splitter gear CO pump, HPP booster pump, and HPP cooling pump have different characteristic behaviors for different ship statuses, with clear delineations between underway and in-port.

From the calculated metrics, a historical view was generated to view daily trend data. To illustrate, historic views of the splitter gear LO pump are presented. The splitter gear for the propulsion system contains both a LO pump and CO pump. The splitter gear LO pump is energized at certain shaft speed ranges while the ship is underway, with the lubrication oil being circulated by an attached gear-driven pump

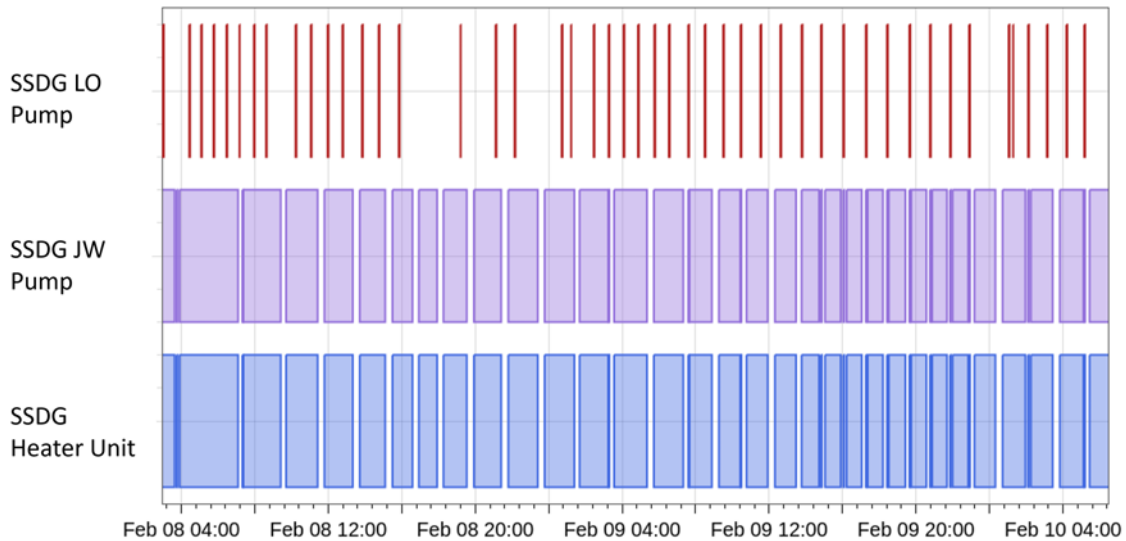


Figure 7-6: Timeline view of two days of operation of the SSDG auxiliary skid while the vessel is in-port.

when the shaft speed permits. Fig. 7-5 shows historic views of the daily duty cycle and daily actuations for the splitter gear LO pump. On February 24, the pump was energized for 96% of the day, in the range expected for underway operations. However, by viewing the number of daily actuations, it becomes clear the splitter gear LO pump was cycling more often than usual, with 27 actuations. This cycling of the splitter gear LO pump indicates that the separate gear-driven pump is not able to maintain the required lubrication oil pressure, most likely a result of lower ship speeds. This situation occurs when vessel speed is right above the separate gear-driven pump’s “take-off” rpm, or the required shaft speed for the gear-driven pump to provide the required lubrication oil pressure. Long periods of operation in this state can be detrimental to the electric pump, as short cycling can add additional stress and strain, wearing down the pump and shortening its useful life.

The timeline view, in conjunction with the historical view can provide additional insight into ship operation. Here, the auxiliary skid on USS Indianapolis will serve as an example. USS Indianapolis is equipped with four ship service diesel generators (SSDGs). These generators have a skid attached to the free-end of the engine, which provides the auxiliaries required to run the generator. The auxiliary skid includes two

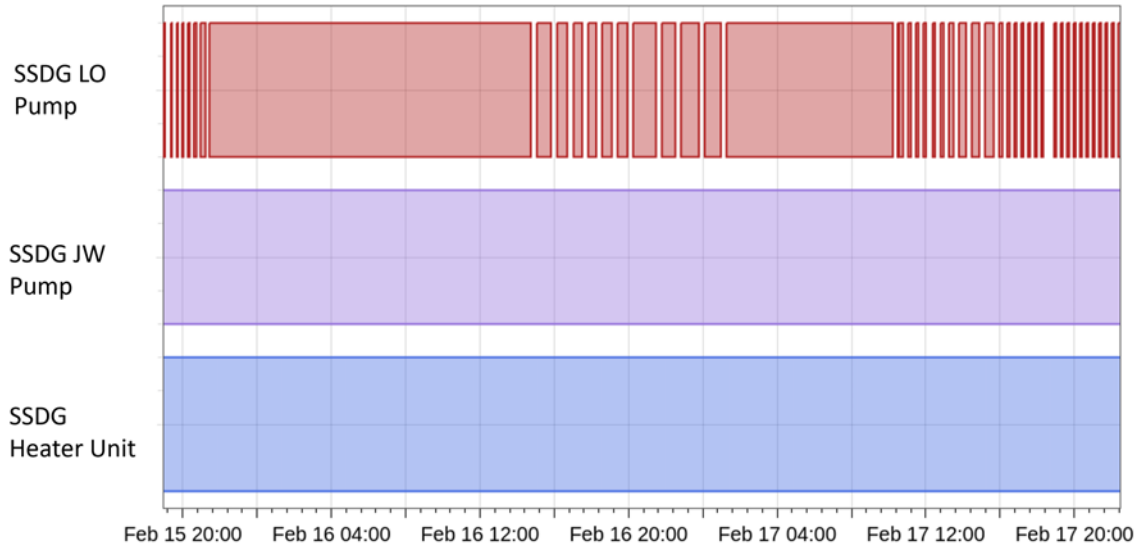
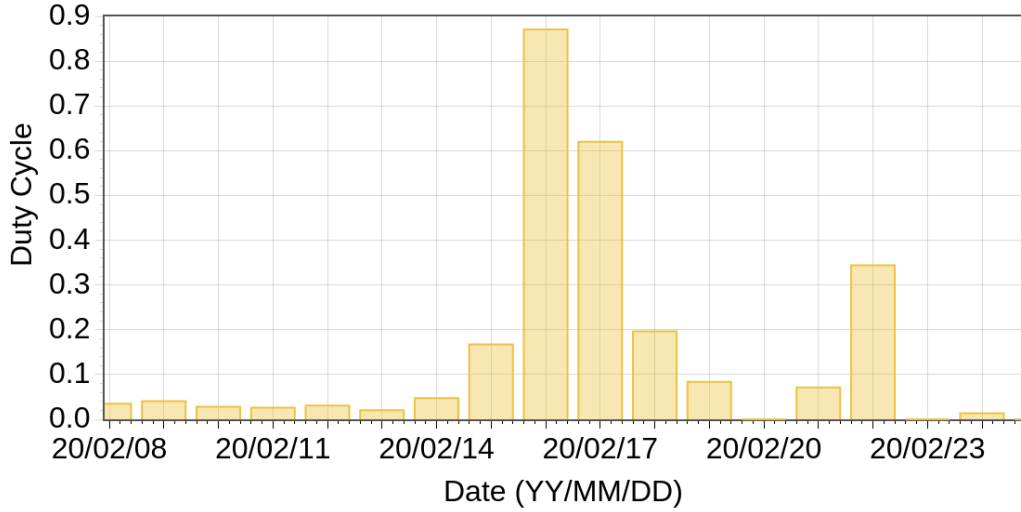


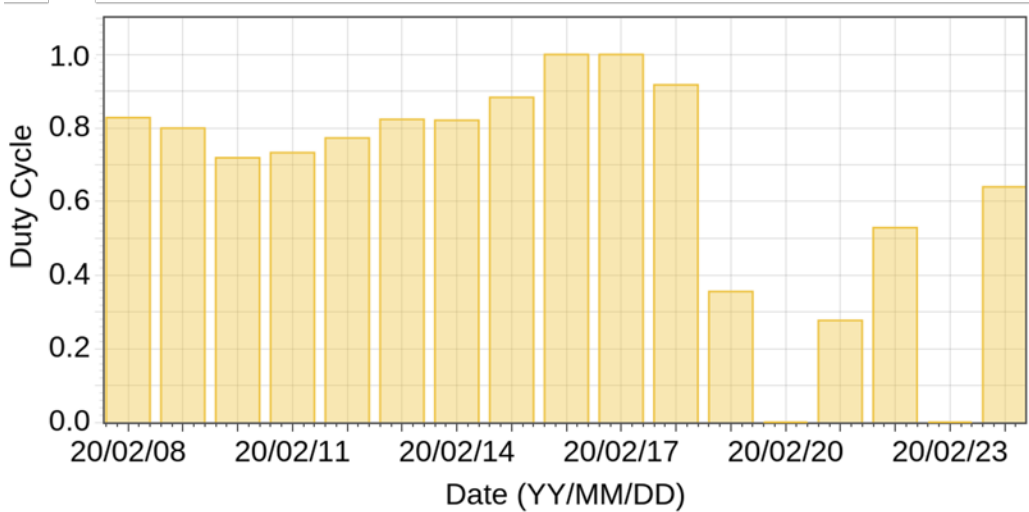
Figure 7-7: Timeline view of two days of operation of the SSDG auxiliary skid while the vessel is underway.

pumps and two heaters: one pump circulates lube oil and the other circulates jacket water throughout the SSDG. Both heaters actuate simultaneously, and are rated to draw a combined 12 kW. Furthermore, the SSDG jacket water (JW) pump, which is rated at 1.12 kW, runs simultaneously with the heating elements. The SSDG lube oil (LO) pump is rated at 1.5 kW. Fig. 7-6 and Fig. 7-7 show the timeline view of the SSDG auxiliary skid for two days of in-port operation and two days of underway operation, respectively. From the timeline view, it is clear that these loads have different behaviors when comparing the two time periods. For the SSDG LO pump, the total on duration is much higher in Fig. 7-7. For the SSDG JW pump and SSDG heater unit, the total on duration looks similar, but the number of actuations is much higher in Fig. 7-6.

These behaviors can be more closely examined by viewing the historical view plots. Fig. 7-8a shows the historical view daily duty cycle for the SSDG LO pump. When the vessel is in-port (and the generator is in standby), the SSDG LO pump exhibits a very low duty cycle, of approximately 0.04 for the period before February 15. On February 16 and 17, the duty cycles are very high, at approximately 0.9 and 0.6, respectively. The SSDG JW pump exhibits a small increase in duty cycle at this



(a) SSDG lube oil pump.



(b) SSDG jacket water pump.

Figure 7-8: Daily duty cycle of the SSDG lube oil pump and jacket water pump.

time, with duty cycles on February 16 and 17 of 1.0, as shown in Fig. 7-8b. This is in agreement with the SSDG’s auxiliary skid circulating lube oil and jacket water to maintain engine block temperatures for immediate operation. Contrasting this is February 20, during which both the SSDG LO pump and JW pump have a duty cycle of zero. There are two operating states, which are in stark contrast with each other, that present this duty cycle profile. First is when the the generator is in “cold iron” status with the auxiliary skid not energized to maintain block temperature for immediate operation. This is typically done during maintenance and repair activities

or when there is no immediate need to run the generator. Second, a duty cycle of zero will also occur if the generator is in continuous operation, because attached gear driven pumps will provide jacket water and lubricating oil circulation rendering the auxiliary skid pumps and heaters unnecessary. In discussions with the vessel's crew, it was determined that the latter condition is what has been observed in the data stream. However, having a complete set of power streams from all generators would also enable the actual operation of the SSDG to be determined.

7.5 Chapter Summary

This chapter demonstrated NILM Dashboard for providing real-time diagnostics to the end-user. The platform was successfully tested on power stream data from USCGC Spencer and USS Indianapolis. The Dashboard platform was installed for real-time monitoring on USCGC Spencer. Examples were presented for each of the three proposed views, Timeline, Metrics, and Historic. With accurate detection of faults and anomalous behavior prior to a ship's deployment, servicing and replacement of equipment can be better scheduled. That is, if presented accurately and in an intuitive way, power stream data can serve as a metric for conducting condition based maintenance, rather than recurring preventative maintenance.

Chapter 8

Conclusion

Nonintrusive load monitoring has widespread applicability for monitoring of electronic and electromechanical systems and has been a widely researched topic for more than three decades [14, 62, 183]. The work presented in this thesis builds upon the decades of previous NILM research and advancements. Most research has focused on absolute accuracy of total load disgregation, but not computational complexity, usability, or concrete applications focused on mission needs [6]. No nonintrusive load monitor will be perfect under all conditions. Load classification algorithms depend on the type and number of target loads in any given data set and the features associated with the loads. This thesis aims to provide tools that can be tailored for different application-specific end goals. A non-exhaustive summary of contributions in this thesis includes the following content in each of the cited chapters. Chapter 2 presents advancements for power system measurement and preprocessing. Work in this chapter builds on previous work on spectral envelopes for power monitoring [9], and expands the practical limits of spectral envelopes for detecting load transients. The data acquisition hardware presented in the chapter offers a first application that can automatically track and reject utility content in nonintrusive current measurements. This chapter also offers a first look at techniques for extending nonintrusive monitoring to ring distribution networks. Chapter 3 presents tools for physics-based feature extraction and selection. A multi-scale framework is presented that separates events based on time-scale into geometric, statistical, and continuous events. This framework was designed

based on the USCG field data described in this thesis, but also previous field demonstrations, such as at a school [184] and research facility building [185]. Variation due to both healthy and faulty operation illustrates the necessity for adaptive classification techniques. A separability check is introduced for evaluating the reliability of a feature space in the presence of gradual load drift. Chapter 4 presents an adaptive classification and drift detection framework for geometric events. This framework can start with a limited initial dataset, while remaining robust to load features drifting over time and without requiring further ground-truth labelled data. Chapter 5 demonstrates load disaggregation for statistical events, building on the techniques presented with the multi-scale framework. The techniques are demonstrated with several loads that exhibit non-constant power demand. The chapter demonstrates that the combination of geometric and statistical event classification allows for more accurate energy estimation and condition tracking than either technique alone. Chapter 6 presents case-studies for applying nonintrusive electrical monitoring for fault detection and diagnostics. The chapter introduces techniques for slot harmonic tracking in a multi-machine environment. The chapter expands nonintrusive fault detection and diagnostics in three-phase systems by introducing a feature space based on changes in system balances. Finally, Chapter 7 describes NILM Dashboard, which displays data as an intuitive decision aid for end-users. NILM Dashboard was successfully installed and demonstrated on a shipboard microgrid, providing interaction with the ship’s crew.

There are several directions for future research. The feature extraction techniques presented in Chapter 3 are focused on finding physically-informed features that ensure load separability. However, there are scenarios that may arise in which power stream data from a single NILM is not sufficient for accurate identification and diagnostics. For instance, the question may arise: how can we distinguish two or more identical loads on a common electrical service? That is, a NILM may be monitoring an electrical panel that is supplying power to multiple loads of the same model, for example, in a manufacturing center. This may also occur on a ship that has its port and starboard loads supplied from a single panel, and not on separate port and

starboard panels. This is the case for the engine room subpanel on USCGC Thunder Bay. Furthermore, some relevant diagnostic measurements cannot be derived from the power stream. These include, but are not limited to, vibration, acoustic, and temperature data. If the data from these extra sensors can be collated in a central location, such as at the NILM computer, it can enhance the capabilities for diagnostic monitoring. Next, the adaptive classification framework presented in Chapter 4 can be improved upon by using unsupervised techniques to allow for clustering of unidentified loads. Clusters that form from the unidentified events could potentially be a fault condition of a load or a new load added to the system. Then user input could be requested to label any new clusters. The adaptive classifier can also be tested on additional shipboard datasets, such as on data from USCGC Marlin and USCGC Thunder Bay. These datasets currently only have labels from the initial installation and sparse labelling afterwards. The adaptive classifier can be used to help ease the burden of creating a labelled dataset. Another area of future work is to expand on the statistical techniques presented in Chapter 5 for use in other case-studies. For instance, USCGC Thunder Bay has a diesel electric propulsion plant. The monitored subpanel in the engineering control center (ECC) provides power to loads primarily related to the electric propulsion machinery. Changes in throttle position during driving and icebreaking operation are visible in the power stream. These fluctuations in power are well suited for analysis with the demonstrated statistical techniques. The analysis presented in Chapter 6 can be used for training a NILM to detect fault conditions using simulated data, since it is assumed the fault conditions cannot be purposefully introduced into the real system. It is hoped that the tools presented in this thesis can be used to provide actionable information for energy management and condition-based maintenance at more facilities and sites.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Shipboard Installations

A typical NILM installation onboard a ship monitors all or part of the main engineering space. Field installations of NILM systems have been conducted on several USCG and US Navy vessels. This appendix gives an overview of the various shipboard NILM installations and monitored loads relevant for this thesis. More specific details on the installations, mounting, and monitored loads can be found in [25, 27, 186–188].

A.1 USCGCs Spencer and Escanaba

Our longest installations provided data for over five years from two US Coast Guard (USCG) 270-ft. medium endurance cutters (MECs), Spencer and Escanaba, homeported in Boston, Massachusetts. Installations on both USCG Cutter (USCGC) Spencer and Escanaba were monitoring the port and starboard subpanels of the main engine room. These installations consisted of two boxes, one for the NILM CPU and one for the NILM meter hardware. The engine-room port-side installation of USCGC Spencer is shown in Fig. A-1a. USCGC Spencer has a radial ac electrical distribution. When at sea, power is provided by two ship service diesel generator (SSDG) sets. When in-port power is provided by either an aft or forward connection to utility shore power. The generator sets, along with the main propulsion diesel engines (MPDE) propelling the ship, are located in the ship's engine room. The SSDGs and MPDEs require auxiliary equipment, e.g., pumps and heaters, to maintain operational readi-



(a)

(b)

Figure A-1: USCGC Spencer installations: (a) Engine-room install with separate NILM software box on the left and NILM meter box on the right and (b) Auxiliary room install with NILM CPU, NILM meter, and touchscreen monitor all in a single box.

ness when in standby mode. Two electrical subpanels, port and starboard, which power these loads along with several other engine room loads critical for ship operation, are depicted in Fig. A-2. USCGC Spencer was additionally monitoring a power panel in the main auxiliary room, beginning in January 2019. The auxiliary room installation, as shown in Fig. A-1b, was the first prototype which contained the CPU, NILM meter, and touchscreen interface all in a single box. This was designed for a more consolidated installation. On USCGC Spencer, the first prototype of the NILM Dashboard graphical user interface (GUI) was installed. The crew could navigate through different loads on the port and starboard panels to plot on the Dashboard to verify operation. Fig. A-3 shows a conceptual diagram of the NILM installation locations on USCGC Spencer. This section has details on the shipboard systems and monitored electrical loads.

A.1.1 Main Diesel Propulsion Engine System

Both the port and starboard subpanels contain loads that service the main diesel propulsion engine (MPDE). While the vessel is at sea, one or both MPDEs will be energized. The NILMs monitor three auxiliaries of the MPDE: electric lube oil (LO) heater, electric jacket water (JW) heater, and electric prelube pump. While the engine

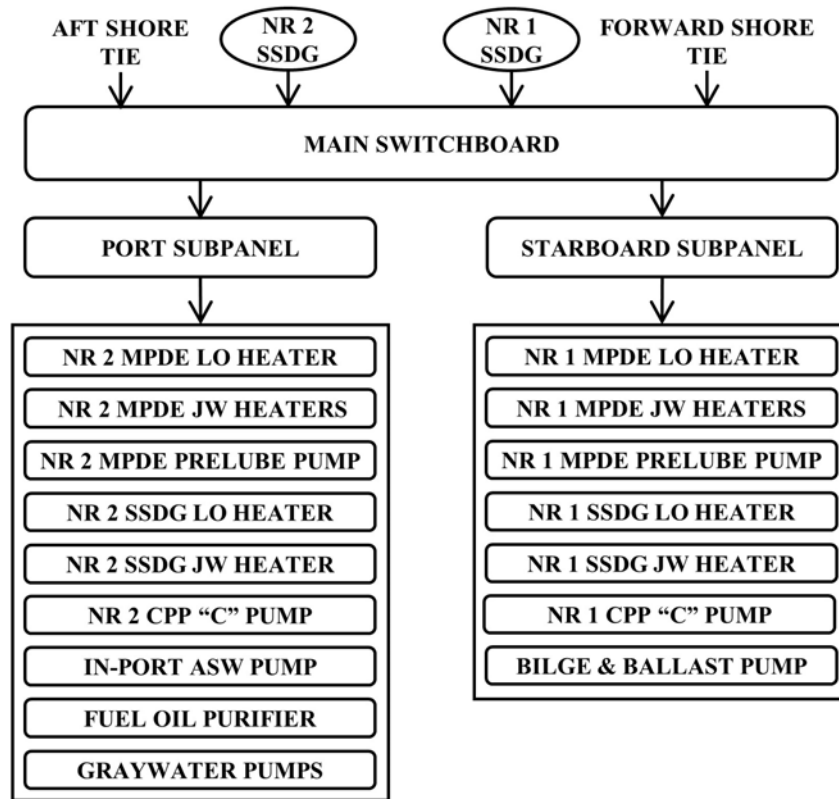


Figure A-2: Partial schematic of radial electrical distribution onboard USCGC Spencer, with details on the port and starboard subpanels [189].

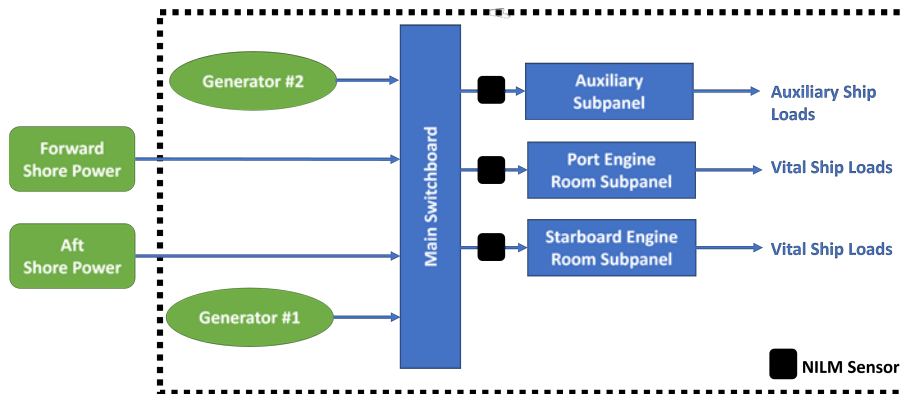


Figure A-3: NILM installation locations on USCGC Spencer.

is secured, the engine requires lubricating oil flow to ensure engine components are continuously lubricated and ready to operate. For this purpose, a 2.2 kW prelube pump operates in automatic mode to energize upon engine shutdown and turn off when the engine starts. Working in tandem with the prelube pump, a 12 kW lube oil

heater maintains the desired lubeoil viscosity and temperature (within 90° to 120° F). A 9 kW jacket water heater helps keep the engine warm during cold temperatures to ensure that the engine does not reach “cold iron” status, i.e., completely cold. If the engine has not been run for some time (e.g. after an in-port period), the MPDE goes through a “blow-down.” During a blow-down, the jacket water and lube oil heaters will turn-off and re-energize shortly after.

A.1.2 Ship Service Diesel Generator System

Ship power is provided by two generator sets. While at sea, power is provided by one or both of these ship service diesel generators (SSDGs), depending on the ship’s power demand. During time of high demand, both are used in parallel to ensure power delivery. The NILM monitors two of the SSDG support systems: the lube oil (LO) heater and jacket water (JW) heater. Similar to the MPDE LO heater, the SSDG LO heater is operated in automatic mode to ensure lubricating oil temperature is between set points. The SSDG LO heater is a 1.32 kW single-phase line-to-line load. An electric lubricating pump, which is not monitored by a NILM, is used to circulate oil when the engine is secured. The 7.5 kW, three-phase SSDG JW heater works similar to the MPDE JW heater, maintaining the temperature of the jacket water to ensure the generator is ready to start.

A.1.3 Fuel Oil Purification System

Before diesel fuel oil can be used by the MPDE or SSDG, it is purified and transferred by the fuel oil purifier (FOP). The FOP system takes in diesel fuel and outputs purified diesel, water, and sludge (a mixture of dirty fuel and dirty water). The flow is manually adjusted by an operator, depending on the purity of the fuel. The purifier system draws fuel from various storage tanks around the ship via a feed pump, and processes the fuel using a motor-driven centrifugal separator system. The NILM monitors the 5.6 kW centrifugal motor and 2.6 kW feed pump that makes up the FOP. The purifier is run frequently when the ship is underway.

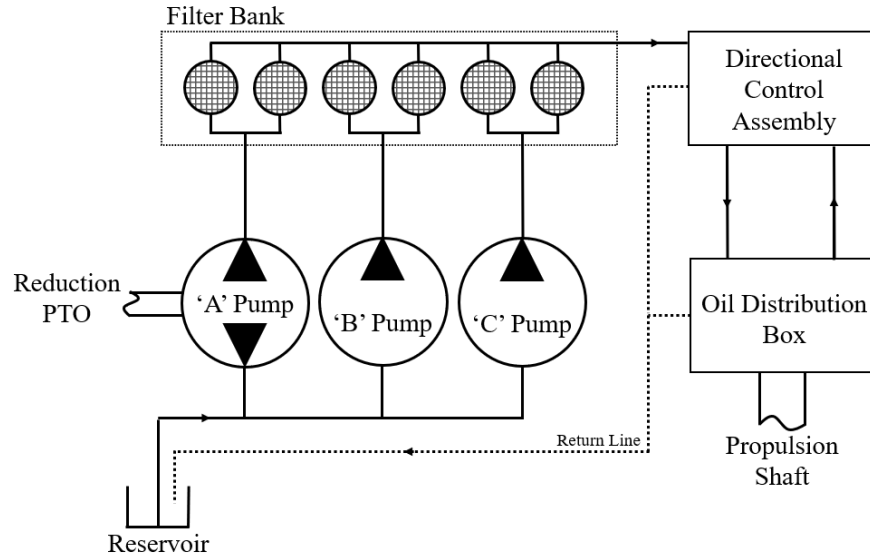


Figure A-4: CPP system conceptual diagram.

A.1.4 Controllable Pitch Propeller Pump

The controllable pitch propeller (CPP) system allows the operator to adjust the amount of thrust generated at a propulsor while maintaining a fixed rotational speed [6]. This provides the vessel greater maneuverability and prevents the underloading of diesel engines while operating at slow speeds. The CPP system consists of three primary hydraulic pumps that provide pressurized hydraulic oil to the CPP system in order to maintain hydraulic control pressure at the propeller. The ‘A’ pump is a gear driven pump that is powered by the propulsion system’s reduction gear. The pressure and flow provided by the ‘A’ pump is dependent on propeller shaft speed. The ‘B’ and ‘C’ pumps are electric hydraulic pumps that supplement the pressure and flow provided by the gear-driven pump. Hydraulic control valves maintain system operating pressure based on demand. A simplified line diagram of the system is shown in Fig. A-4. After leaving the pumps, hydraulic oil passes through a filter bank to remove contaminants. The cleansed oil then enters a directional control manifold that directs oil to the oil distribution (OD) box or returns it to the reservoir based on a user’s pitch command. Oil that enters the OD box is sent through ports within the propeller shaft to actuators within the hub that adjust blade pitch. NILM systems

monitor both the port and starboard CPP ‘C’ pumps, which are rated at 7.5 kW.

A.1.5 Graywater Pumps

The graywater pumps operate based on water-level sensors. The graywater pump is part of the graywater disposal system, designed to transfer, retain, process, and dispose of the relatively clean waste water from showers, sinks, and other appliances to a holding tank in the main engine room. When the holding tank is full, the graywater pump discharges the water from the holding tank either overboard or to a larger storage tank. There are two pumps (for redundancy) which alternate each cycle to empty the tank. Normally, a pump turns on and begins discharging when water reaches the “high level” sensor set point (92-gallon mark), and the pump turns off when water reaches the “low level” sensor set point (13-gallon mark). The graywater pumps are rated at 3.7 kW.

A.1.6 In-port Auxiliary Saltwater Pump

The in-port auxiliary saltwater pump is powered by a 7.5 kW three-phase motor. The pump is energized when the vessel is in port and turns off when the vessel departs for sea. The pump provides saltwater cooling for A/C and refrigeration units while the vessel is in port.

A.1.7 Bilge and Ballast Pump

The bilge and ballast pump is used for emptying machinery space bilges of excess water in an emergency and for taking on ballast water for stability purposes. The bilge and ballast pump is powered by a 5.6 kW motor. When pumping bilges and ballast tanks, operators try to get the tanks and bilges to the lowest level possible, and as a result, the pump takes in a mixture of air and water. After the pump is turned off and suction is shifted to a new tank, the air remains in the system, resulting in a prolonged start sequence in which the pump draws a variable amount of power.

A.1.8 Auxiliary Room Loads

The auxiliary room panel provides power to an air compressor, air conditioning unit compressor, hydraulic watertight door, and auxiliary seawater cooling pump. The ship's service air compressor provides medium pressure air (450 psi) to the cutter for various uses, including the pressurized air necessary for pneumatic pumps and tools, as well as start and control air for the engines in the main engineering plant. The system consists of a 7.5 kW electric motor that is belted to the air compressor. The auxiliary subpanel also feeds one of the air conditioning (A/C) unit compressors. The A/C compressor compresses low pressure gaseous refrigerant to a higher pressure for use in the HVAC loop. Between the main engine room and the auxiliary machinery room is a hydraulically actuated watertight door. This allows easy passage between the two spaces when in port, while providing watertight integrity between the two engineering spaces when the cutter is underway.

A.2 USS Indianapolis

In February 2020, the second prototype AIO box was installed on three power panels on USS Indianapolis, a Freedom-class littoral combat ship (LCS) in Mayport, Florida. This was the first field installation of the new AIO box. These installations demonstrated the capability of the new AIO box with minimal installation time. These NILMs, as shown in Fig. A-5, collected over a month of data. Several loads of interest are monitored on the three installed NILMs in the auxiliary machinery room, water jet machinery room, and machine shop. Specifically, the loads include the anchor windlass variable speed drive, No. 2 splitter gear lube oil and control oil pumps, No. 3 SSDG auxiliary loads, chill water pumps, and the hydraulic power pack cooling and booster pumps.

The microgrid on USS Indianapolis is a delta-configured 60 Hz, 450 V system. When at sea, there are four ship service diesel generators (SSDGs) providing power to the ship's microgrid. For propulsion, USS Indianapolis uses a combined diesel and gas (CODAG) turbine plant coupled with four water jets, as seen in the conceptual



Figure A-5: NILM installs on USS Indianapolis (LCS 17).

diagram in Fig. A-6. The CODAG system uses diesel engines for cruising speeds and gas turbines for high-speed transits. To achieve the required gear ratios and shaft speeds the gas turbines' and propulsion diesels' outputs are combined in a combining gear. This shaft is then subsequently split using a splitter gear to power two water jets. The SSDGs and propulsion system require electrical equipment both while operating and during standby in order to maintain operational status. The splitter gear requires a lubrication pump and a control pump for the lubrication and control oil, respectively. The splitter gear electric lube oil pump is energized at certain shaft speed ranges while USS Indianapolis is underway, with the lubrication oil being circulated by an attached pump when the shaft speed permits. The SSGs each have a skid attached to the free-end of the engine, which provides the auxiliaries required to run the generator. The power for the skid and for the generator control comes from the local operating panel. The auxiliary skid consists of two pumps and two heaters: one pump circulates lube oil and the other circulates jacket water throughout the SSDG.

A.2.1 Anchor Windlass VFD

Variable frequency drives (VFD) are often used to optimize and control energy consumption, speed and torque of induction motors. The anchor windlass on the machine shop subpanel is powered by a VFD. The VFD provides up to 111 kW to control anchor windlass speed while raising and lowering the anchor or adjusting and working

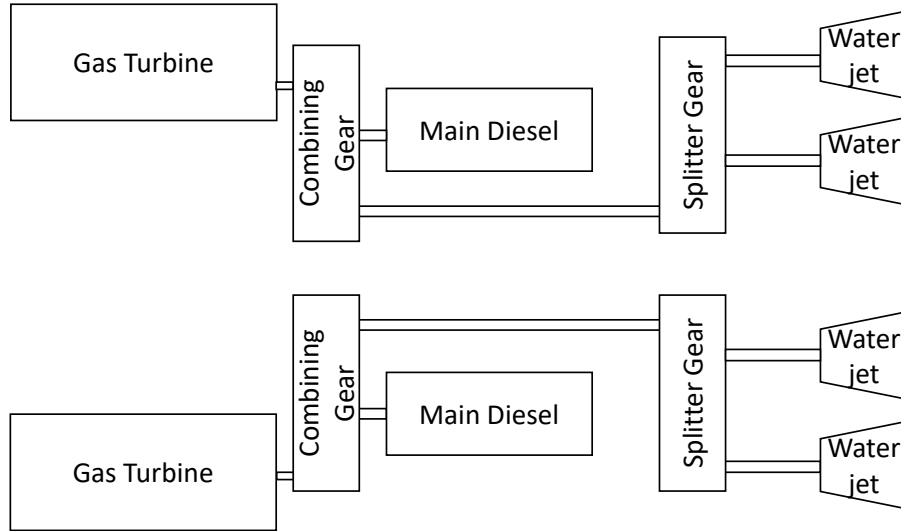


Figure A-6: USS Indianapolis propulsion schematic.

mooring lines.

A.2.2 Splitter Gear

The splitter gear for the propulsion system, as seen in the conceptual diagram in Fig. A-6, contains both a lubrication oil pump and control oil pump. The splitter gear electric lube oil (LO) pump, which is monitored by a NILM, is energized at certain shaft speed ranges while the USS Indianapolis is underway, with the lubrication oil being circulated by an attached gear-driven pump when the shaft speed permits. The splitter gear is used to divide the output of the combining gear to power two separate water jets. To clutch in and out this gear uses high pressure oil, which is supplied and pressurized by an 18.6 kW electric control oil pump and an attached gear-driven pump (a pump that is hard mounted into the gear and operates based on shaft speed). The control oil provides the splitter gear pressure to clutch in, which must be maintained above a specific threshold for the gears to properly mesh. At a “high” propulsion shaft speed the attached pump provides all the required pressure. The electric pump secures when the attached pump provides sufficient pressure for clutch operation, which corresponds to a specific propulsion shaft speed. However, at lower shaft speeds, both the electric and attached pump are necessary to provide

the required pressure. If the attached pump is providing a significant portion of the load and the shaft speed drops due to mechanical interference (e.g., high speed, wave action, weather, maneuvering, etc.) the electric pump will pick up the load to ensure the gear remains clutched in. The electric pump is required to “surge” to compensate for the attached pump.

A.2.3 Ship’s Service Diesel Generator No. 3 Auxiliary Skid

USS Indianapolis is equipped with four ship service diesel generators (SSDGs). These generators have a skid attached to the free-end of the engine, which provides the auxiliaries required to run the generator. The power for the skid and for the generator control comes from the Local Operating Panel (LOP), which is fed by one of the monitored panels. The auxiliary skid includes two pumps and two heaters: one pump circulates lube oil and the other circulates jacket water throughout the SSDG. Both heaters actuate simultaneously, and are rated to draw a combined 12 kW. Furthermore, the SSDG jacket water pump, which is rated at 1.12 kW, runs simultaneously with the heating elements. The SSDG lube oil (LO) pump is rated at 1.5 kW.

A.2.4 Chill Water Pumps No. 2 and No. 3

The chill water system onboard USS Indianapolis is responsible for providing chilled water to all electronic equipment and for providing air handlers with cooling water for air conditioning. There are three chill water pumps onboard, two of which (No. 2 and No. 3) were monitored by the NILM system, with a minimum of one running at all times. The pumps are rated at 22.3 kW, operate at 3535 rpm, and are remotely operated by watchstanders via the machinery control system (MCS).

A.2.5 Hydraulic Power Pack Cooling and Booster Pumps

The propulsion system onboard USS Indianapolis utilizes four water jets, two of which are hydraulically controlled. This control system enables the ship to maneuver by directing the thrust vector of two of the water jets, eliminating the need for a



Figure A-7: NILM installations on (a) USCGC Marlin and (b) USCGC Thunder Bay.

traditional rudder system. This hydraulic system consists of a hydraulic power pack (HPP) cooling pump and booster pump. The port-side system’s cooling pump and booster pump are both powered via the monitored subpanel in the water jet machinery room. The HPP cooling pump is rated at 2.2 kW. The booster pump for the water jet’s hydraulic control system enables the movement of the large water jets. This pump pressurizes the hydraulic fluid which is then used to position the water jets. This pump is rated for 55.9 kW.

A.3 USCGC Marlin

In August 2020, two AIO NILMs were installed on USCGC Marlin, a 87-ft. patrol boat homeported in Boston, Massachusetts. This Marine Protector class cutter is typically underway for several days at a time off of New England conducting search and rescue, fisheries regulation, and law enforcement. The monitored subpanels, port and starboard, effectively monitors the entire ship with the exception of the steering pumps. The NILMs installed on USCGC Marlin are shown in Fig. A-7a. Fig. A-8 shows the high-level overview of monitored loads on USCGC Marlin. There are also several other loads further downstream on various subpanels. Here, an overview of several loads of interest that are relevant in this thesis are presented.

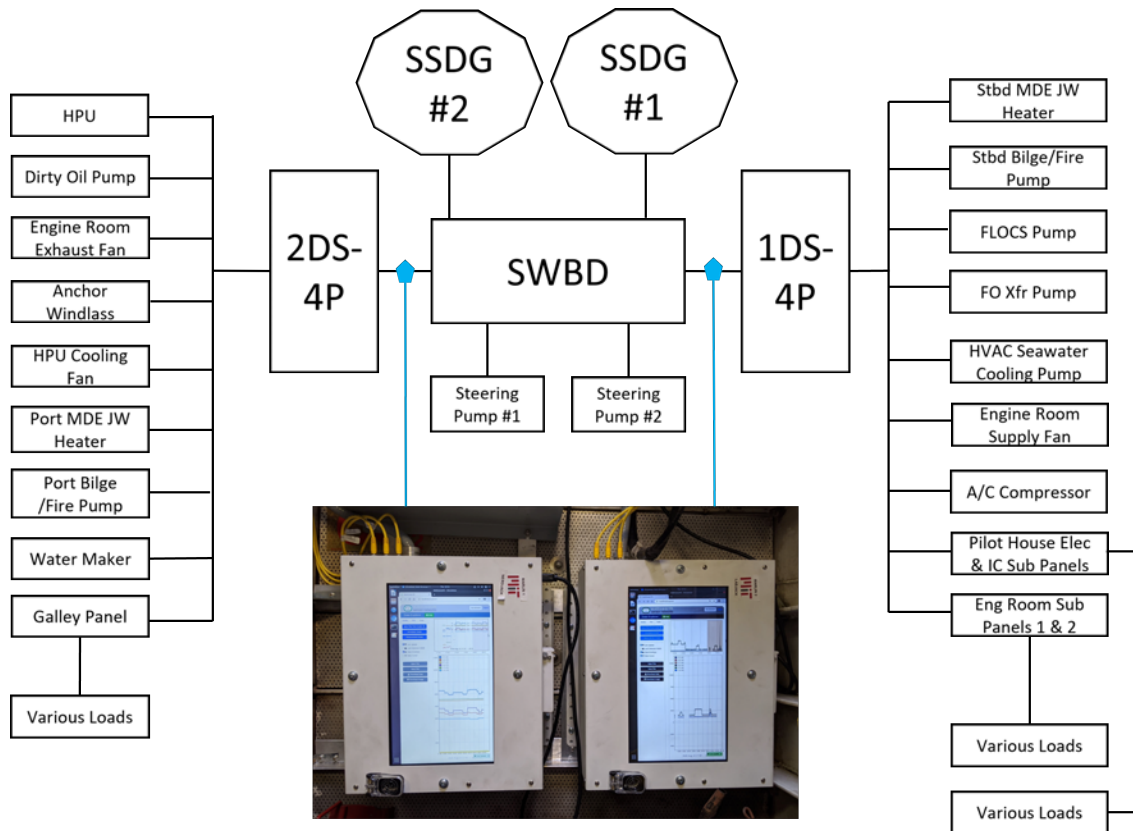


Figure A-8: USCGC Marlin loads.

A.3.1 Engine Room Exhaust and Supply Fans

USCGC Marlin’s engine room is ventilated by two fans, one supply and one exhaust fan. Both are controlled by variable frequency drives (VFDs), enabling the crew to determine the supply and exhaust levels. Under typical operation these fans are running at over 50%.

A.3.2 Heating Ventilation Air Conditioner

The heating ventilation air conditioner (HVAC) controller is a power electronic control system that has various settings depending on the ambient conditions. It incorporates the ship’s A/C plant, space heaters, and fan ventilation to produce desired livable environments inside the habitable compartments. The primary system is segregated into supply and exhaust; with supply drawing air from the weatherdeck and internally for re-circulation. The supply side is driven by a single centrifugal fan, and supplies

air to all habitable spaces. The exhaust system is similarly designed, however it draws from most compartments and utilizes a much smaller centrifugal fan, moving stagnant internal air outside. The air is filtered prior to the fan and is subjected to a multi-stage heating process. Due to the nature of the HVAC system, the system is always running under normal ship operation and is shutdown only for emergency, maintenance, or in the case of power loss.

A.3.3 Steering Hydraulic Power Unit Cooling Fan

The hydraulic power unit (HPU) cooling fan is a tube axial 3-phase fan mounted in the aft portion of the engine compartment. The flow is oriented to cool the steering HPU under longer periods of operation. Additionally, the fan serves as re-circulation for the engine compartment. The HPU cooling fan is rated at 0.37 kW.

A.4 USCGC Thunder Bay

In February 2021, two AIO NILMs, as shown in Fig. A-7b, were installed on USCGC Thunder Bay, a 140-ft icebreaking tug homeported in Rockland, Maine. USCGC Thunder Bay's primary mission involves breaking ice within navigable New England waterways during the winter months. Breaking ice can place large amounts of stress on a ship's hull and various subsystems, and being able to accurately monitor these systems for faults and failures is critical to maintaining operational capability. USCGC Thunder Bay is one of few Coast Guard assets powered by a diesel electric propulsion plant, with two main propulsion diesel engines (MPDE) driving two main propulsion generators (MPG), which in turn provide power to the main propulsion motor that drives the ship's single propulsion shaft. There are also two separate ship service diesel generators (SSDGs) that are used to power the remainder of the ship's loads. Icebreakers and other vessels that have a large variation in operation profile were among the first to adopt electric propulsion due to the significant torque and constant load variations while traveling at low speeds [190]. A conceptual diagram of the layout of the ship's propulsion system is shown in Fig. A-9.

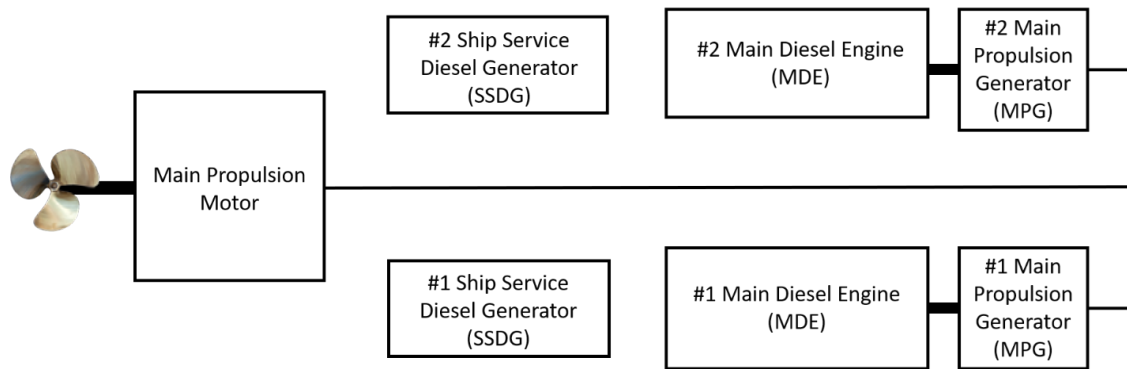


Figure A-9: Conceptual diagram of USCGC Thunder Bay propulsion system.

A NILM system is installed on panel 1-66-2, located in the aft section of the engine control center (ECC). The monitored subpanel provides power to loads primarily related to the electric propulsion machinery. The two dc propulsion generators are each rated at 1,000 kW, 900 V, and 850 rpm. There are three propulsion generator exciters (one exciter for each propulsion generator, plus an additional standby exciter), which provide the source of main shunt field excitation for the generators. The exciters use 450 Vac, 3-phase, 60 Hz input power, as supplied from the ship service switchboard. The ac input power is converted to a variable controlled maximum output of 360 Vdc and 80 A. The propulsion motor is a separately-excited dc motor rated at 2,500 hp. For motor control, there are two exciters: a main propulsion motor field exciter and a standby propulsion motor field exciter. Each of the discrete throttle positions corresponds to a commanded motor field current, diesel engine speed, and dc propulsion generator and motor armature voltage. The control uses these three commands to establish the motor field excitation, diesel engine speed, and generator field excitation, respectively. For throttle inputs between 0 and 90%, the motor field current is commanded to a constant 48.3 A and 39.0 A for one-engine and two-engine mode, respectively. Only at the maximum throttle position of 100% does the motor field current command increase. That is, except for the throttle position at 100%, changes to the throttle position only changes the commanded diesel speed and generator field excitation. The NILM is monitoring both the generator and motor field exciters. Changes in throttle position are visible in the power stream. Other

loads on the panel include the main motor cooling pump NR 1 and main motor cooling pump NR 2.

The other NILM system is installed on panel 2-32-1, located in the forward section of the engine room. The loads on the engine room panel are primarily related to the two MPDEs and two SSDGs. The monitored loads related to the MPDEs are the crankcase blowers, pre-lube pumps, lubeoil heaters, lubeoil circulation pumps, jacketwater heaters, and jacketwater circulation pumps. Other monitored loads include the SSDG jacketwater heaters, engine room supply fans, and a gray water pump.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Jacket Water Heater Fault Analysis

This appendix provides root-cause analysis for the observed main propulsion diesel engine (MPDE) jacket water (JW) heater faults on USCGCs Spencer and Escanaba. The electrical analysis presented in Section 6.2 showed that a NILM can detect failures related to the MPDE JW heater, such as open-circuits, arcing, and high-impedance ground faults. The electrical signatures are effective in identifying degraded heater operation, but cannot fully explain what is causing the faults. After detection of degraded power signatures, inspection of the heaters revealed significant corrosion and damage, including arcing, melting, and cracking of the heating elements. Material and vibration analysis of the observed field failures are presented to demonstrate the connection between observed electrical failure signatures and root-cause failure and degradation mechanisms. The root-cause analysis presented here was in collaboration with Devin Quinn, Samuel Madden, and Peter Lindahl.

Industrial processes rely on materials that are susceptible to corrosion and mechanical failure [140, 191, 192]. For instance, heat exchangers in industrial facilities often exploit the high thermal conductivity of copper. In high-stress environments, copper structures are susceptible to corrosion and cracking. The copper sheathing of a heating element can degrade to the point of failure and expose the electrically active heating elements. Prior to complete system failure, degradation can lead to arcing and water intrusion. Early detection can prevent service delays and hazards like electrical fires. However, the detection and evaluation of corrosion in critical structures

remains a significant maintenance and repair challenge [193]. Electrical monitoring can provide noninvasive means to spot these failures before they become disasters.

The typical construction of large marine diesel engines incorporates a jacket water (JW) system to provide engine cooling while operating. The JW acts as a heat absorbing medium that removes heat from engine components and releases it to the environment through a heat exchanger. Thermostatic valves often regulate the flow of JW through the heat exchanger based on engine temperature. While the engine is not running, the JW system also plays a significant role in maintaining uniform engine temperature by circulating water through resistive heaters. This minimizes time to reach standard operating temperatures upon start up. The USCG 270-ft medium endurance cutter fleet uses copper-sheathed heaters in the main propulsion diesel engine (MPDE) JW system [27].

Per manufacturer recommendations, the USCG conducts semi-annual visual and electrical state examination of the three-phase JW heaters and control systems used on the MPDE. Identification of degraded transient signatures enables the use of non-intrusive electrical monitoring for detecting incipient failures and permitting condition-based, instead of scheduled maintenance [194].

B.1 Material Analysis

To assist with failure root-cause evaluation a material analysis of failed JW heaters was conducted. The post-mortem analysis of failed heating elements revealed significant corrosion, both general and pitting, as well as possible stress corrosion cracking (SCC) of the copper sheathing. Computed tomography (CT) of a heater showed significant material loss in addition to through-wall cracking as shown in Fig. B-1. The area highlighted in the CT analysis was on a sheath section that did not appear to have been directly involved in a shorting event. Neighboring sections of sheathing experienced significant melting and material loss. These areas were avoided in the CT analysis due to the expected material damage and the desire to observe the condition of the copper prior to an arcing event. The analyzed area would have been thermally

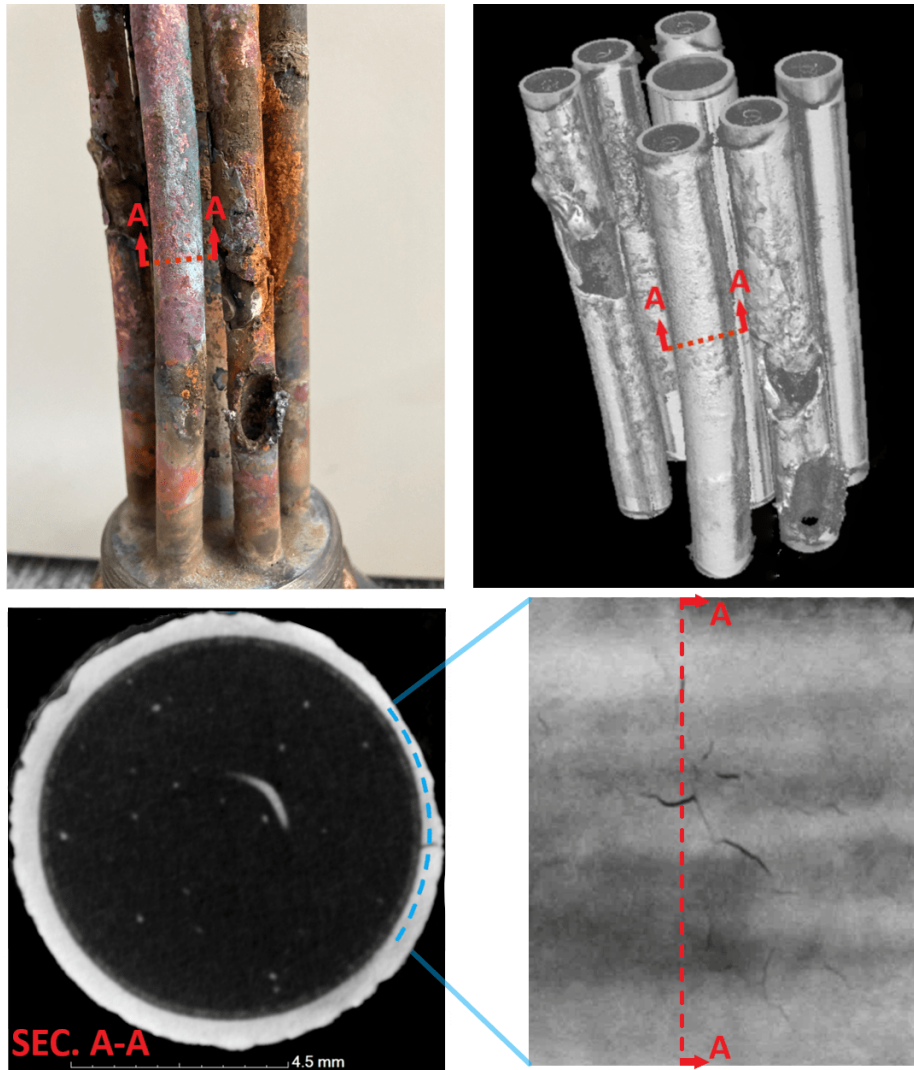


Figure B-1: Top Left - degraded JW heater from USCGC Escanaba. Top Right - CT scan of heater, 3D volume reconstruction, high density materials appear bright in contrast. Lower Left - virtual cross section showing through wall crack at 3 o'clock. Lower Right - virtual unrolling of sheathing showing branched network of cracking. Dashed lines - plane of cross section and unrolling.

isolated from the short by the JW it was submersed in.

The copper heating element sheath is electrically isolated from the iron chrome aluminum (FeCrAl) heating wire by magnesium oxide (MgO) powder. A breach in the sheathing by through-wall pitting and cracking can cause infiltration of the insulating MgO powder with JW. One contributing factor in the cracking of the copper sheathing may actually have been in the selection of corrosion inhibitor for the JW. An additive rich in nitrites was used to inhibit corrosion, with concentrations

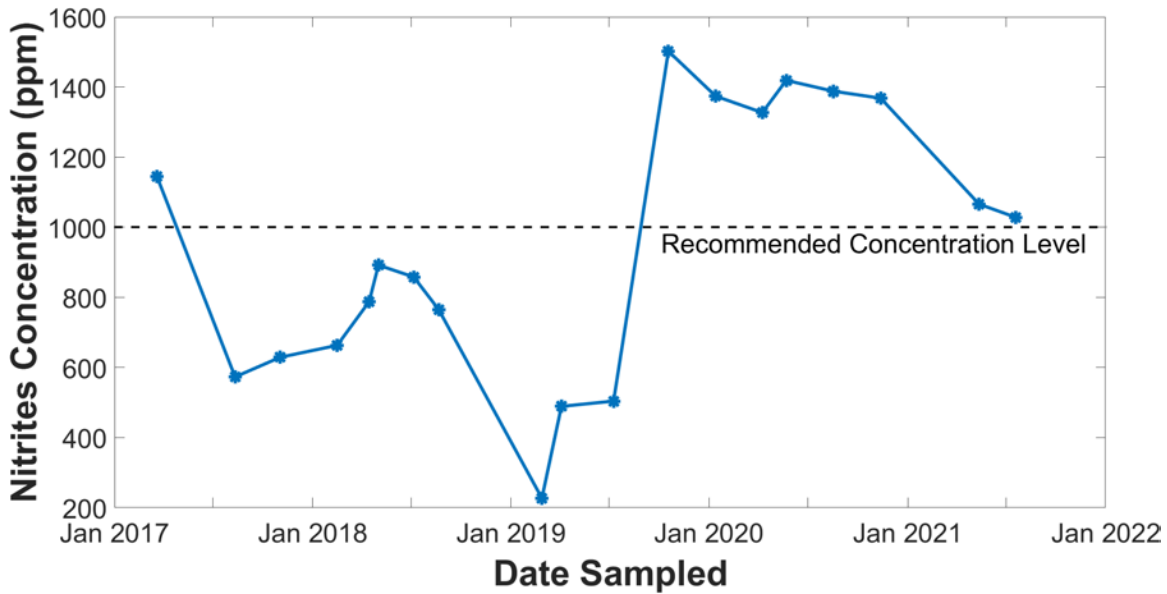
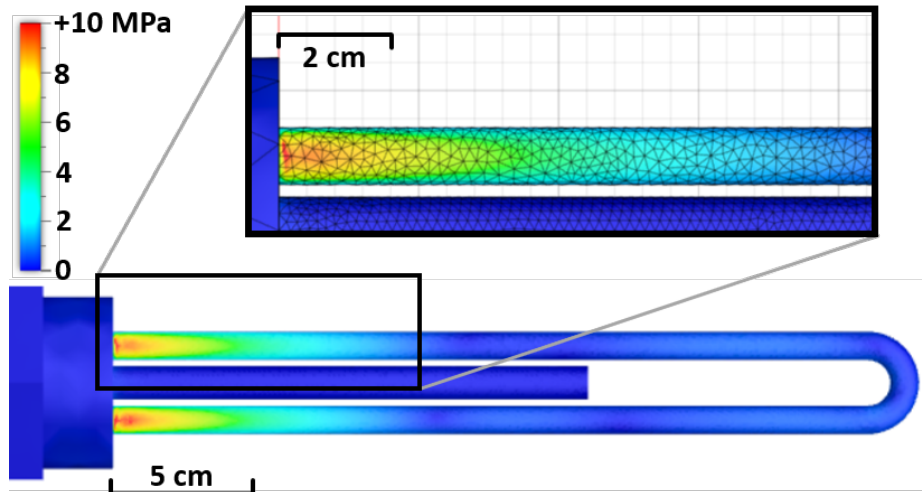


Figure B-2: ESCANABA JW chemistry from March 2017 to May 2021.

hovering around 1000 ppm, as shown in Fig. B-2. Scientific literature demonstrates that copper exposed to nitrites may induce SCC [195,196]. The loss of wall thickness and the presence of large pits shown in Fig. B-1 suggests that cracking is one possible route for JW to enter the insulator. In our analysis the cracked regions were the only areas, outside of the obvious arc locations, in which through-wall perforations in the sheathing were observed.

To further evaluate the possibility of SCC, a model of the JW heater was generated and assessed using Autodesk’s finite element event simulator. This analysis assumed ideal geometry of a new heater and used an adaptive triangular mesh. Accelerometer measurements of a JW heater while installed on a running MPDE were used to develop an impulse event for the analysis. This event was sinusoidal in nature, cycled at 1640 Hz, and reached a max amplitude of 44.3 N. For the simulation this impulse was applied perpendicular to the element’s arched plane at its center of mass. To emulate the submerged condition of an installed heater, a hydro-static pressure equivalent to 3m of water was placed over the heating elements. Simulation results at the moment of peak impulse are shown in Fig. B-3a. The simulation found that induced stresses from engine excitation are greatest at the base of the heating element and diminish along the element’s length. A maximum induced stress of 11.8 MPa was



(a) Stress simulation results, color gradient in MPa.



(b) Holes observed at approximately 1 cm and 5 cm from the heater's base.

Figure B-3: Finite element stress simulation and observed damage.

observed approximately 1 mm from the heater's base. Table B.1 contains induced stress magnitudes and percentage of maximum stress at various distances from the heater's base. These findings align with observed holes that typically occurred within 5 cm of the heater's base, as shown for two example heaters in Fig. B-3b.

As mentioned, through-wall pitting and cracking of the sheathing may allow JW into the insulating MgO powder. The result of JW ingress into the insulator was

Table B.1: Induced stress magnitudes - CAD model

Dist. From Base (cm)	Induced Stress (MPa)	% of Max Stress
0.1	11.8	100%
1	9.26	78.8%
5	4.40	37.4%
10	0.563	4.79%
20	0.190	1.62%

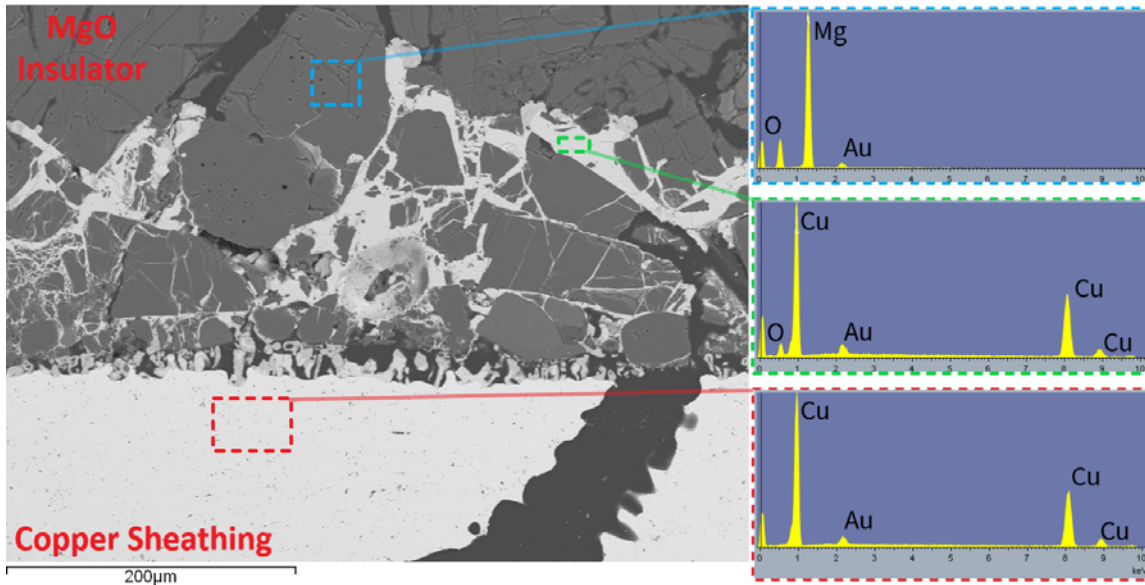


Figure B-4: SEM/EDS of the interior surface of the sheathing showing preferential growth of a copper rich phase (green/middle box) towards the energized heating element.

observed in scanning electron microscopy (SEM) and energy-dispersive X-ray spectroscopy (EDS) which is shown in Fig. B-4. The dark void through the copper sheathing in Fig. B-4 is the crack from Fig. B-1. Preferential deposition of copper through the insulating MgO powder towards the energized heating element was observed. Continued growth of the copper phase through the insulator could explain the shorting experienced and the significant arc damage observed in the failed heating elements. The development of SCC can be partially attributed to continuous exposure to high concentrations of nitrites, but excitation at the heater's natural frequency may induce resonance that could accelerate the process. To evaluate this potential the heater's natural frequency was determined, as detailed in the following section.

B.2 Natural Frequency Determination

To further investigate the possibility of stress corrosion cracking (SCC) found during the material analysis, the heater's natural frequency was estimated and compared to the MPDE's operating speed. The JW heater consists of a housing and three heating elements. The housing threads directly into the MPDE block. The heating elements resemble a set of cantilever beams or prongs of a tuning fork and react similarly to disturbances or impulses. The natural frequency of the JW heater was predicted with the measurement of physical parameters and analytic equations, then was verified experimentally.

First, physical parameters of the JW heater were measured using a deflection experiment. The deflection experiment was conducted on a single heating element with known weights and a dial indicator. The shape of the heating elements complicates separate calculation of the second moment of area, I , and Young's modulus of elasticity, E . Instead, the product EI was determined with,

$$EI = \frac{Fl^3}{3\delta}, \quad (\text{B.1})$$

where F is the applied force, l is the length, and δ is the deflection [197] [198]. This is the deflection equation for a cantilever beam (boundary conditions fixed-free) with a point load applied at the free end. Point loads ranging from 1.7 to 4.5 N were applied 26 cm from the fixed end of the heating element. This resulted in an average EI value of 62.2 N·m². It was ensured that loading of the heating element remained within the elastic region of the component's stress-strain curve by verifying it returned to its neutral position after removing the applied load. Deflection from the heating element's own weight was deemed negligible, as it was calculated to be less than 1% of the total deflection.

For determining the natural frequency of the element, its own mass must be considered. Based on CT scanning and EDS performed during the material analysis, the heating element's mass per unit length was determined to be 0.33 kg/m and was confirmed by cutting and weighing a small section of the heating element. The

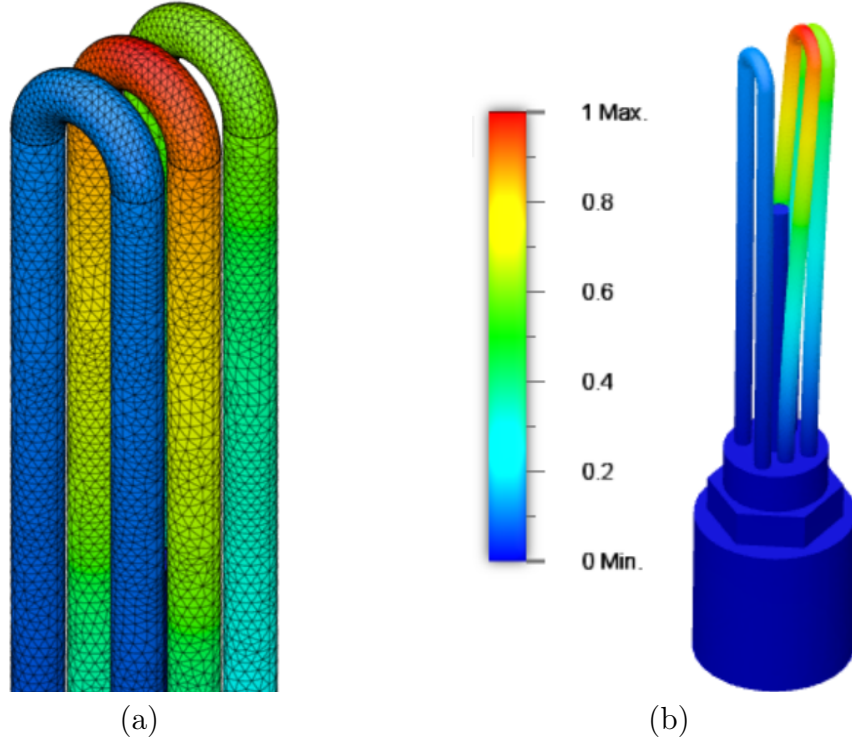


Figure B-5: Natural frequency modelling using finite element analysis: (a) finite element mesh, (b) natural frequency simulation, color gradient indicating relative displacement.

natural frequency, f , of a heating element was calculated with,

$$f = 0.56 \sqrt{\frac{EI}{ql^4}}, \quad (\text{B.2})$$

where q is the mass per unit length [197]. This provided a predicted first order natural frequency of 93.6 Hz for a single heating element. Due to the rigidity of the housing, mounting configuration, and overall slenderness of individual elements, higher order modulations and interactions between elements were not considered. To further evaluate these considerations, a CAD model was generated and assessed using Autodesk's finite element modal frequency analysis. This analysis assumed ideal geometry of a new heater. A scaled triangular mesh based on 10% of component size with more than 175,000 elements was used and is shown in Fig. B-5a. Results of the analysis indicated a primary natural frequency at 96.8 Hz with the center heating

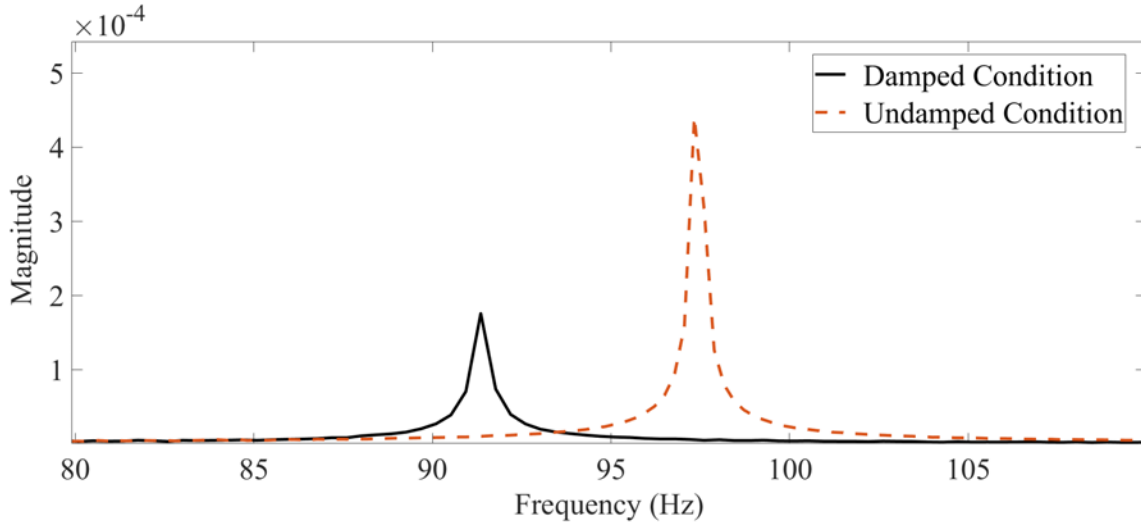


Figure B-6: DFTs to determine JW heater natural frequency in damped and undamped conditions.

element oscillating in the same plane as the outer two, as shown in Fig. B-5b.

To verify the predicted natural frequency values generated from the physical parameter calculations and the finite element analysis, an impulse impact experiment was conducted. The experiment was conducted for two scenarios, an undamped case, i.e., an unsubmerged heater, and a damped case, i.e., a heater submerged in water. The damped case simulates the actual operating condition of the heater. The vibration of the heater was measured using a single axis accelerometer affixed to the side of the heater housing and data acquisition device discussed in reference [199]. The accelerometer used has a sensitivity of 505 mV/g and output range of 0 to 5 V. An impulse impact was used to excite the system [200], and the DFT of the responses are shown in Fig. B-6. For the undamped system the response shows a peak at 97.3 Hz. This result closely matches predicted values and verifies the deflection experiment and analytic formulation. For the damped system the response shows a smaller, but still prominent peak at 91.3 Hz. The natural frequency as predicted by the deflection experiment, CAD model, and impulse impact tests are summarized in Table B.2.

The impulse impact method was also used on JW heaters with visible signs of deterioration. The DFT of the responses are shown in Fig. B-7. For the corroded heater without holes, the primary response shows a peak at 90.9 Hz and 96.8 Hz in

Table B.2: New heater natural frequency

Method	Frequency
Deflection Experiment	93.6 Hz
CAD Model	96.8 Hz
Hammer Impact (Undamped)	97.3 Hz
Hammer Impact (Damped)	91.3 Hz

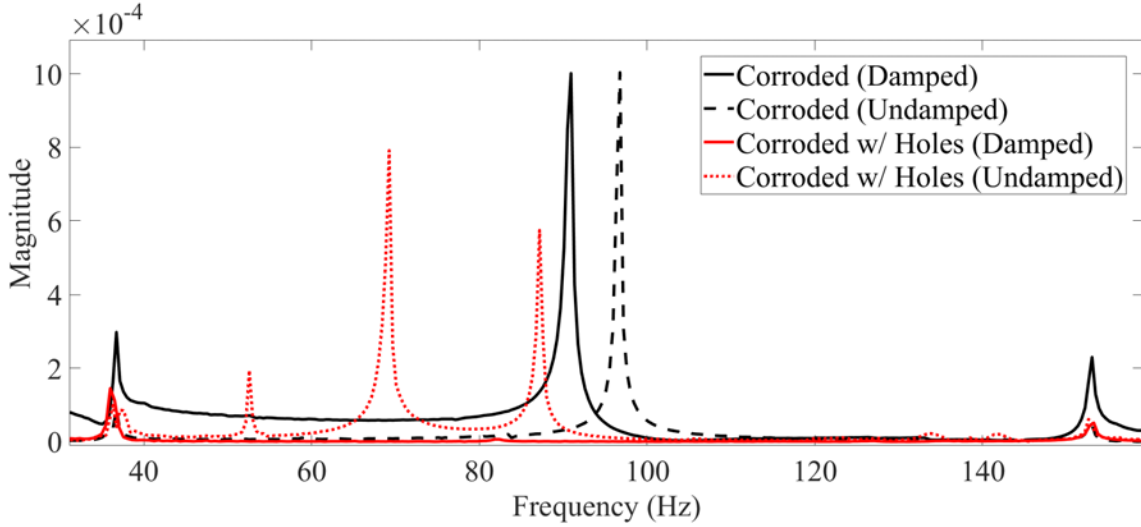


Figure B-7: DFTs to determine corroded JW heater natural frequency in damped and undamped conditions.

the damped and undamped conditions, respectively. These closely resemble the peaks of the new heater. However, smaller peaks are also present around 36 Hz and 153 Hz. For the corroded heater with holes there is a more prominent shift in DFT peaks. In the damped condition there is no longer a primary peak around 90 to 98 Hz. Instead, there are small peaks at 36 Hz and 153 Hz. In the undamped condition there are substantial peaks at 87.2 Hz and 69.3 Hz, with minor peaks around 36 Hz, 53 Hz, and 153 Hz. These varying results are anticipated because any imperfections in the heating element caused by corrosion or heating will impact the structure's natural frequency.

Field vibration measurements were gathered while the MPDE was idling at 440 rpm. These measurements were used to investigate possible excitation of the heaters during operation. An accelerometer was attached to the heater housing as shown in Fig. B-8, in the same manner as in the laboratory hammer impact experiment. Mea-

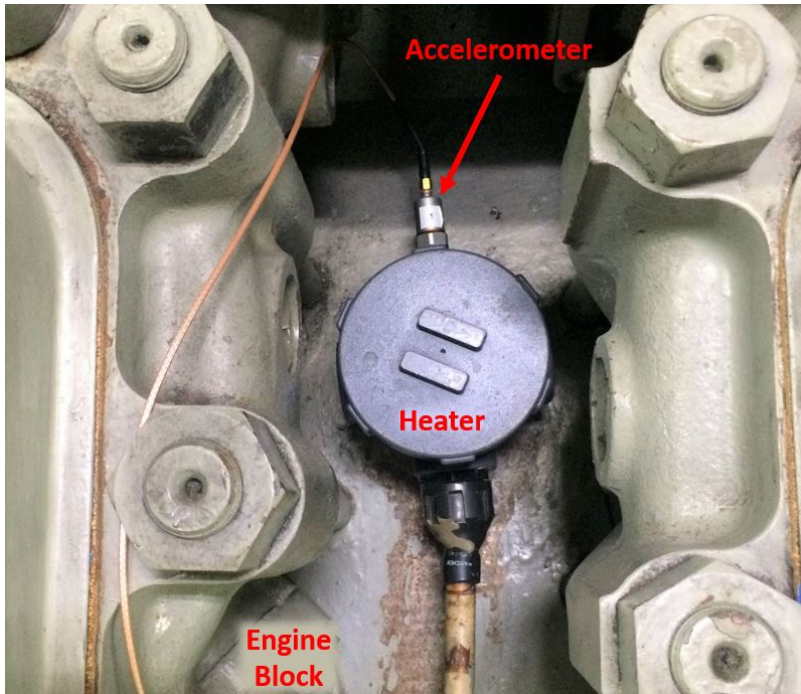


Figure B-8: Configuration for monitoring vibration of installed JW heater.

Measurements were also taken with the accelerometer attached directly to the MPDE block. Both measurements present a vibration spectrum that is rich in content due to phenomena such as cylinder firing, shaft rotation, possible imbalances, misalignment, and other non-idealities associated with mechanical systems. However, the spectrum is dominated by the fundamental engine order of 3.67 Hz and its subsequent harmonics. A peak was also observed at 97 Hz in measurements from both the JW heater and MPDE block. Based on vibration measurements gathered directly from the engine's turbo, this peak was determined to be a harmonic of the turbo's rotational speed. The DFT of a JW heater measurement is shown in Fig. B-9. It is suspected that the impulses imparted by the MPDE onto the JW heater near the calculated natural frequency values presented previously [see Table B.2], are likely to trigger heater resonance excitation. For a healthy heater, with undamaged copper sheathing, resonant excitation due to engine vibration is unlikely to cause failure. However, if the copper sheathing is weakened due to corrosion or a manufacturer defect, resonant excitation could exacerbate structural pathologies and induce stress corrosion cracking.

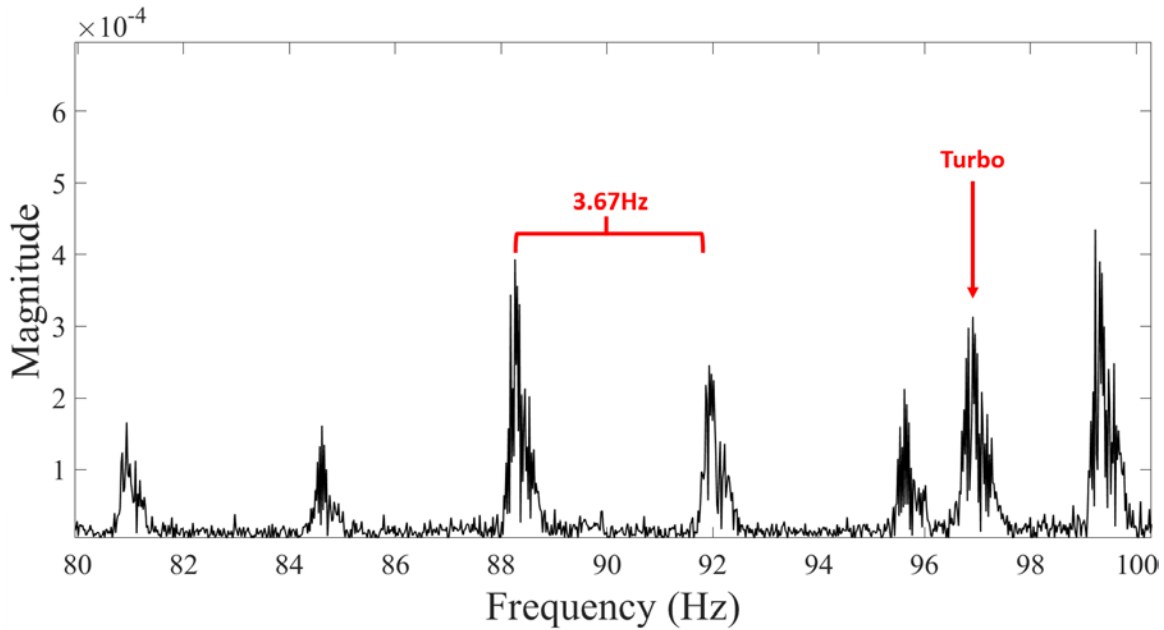


Figure B-9: DFT of monitored vibration of JW heater with MPDE idling.

B.3 Conclusion

Electrical analysis with nonintrusive load monitoring unveiled degraded MPDE JW heater power signatures, arcing, and high impedance ground faults. Diagnostic signatures developed for electrical monitoring can be used to detect material degradation and structure failure. Subsequent removal and inspection of the heaters in our field tests revealed corrosion, arcing, melting, and cracking of the heating elements. Monitoring the JW heaters for corrosion and related damage is critical for ensuring the efficient and safe operation of the JW keep-warm system. Power monitoring can alert watchstanders to corroded heating elements and prevent progressively worsening safety hazards. It can also be used to streamline maintenance activities, by using power stream data as a metric for conducting condition based maintenance, rather than recurring preventative maintenance.

Appendix C

NILM Software Documentation

This manual will guide a user through the process of setting up and using the NILM software stack. This starts with building a bootable NILM Ubuntu disk image (ISO) and putting the ISO on a USB drive. Then, the custom NILM Ubuntu operating system and software can be installed on a computer. Once installed, the configurations need to be set up for data capture. For backup, data can be copied to an external hard drive. A demonstration can be set up and displayed with NILM Dashboard. This documentation was co-written with Aaron Langham.

C.1 Customizing Installable NILM Ubuntu ISO

To create a NILM Ubuntu ISO file, the software Cubic can be used, which provides a graphical user interface (GUI) for customizing Ubuntu ISOs. From an Ubuntu system, Cubic can be installed via terminal as follows:

Command Line:

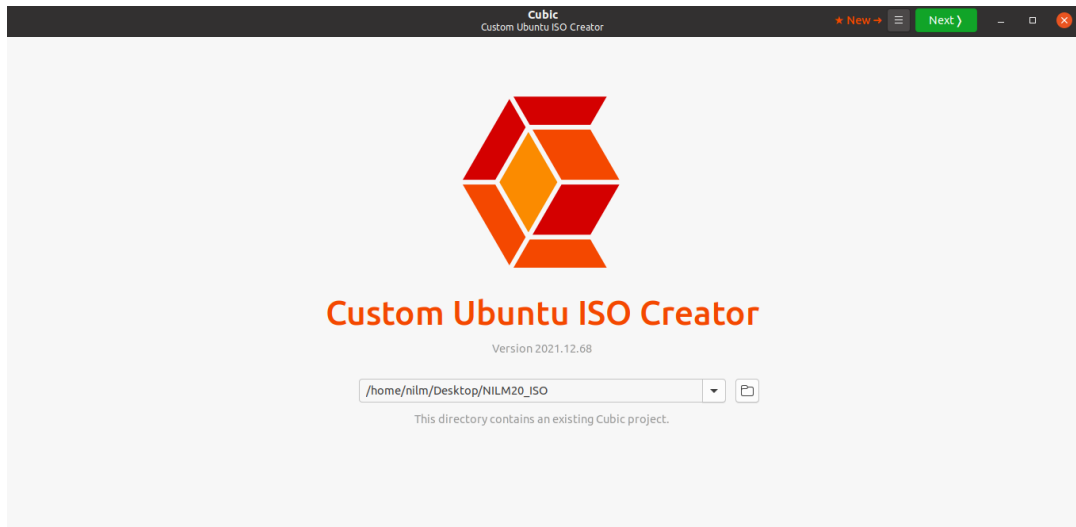
```
$> sudo add-apt-repository ppa:cubic-wizard/release
$> sudo apt update
$> sudo apt install cubic
```

We will be adding customization to the Wattsworth/Joule/NILM ISO provided by Prof. John Donnal. The additional capabilities include NILM Dashboard, NILM

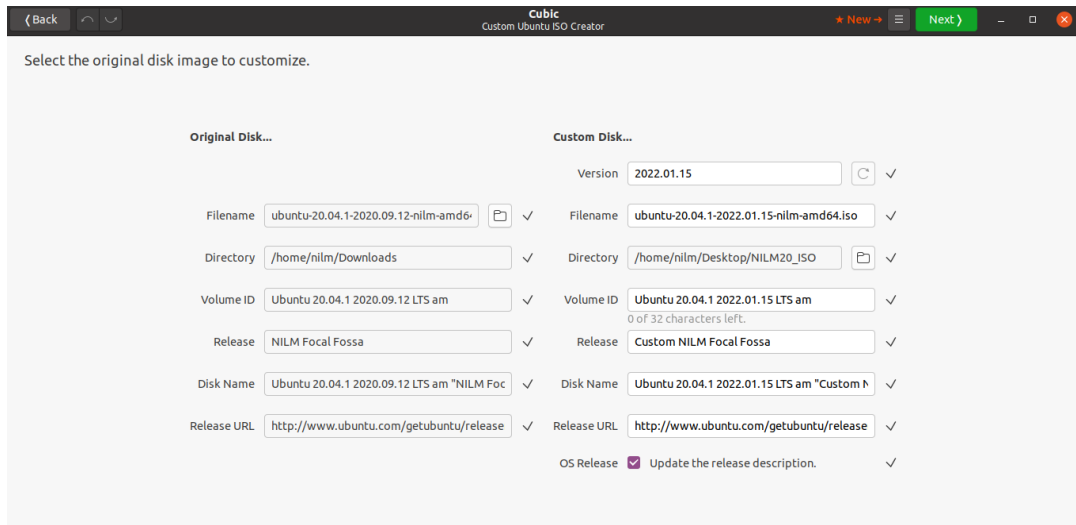
configuration and setup files, and cron jobs for checking the status of data capture and storage.

Once Cubic is installed the custom NILM Ubuntu ISO is created with the following steps:

1. Launch Cubic and select a directory for the project. Click Next.

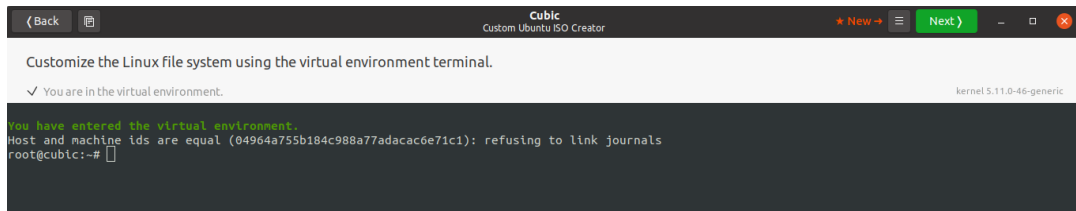


2. Select the Ubuntu ISO to be customized. Click Next.

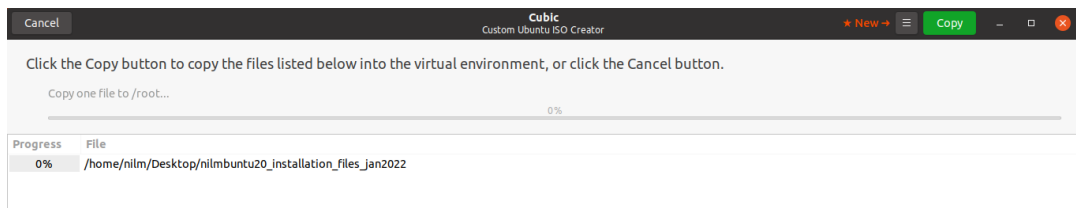


3. After the original ISO has been opened, an embedded terminal window will be displayed in either a chroot environment or virtual environment. Drag and drop

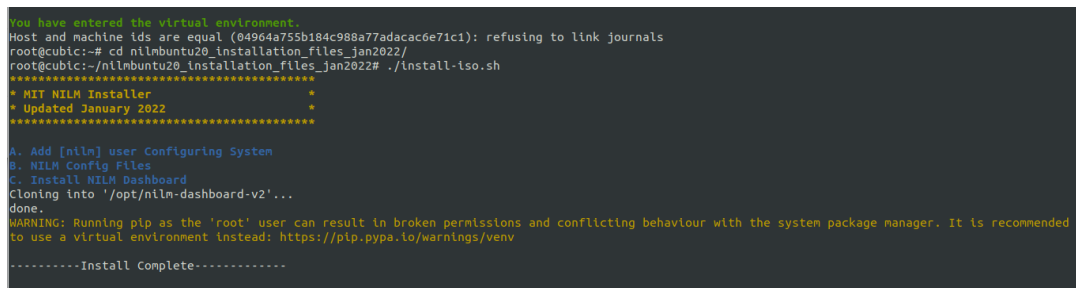
the folder with the installation files (e.g., “nilmbuntu20_installation_files_jan2022”). The `install_iso.sh` and `install.sh` files are in Appendix C.9.



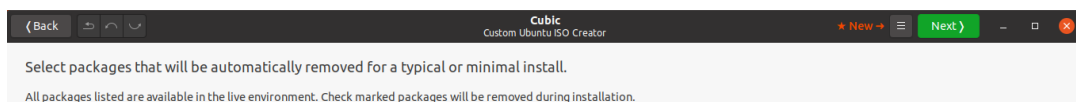
4. Click the Copy button to copy the files.

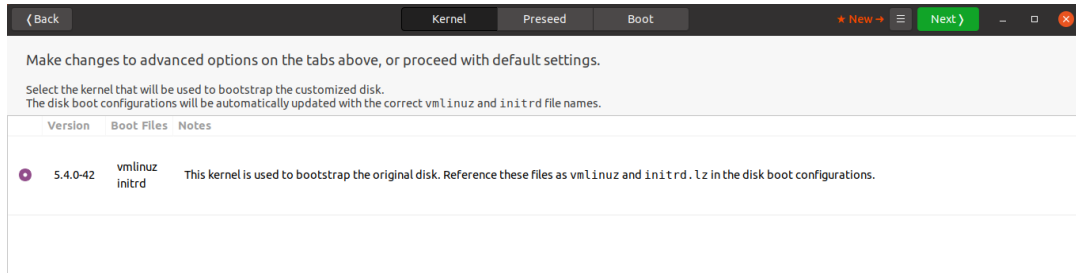


5. Within the Cubic virtual environment, run the following commands, where `cd` should point to the installation folder copied in the previous step. This should finish without error and say “Install Complete.”

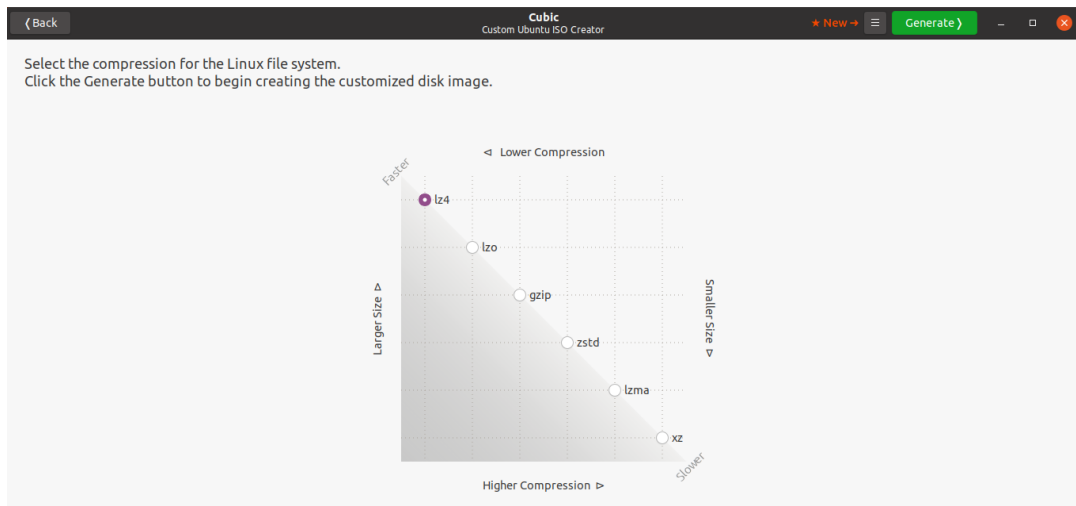


6. There should now be options for changing the packages and kernel. Click the Next button, possibly several times (no changes need to be made).

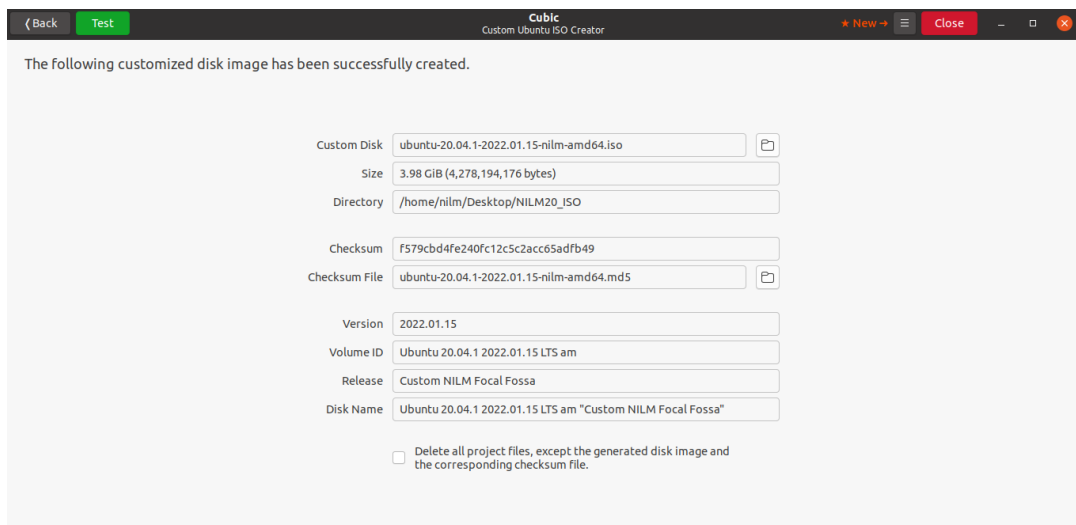




7. Select the compression option of choice (or simply keep the default), then click Generate. Cubic will now generate the customized disk image.



8. Once the ISO has been generated, click Close.



9. Copy the ISO file to a USB drive by running the following commands, where `ISO_NAME` is the full name and location of the ISO and `/dev/sdX` is the location of the USB drive:

Command Line:

```
$> sudo dd if=/dev/zero of=/dev/sdX bs=1k count=2048
$> sudo dd if=ISO_NAME of=/dev/sdX bs=1M
```

The first line erases all data on the drive located at `/dev/sdX`. The drive may need to be unmounted (both in software with the `umount` command and physically) and then physically remounted after this step. Then the second line copies a bootable version of the ISO to the drive.

WARNING: Double-check that you use the correct drive location. You do not want to erase the incorrect drive. It will have the format `/dev/sdX`, where `X` is a letter. You can check a list of all drives by running the `lsblk` command.

C.2 NILM Computer Setup

C.2.1 Installing the ISO

C.2.1.1 Acer

For Acer installations, the NILM Ubuntu operating system can either be installed on the standalone laptop, or while it is in an “all-in-one” (AIO) box. Follow the appropriate steps below to boot from a USB drive containing the NILM Ubuntu ISO.

1. Plug in ac power to the Acer. Insert the bootable NILM Ubuntu USB drive. If installing with the Acer in the AIO box, plug in an external USB keyboard.
2. Power on the Acer. As soon as it starts booting, press the F2 button to launch the BIOS setup utility.
3. Once in the BIOS setup utility, navigate to the Boot page with the right arrow key.
4. On the Boot page, locate the USB drive that was inserted. Scroll down to highlight it and increase its boot priority to 1 using the F6 key.
5. Navigate to the Exit page with the right arrow key and choose “Exit Saving Changes.”

C.2.1.2 GK41

1. Plug in power to the AIO box, and make sure that the GK41 is powered off. Insert the bootable NILM Ubuntu USB drive. Plug in an external USB keyboard.
2. Turn on the GK41. While doing so, press ESC while the GK41 is booting to launch the BIOS setup utility.
3. Once in the BIOS setup utility, navigate to the boot page.
4. Change the boot priority of the USB device to 1.

5. Exit the BIOS setup utility, saving changes.

Now, the system should boot from the USB drive and proceed to the Ubuntu installation dialog. Select “Install Ubuntu” and follow the guided setup. Erase any other operating system installed. If the computer for the installation uses UEFI instead of BIOS, it is necessary to connect to the internet during installation and to allow for updates by selecting “Install Updates.” After the installation is complete and the computer is able to boot to the installed NILM Ubuntu operating system, the BIOS/UEFI settings can be changed to the previous boot priority order so that the drive with the NILM Ubuntu operating system has first boot priority.

C.2.2 Initial Setup

Once installed, follow the steps outlined in the `NILM_README` file located on the Desktop, which are elaborated on here:

1. Authorize local user access to the Joule server:

Command Line:

```
$> sudo -E joule admin authorize
```

2. Connect Joule to the local Lumen server:

Command Line:

```
$> joule master add lumen 127.0.0.1
```

Enter the desired credentials information, which will be used for the Wattsworth data viewer app.

3. To configure data capture, edit `meters.yml` on the Desktop and run the following command. This command must be run any time `meters.yml` has been updated. More details about setting up a `meters.yml` file are in the next two subsections, for contact and noncontact meters, respectively.

Command Line:

```
$> sudo nilm configure
```

4. Restart Joule:

Command Line:

```
$> sudo service joule restart
```

5. Check that data capture and process modules are running:

Command Line:

```
$> joule module list -s
```

The output should look similar to the following:

Name	Inputs	Outputs	CPU %	Mem %
meter1 capture		/meter1/sensor	2	1.2
meter1 process	/meter1/sensor	/meter1/IV /meter1/sinefit /meter1/prep	10	1.3

If the entries for CPU % and Mem % have dashes instead of numbers, there is an error with the corresponding module. The logs for a specific module can be viewed by running the following command. This can help with debugging any errors.

Command Line:

```
$> joule module logs "ModuleName"
```


C.2.3 Setting up a Contact Meter

Connect to the contact meter to the NILM computer using wired Ethernet. Once physically connected, go to the Wired settings in the Network settings. Manually set up an IPv4 connection. Set the Address to an IPv4 address of the form 192.168.X.Y, as long as this address is not the same as the IP address of the LabJack or NerdJack. The default IP address of both the LabJack and NerdJack is 192.168.1.209. Assuming the default IP is used, an example of a valid IP address for the NILM computer is 192.168.1.201. Set the subnet mask (“Netmask”) to 255.255.0.0 (or 16). Click Apply. The settings should look similar to the following:

The screenshot shows the 'Wired' network settings window. At the top, there are 'Cancel' and 'Apply' buttons. Below the title bar, there are tabs for 'Details', 'Identity', 'IPv4', 'IPv6', and 'Security', with 'IPv4' selected. Under 'IPv4 Method', the 'Manual' option is selected with a purple radio button. Other options include 'Automatic (DHCP)', 'Link-Local Only', 'Disable', and 'Shared to other computers'. The 'Addresses' section contains a table with columns for 'Address', 'Netmask', and 'Gateway'. The first row has '192.168.1.201' in the Address column and '255.255.0.0' in the Netmask column. Below this is an empty row. The 'DNS' section has a toggle switch set to 'Automatic' and an empty text input field. A note at the bottom says 'Separate IP addresses with commas'.

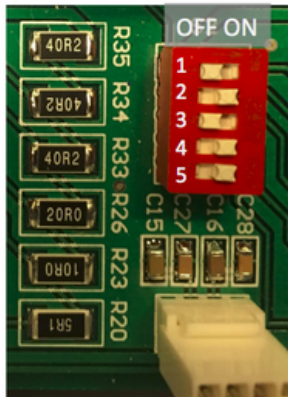
Address	Netmask	Gateway
192.168.1.201	255.255.0.0	

An example `meters.yml` file is shown below for a contact meter:

meters.yml

```
meter1:
  type: contact
  enabled: true # set to false to disable
  daq_type: nerdjack # labjack or nerdjack
  ip_address: 192.168.1.209 # default LJ or NJ address
  phases: 3 # 1 - 3
  sensors:
    voltage:
      sensor_indices: [3,4,5] # maps to phase A,B,C
      sensor_scales: 0.0919 # built-in constant
      sinefit_phase: A # [A,B,C] voltage
      nominal_rms_voltage: 120 # used to scale prep to W
    current:
      sensor_indices: [0,1,2] # maps to phase A,B,C
      sensor_scales: 0.00156402587 # set by resistors and LEM
      sinefit_rotations: [240,0,120]
  filter:
    enabled: false
    #filter_index: 0 # filter index 0 - 5
  streams:
    sinefit:
      decimate: true
      keep: 1m
    iv:
      decimate: true
      keep: 1w
    prep:
      decimate: true
      keep: 3w
      nshift: 1
      goertzel: true
  sensor:
    decimate: true
    keep: 1w
```

The sensors must be scaled correctly to convert the measurements to volts and amps. The voltage scale factor is set in hardware to 0.0919. The current scale factor depends on the load resistance set by the channel DIP switches and the conversion ratio found on the LEM datasheet. The load resistance, R , should be set according to the DIP switches as below, and the corresponding current conversion factor (`sensor_scales` in the `meters.yml` file) can be calculated, where α_{LEM} is the conversion ratio obtained from the LEM datasheet:



Switch	OFF Value
1	80 Ω
2	40 Ω
3	20 Ω
4	10 Ω
5	5 Ω

ON switches add 0 Ω

Current Conversion Factor

$$\frac{5.07 - (-5.18)}{2^{16}} \times \frac{\alpha_{LEM}}{R}$$

α_{LEM} : from LEM datasheet

Voltage Conversion Factor

0.0919 (fixed)

For shipboard installations, R is typically set to 35 Ω . Typically deployed LEMs are the LF 305-S (with maximum measurable current of 300 A) and LF 505-S (with maximum measurable current of 500 A). In the lab, R is typically set to 100 Ω , with the LA-55P LEMs (with maximum measurable current of 50 A). The current conversion ratios for these setups are listed in the table below.

	α_{LEM}	R	Current conversion factor
LA 55-P	1000	100 Ω	0.001564026
LF 305-S	2000	35 Ω	0.008937291
LF 505-S	5000	35 Ω	0.02234323

Different LEMs and different R values will yield different current conversion factors. Typically these values should be set such that R is within the allowable maximum load resistance as specified in the LEM datasheet, while also using the full voltage range of the DAQ without clipping (usually ± 5 V). For instance, for the LA 55-P

setup described, the voltage range is $\pm 50 \cdot \frac{1}{1000} \cdot 100 = \pm 5$ V. The above assumes that a closed loop current transducer with current output is used. An open loop current transducer with voltage output has also been tested, which has a split core for easy mounting. The tested sensor, LEM HTR 300-SB has an output voltage of ± 4 V. For this sensor, a resistance greater than 10 k Ω should be used (i.e., one of the resistors set by the DIP switch will need to be replaced).

Additional notes about the `meters.yml` file:

- The `sinefit_rotations` will need to be updated by cycling a known load and viewing the prep streams. For a wye-configured contact box, the sinefit rotations will be some permutation of [0, 120, 240] degrees. For a delta-configured contact box, the sinefit rotations will be either a permutation of [30, 150, 270] degrees or [90, 210, 330] degrees. This comes about because it is possible for there to be a phase shift of 180 degrees in the voltages with respect to the currents.
- For each of the streams, a time duration can be specified for how much data to keep (h for hours, d for days, m for months, or y for years).
- If using the notch filter of the NerdJack, the `filter: enabled` should be set to true. The `filter_index` should be uncommented and the desired filter index specified.

IMPORTANT: Enabling the NerdJack filter results in a sensor stream with seven columns. Using a NerdJack with the filter disabled or using a LabJack results in a sensor stream with six columns. These cannot be combined. A new meter will need to be created (e.g., meter2) if switching between LabJack and NerdJack or disabling and enabling the NerdJack filter.

C.2.4 Setting up a Noncontact Meter

To calibrate a noncontact meter, first set up the `meters.yml` file. An example `meters.yml` file is shown:

meters.yml

```
meter3:
  type: noncontact
  enabled: true
  serial_number: meter1002          # found on the meter case
  phases: 3                        # 1 - 3
  sensors:
    voltage:
      sensor_index: 0              # electric field sensor
      digitally_integrate: true    # if true, integrate using
        FIR filter
      nominal_rms_voltage: 120    # scale the electric field
    current:
      sensor_indices: [1,3,5,7]   # D-Board with 4 A-Boards
  calibration:
    duration: 30                  # calibration time (seconds)
    on_duration: 1                # load on duration
    off_duration: 2              # load off duration
    watts: 205                    # power of calibration load
    has_neutral: true            # [false] if the system has
      no neutral bus
  streams:
    sinefit:
      decimate: true
      keep: 1m
    iv:
      decimate: true
      keep: 1w
    prep:
      decimate: true
      keep: 3w
      nshift: 1                   # output freq = nshift*60
      goertzel: true              # if [false] uses fft
    sensor:
      keep: false
```

Make sure to double-check the `serial_number` and the information about the calibration load under `calibration`. The `duration` is the total calibration time (for each phase), `on_duration` is the on time of the calibration load, `off_duration` is the off time of the calibration load, and `watts` is the steady-state power of the calibration load.

The calibration procedure is given below:

1. Stop Joule:

Command Line:

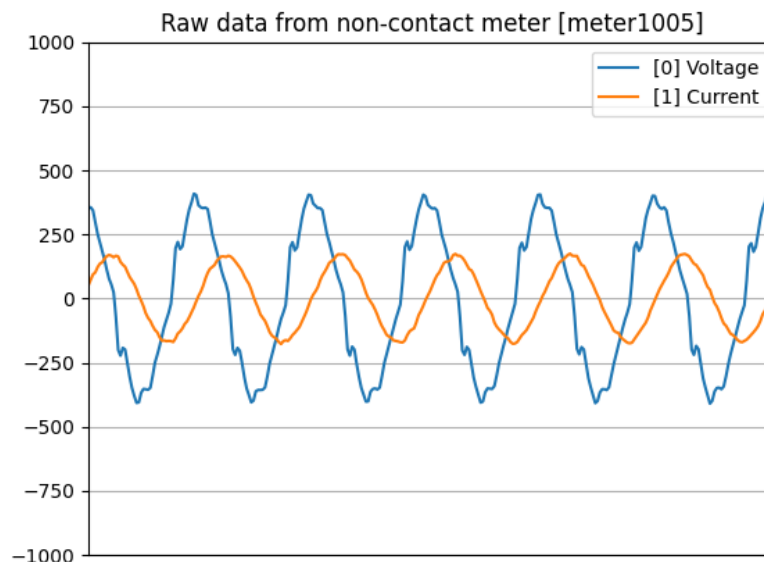
```
$> sudo service joule stop
```

2. (OPTIONAL) You may want to `nilm scope` the meter before starting calibration to check the signal:

Command Line:

```
$> sudo nilm scope meterX -c 0 1
```

The `-c, --channel` argument indicates which channel(s) to plot. An example `nilm scope` output for a single-phase non-contact sensor with channel argument `0 1` is shown:



3. Run `nilm calibrate` for the corresponding meter in the meters.yml file:

Command Line:

```
$> sudo nilm calibrate meterX
```

4. Double-check the calibration settings. If correct enter y and press ENTER.

```
Calibrating meter3
+ The power system has 3 phases and neutral
+ Digitally integrating sensor 0 for voltage measurement
+ Using sensors 1,3,5,7 for current measurements
+ Calibration load is 205W and will run for 30 seconds
+ Calibration load is has pattern: 1.000000 seconds ON and 2.000000 seconds OFF
+ The reference voltage is 120V rms
+ The meter serial number is [meter1002]
+ Visualization is [OFF]
Is this correct? (y/n) █
```

5. Follow the instructions given for calibrating each phase.
6. Save calibration results when prompted.

IMPORTANT: The default calibration pattern is 1 second on and 2 seconds off. This can be changed by connecting to the smart plug through USB.

To change the calibration pattern, plug in the smart plug, then connect to it with the following command:

Command Line:

```
$> nilm-plug --cli /dev/ttyACM0
```

Note that the location of the device might be different. The location can be found by running the following command and finding the device corresponding to the smart plug (on the Acer laptops it usually starts with ttyACM):

Command Line:

```
$> ls /dev/tty*
```

Connecting to the smart plug will open the plug command line interface. If successful, the light on the smart plug should turn blue, and the output should look like the

following:

```
nilm@nilm:~$ nilm-plug --cli /dev/ttyACM0
/dev/ttyACM0, 115200 baud
^C to exit
-----
> |
```

Once in the smart plug command line interface, the calibration pattern can be set with the following command:

```
Command Line:
$> calibrate start on_time_ms off_time_ms
```

The on and off times should be in milliseconds (e.g., for a 1 second on and 2 seconds off pattern, it should be set to 1000 and 2000, respectfully).

C.3 Data Backup

For redundancy, it is helpful to store data both on the computer's hard drive and on an external hard drive for easy access and retrieval. To do so, the Linux `rsync` command is used to copy data from the primary NilmDB database at `/opt/data` to a secondary NilmDB database at `/opt/secondary`. The first step is to create an `/opt/secondary` directory. This only has to be done once:

```
Command Line:
$> sudo mkdir /opt/secondary
```

C.3.1 External Drive Setup

To set up an external drive as `/opt/secondary`, it first must be formatted correctly, as outlined in the steps below.

1. When the drive is plugged in, if it has a mount point (e.g., `/media/nilm/...`), unmount it with the `umount` command.

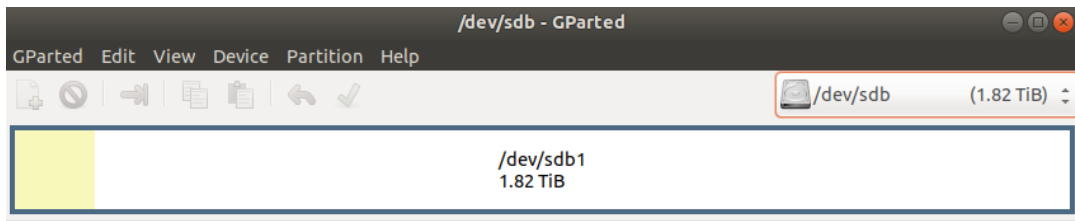
2. Format the drive using GParted, which can be installed with the following:

```
Command Line:  
  
$> sudo apt update  
$> sudo apt install gparted
```

Once GParted is installed, run GParted:

```
Command Line:  
  
$> sudo gparted
```

3. Now, select the correct drive for formatting. For example:

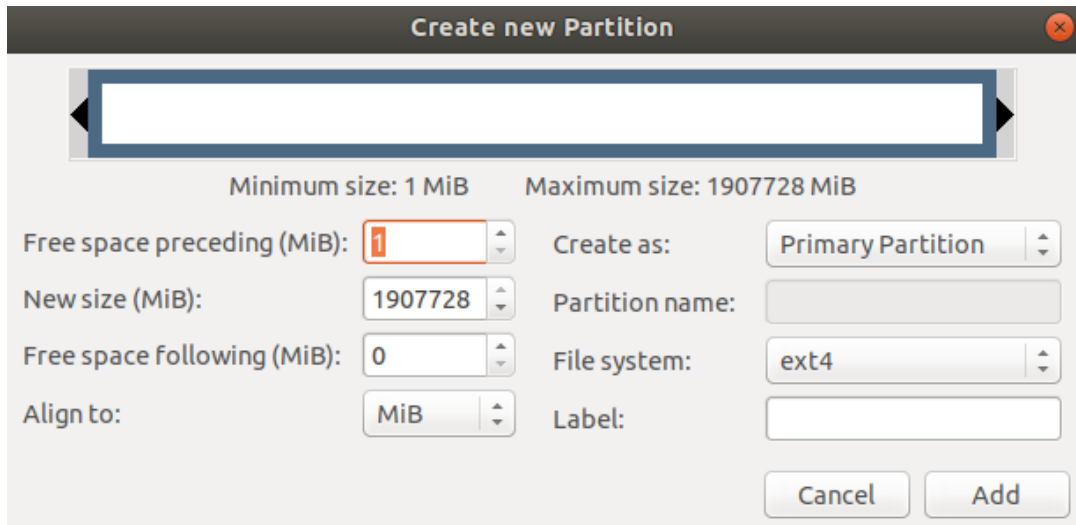


WARNING: Make sure you select the correct drive. It will have the format /dev/sdX, but it will NOT always be /dev/sdb.

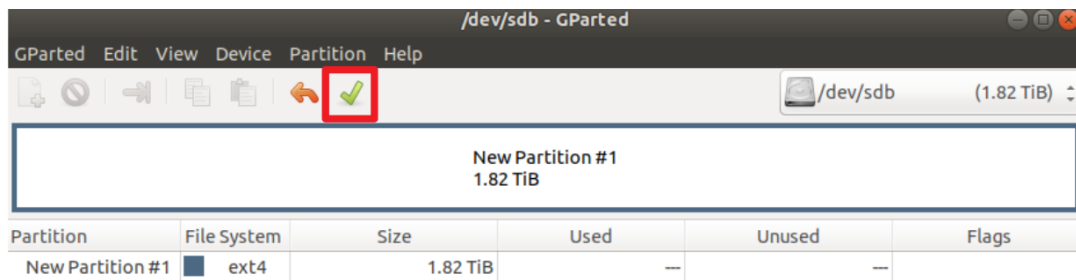
4. Select Device > Create Partition Table. Select msdos in the “Select new partition table type:” dropdown and click Apply.



5. Create a new partition by selecting Partition > New. In the menu options, make sure to select ext4 for the File system. Click Add.



6. To apply all of these settings, click the green check mark, as shown below:



7. Once the partitioning is complete, exit GParted.

C.3.2 Manually Mounting `/opt/secondary`

Every drive should be manually mounted and tested to make sure everything is operating correctly, as outlined below:

1. Make sure the external drive is unmounted.
2. Mount it in command line. For example, if the external drive is located at `/dev/sdb1`:

```
Command Line:
$> sudo mount /dev/sdb1 /opt/secondary
```

3. Give the nilm user ownership of `/opt/secondary` :

Command Line:

```
$> sudo chown -R nilm:nilm /opt/secondary
```

4. Remove `lost+found` files from `/opt/secondary` :

Command Line:

```
$> rm -rf /opt/secondary/lost+found
```

IMPORTANT: These steps should proceed without error. The external drive is now set up and formatted correctly. The external drive will not be mounted by default when the computer is rebooted.

C.3.3 Automatically Mounting `/opt/secondary`

For a more permanent solution, a cron job can be created to automatically mount `/opt/secondary` and make sure it remounts if it ever disconnects or the computer reboots. `cron` is a Linux tool to schedule commands to be run at user-designated time intervals. The universally unique identifier (UUID) of the drive will be used to automatically mount it. To find the UUID of the external drive, use the command `sudo blkid`. Once you have the UUID, the following file should be created, where `XX` is the UUID of the external drive:

`/etc/cron.d/mount_secondary`

```
*/15 * * * * root mount UUID=XX /opt/secondary && chown -R  
nilm:nilm /opt/secondary
```

IMPORTANT: Make sure the drive has already been manually formatted without error. Make sure you are using the correct UUID.

An example file is stored in `/home/nilm/Desktop/example_cron_jobs/mount_secondary`.

C.3.4 Copying Data to External Drive

To copy the data to the external drive, the `rsync` command can be used:

Command Line:

```
$> rsync -av --delete /opt/data/ /opt/secondary/
```

To do this automatically at a set interval, a cron job can be used by creating the following file:

```
/etc/cron.d/rsync_external
```

```
*/15 * * * * nilm rsync -av --delete /opt/data/ /opt/secondary/
```

An example file is stored in `/home/nilm/Desktop/example_cron_jobs/rsync_external`.

C.3.5 Change External Drive UUID

For easy data access, it is convenient to be able to swap out the external drive without needing to change any configurations. This can be achieved by using two (or more) external drives with the same UUID. The new drive will automatically be mounted by the cron job given in Appendix C.3.3.

Select one drive and find its UUID by running the following command:

Command Line:

```
$> sudo blkid
```

Once you have the UUID that you would like the other drives to have, unmount and unplug the original drive. Plug in the new drive whose UUID is to be changed. Run the following command with the desired UUID, where `/dev/sdXY` is the partition on the new drive:

Command Line:

```
$> tune2fs -U <UUID> /dev/sdXY
```

C.3.6 Check NilmDB Connection

By creating the following file, a cron job will periodically run the `check_nilmdb.py` script to check the NILM's connection to NilmDB and Joule:

```
/etc/cron.d/check_nilmdb

*/15 * * * * nilm /usr/bin/python3 /home/nilm/Desktop/
example_cron_jobs/check_nilmdb.py
```

Unlike the other cron jobs, this one is already set up by default. The script `check_nilmdb.py` checks the connection to NilmDB and Joule. If it cannot connect to Joule or if there is no data in the specified stream (the script automatically selects the stream of interest using the enabled stream in the `meters.yml` file), then Joule is restarted. Every time Joule restarts this way it is logged in the `check_nilmdb.txt` file on the Desktop for user reference and debugging.

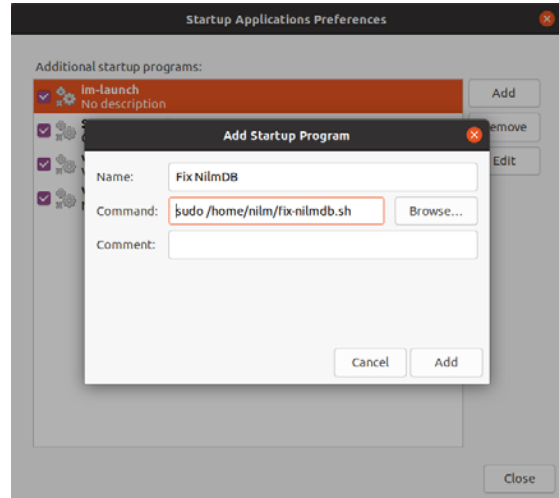
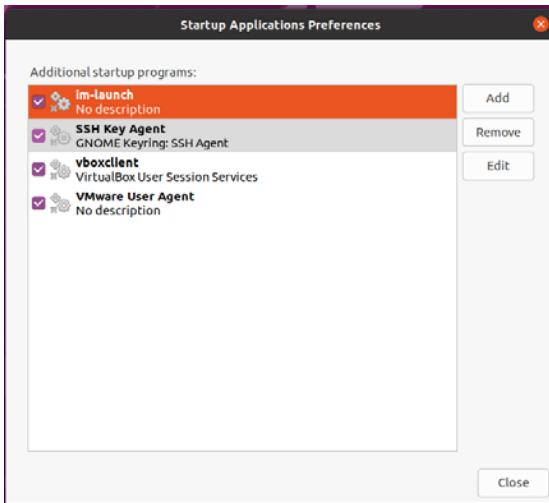
C.4 Fix NilmDB on Startup

Whenever the NILM computer reboots suddenly, there is a chance that NilmDB will be corrupted. This can be automatically fixed using a startup script. This is particularly helpful for NILM computers that do not run on a battery backup. To set this up, create the following file, changing DATABASE to the location of the NilmDB database (`/opt/data` by default):

```
/home/nilm/fix-nilmdb.sh

#!/bin/bash
service joule stop
service nilmdb stop
nilmdb-fsck DATABASE --fix -n > /home/nilm/nilmdb-log.txt
service nilmdb restart
service joule restart
echo "Fixed nilmdb at $(date)." >> /home/nilm/nilmdb-log.txt
```

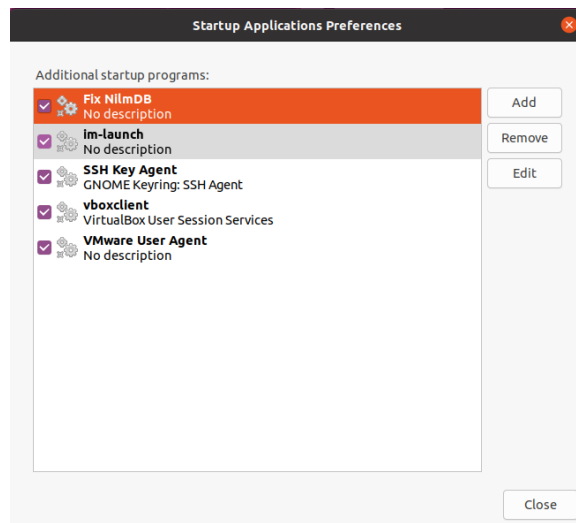
Next, open “Startup Applications” and click Add:



Set Name to be Fix NilMDB and Command to the following:

```
sudo /home/nilm/fix-nilmdb.sh
```

Click Add. The new entry should show up in the list as shown below:



Finally, in the terminal, run `sudo visudo`. A file similar to the following should be brought up for editing:

```
nilm@nilm: ~
GNU nano 4.8 /etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Add the following line to the end of the file that is brought up for editing:

```
nilm ALL=(ALL:ALL) NOPASSWD:/home/nilm/fix-nilmdb.sh
```

Save and exit the file.

WARNING: Anything in `/home/nilm/fix-nilmdb.sh` will be automatically run with root privileges whenever the system starts up. For this reason, be very careful when modifying this file.

C.5 NILM Dashboard Demo

For demonstration, a NILM Dashboard demo can be set up. This demo will use an event detector to identify loads with known rated real power. The event detector script for this demo is in Appendix C.10. The event detector is a Joule module which takes a NilMDB prep stream as input. The stream of interest should be specified in the Joule configuration file, similar to the following example (located at `/home/nilm/Desktop/dashboard-settings/example_module/event_detector.conf`).

event_detector.conf

```
[Main]
name = Demo Event Detector
exec_cmd = python3 /opt/nilm-dashboard-v2/example/event_detector
          .py --config /home/nilm/Desktop/dashboard-settings/
          development.yml

[Inputs]
input = /meter1/prep
```

The configuration file for the event detector Joule module can be created and edited using the following command:

Command Line:

```
$> sudo nano /etc/joule/module_configs/event_detector.conf
```

The event detector does not output to NilmDB. Instead, it outputs the detected events to a SQL database. The database file for the demo has already been created (as an empty file), and it is located at `/opt/nilm-dashboard-v2/demoDB.db`.

C.5.1 Server Configuration Files

Dashboard has several configuration files. The first two, `/opt/nilm-dashboard-v2/config/development.yml` and `/opt/nilm-dashboard-v2/adev_vars.sh` have already been set up. These are shown below for reference. If the name of the SQL database changes, the `/opt/nilm-dashboard-v2/config/development.yml` file should be updated to reflect the change.

development.yml

```
# Database connection information in DSN notation
# See https://docs.sqlalchemy.org/en/13/core/engines.html#
  database-urls
database_dsn: 'sqlite:///opt/nilm-dashboard-v2/demoDB.db'

# IP address and port, ignored by the adev server, see adev_vars
  .sh
host: 127.0.0.1
port: 8080
```

adev_vars.sh

```
# To run the development server change to the dashboard
  directory then source this file:
#   $> cd dashboard/
#   $> source adev_vars.sh
#   $> adev runserver

# Configuration file location
export DASHBOARD_CONFIG=/home/nilm/Desktop/dashboard-settings/
  development.yml

# For information on all development server settings run
#   $> adev runserver --help
export AIO_PORT=8080
export AIO_HOST=127.0.0.1
```

C.5.2 Configuring library.yml

There is a user-configurable file located in `/home/nilm/Desktop/dashboard_settings/library.yml`. This lets the user add loads for detection. There are general settings, starting with `enabled`. This should be set to true or false. For `input`, the prep stream for performing identification should be specified. For `phases` either 1 or 3 should be specified, indicating the number of phases of the power system (not the

load). For each load, there is a `name` field which is the name that is displayed on Dashboard. Each load should have a unique number in the `index` field. The `power` is the expected real power consumption for the load. An example file is shown below:

```
library.yml

# This file lists the appliances for this NILM #
demo:
  enabled: true
  input: /meter1/prep
  phases: 3
  loads:
    appliance1:
      enabled: true           # true or false
      name: 270 W Light      # name of appliance
      index: 0               # index of appliance
      power: 270             # rated wattage
    appliance2:
      enabled: true           # true or false
      name: 210 W Light      # name of appliance
      index: 1               # index of appliance
      power: 210             # rated wattage
    appliance3:
      enabled: true           # true or false
      name: 60 W Light       # name of appliance
      index: 2               # index of appliance
      power: 60              # rated wattage
```

C.5.3 Running Dashboard Server

Once all configurations are set up, the server can be run as follows:

```
Command Line:

$> cd /opt/nilm-dashboard-v2
$> python3 dashboard/main.py --config ./config/development.yml
```

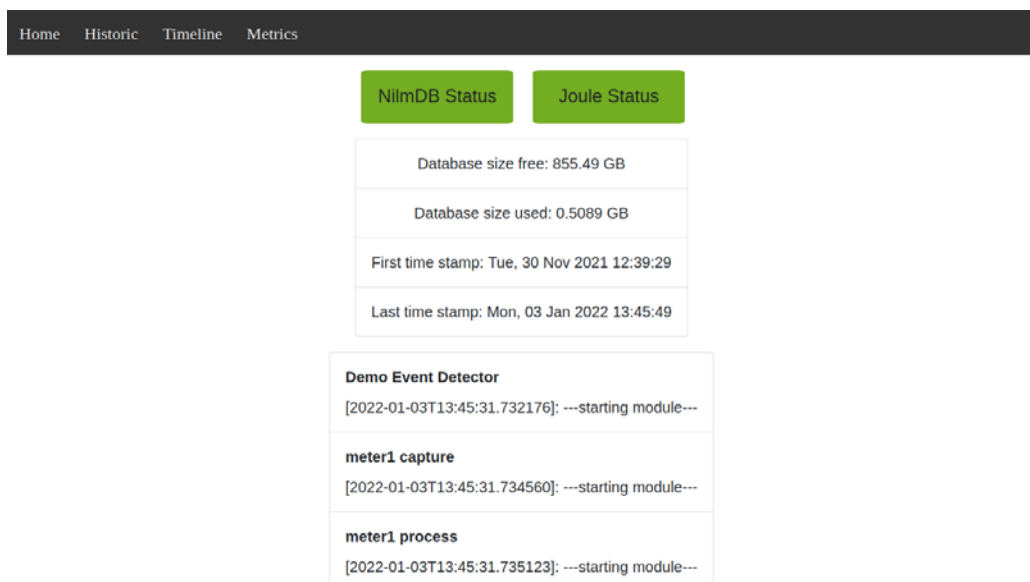
NILM Dashboard can now be accessed at `localhost:8080` on a web browser.

C.5.4 Example Dashboard Views

Dashboard has four pages: Home, Timeline, Historic, and Metrics. This section summarizes these four views.

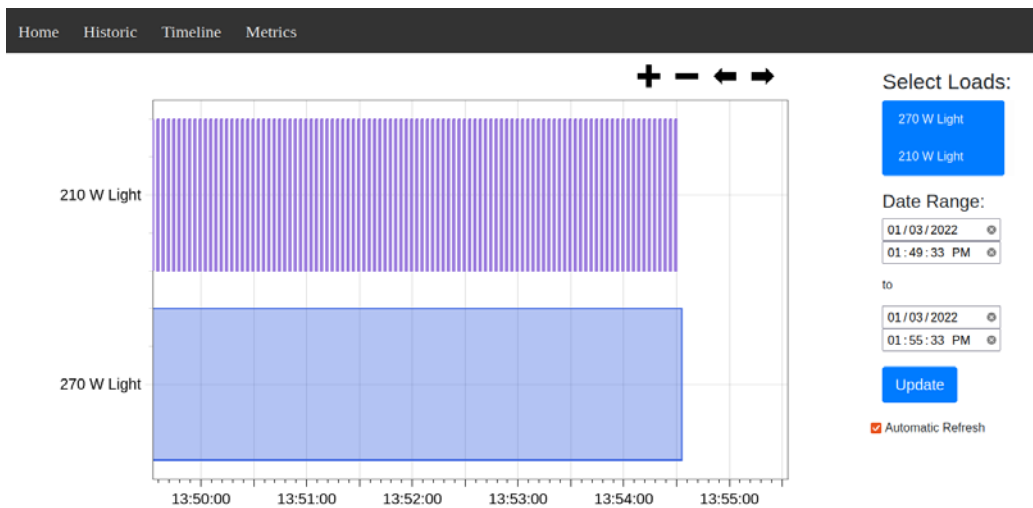
C.5.4.1 Home Page

The Home page displays the status of NilMDB and Joule. If NilMDB and Joule are operating properly, both are shown as green. If there is no connection to NilMDB, the NilMDB Status is shown as red. Likewise, if there is no connection to Joule, the Joule Status will be shown as red. The Home page also shows the NilMDB disk space remaining and used so far, so that the user knows how much space is left. Also displayed are the first and last timestamps of the prep stream. Dashboard uses the `meters.yml` file to automatically select which meter's prep stream to look at. If the last timestamp is not the current time, there is a problem with the data capture. Finally, the Joule module logs are displayed for all running modules. If there are problems in the data capture, they will likely be displayed in the logs. The Home page automatically refreshes every 10 seconds. An example Home page is displayed below:



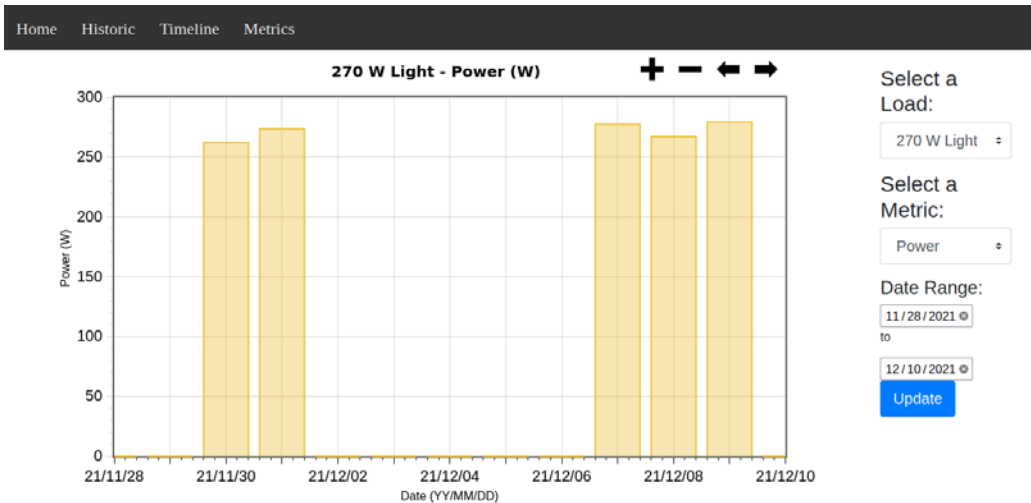
C.5.4.2 Timeline Page

The Timeline page provides a live view of equipment status, displaying colored blocks to represent periods when each load is energized. There is an “Automatic Refresh” option in the lower-right side that, when selected, automatically refreshes the page every five seconds. It keeps the view to the last five minutes of operation. When “Automatic Refresh” is not selected, the user can zoom in and out and pan. An example Timeline page is displayed below:



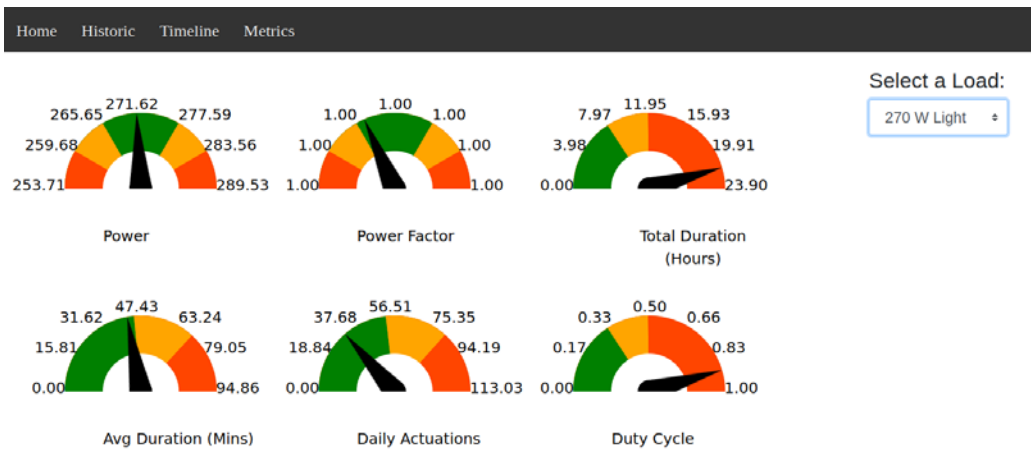
C.5.4.3 Historic Page

The Historic page provides short- and long-term trend data. The user can select a single load and single metric from the dropdown menus. The six metrics are power, power factor, total duration, average duration, daily actuations, and duty cycle. The Historic view is presented as a bar graph, where each bar represents one day of operation. An example Historic page is displayed below:



C.5.4.4 Metrics Page

The Metrics page provides the user with six metrics for a selected piece of equipment as a set of gauges. Each metric is displayed as a gauge with green, yellow, and red sections. Green indicates normal operations, while yellow and red indicate increasing likelihood of a fault. The colored sections are derived from statistics of previous load operation. However, this may need to be changed if there is other known information about normal operation. The gauge needle position represents the value for the previous 24 hours of operation. An example Metrics page is displayed below:



C.6 Useful Joule and NILM Commands

Stop, start, restart, or check status of Joule:

Command Line:

```
$> sudo service joule [stop|start|restart|status]
```

Stop, start, restart, or check status of NilMDB:

Command Line:

```
$> sudo service nilmdb [stop|start|restart|status]
```

Get information about NilMDB server, such as used and available storage space:

Command Line:

```
$> nilmtool info
```

List active Joule modules (the -s flag includes memory and CPU statistics):

Command Line:

```
$> joule module list -s
```

List all Joule streams:

Command Line:

```
$> joule stream list
```

Additional options include:

- l, --layout: include stream layout
- s, --status: include stream status
- i, --id: show IDs

IMPORTANT: The Joule IDs are necessary when using `nilm-copy` to copy data streams, as NilmDB stores the streams as `/joule/1`, `joule/2`, etc.

List all NilmDB streams:

Command Line:

```
$> nilmtool list -Enl
```

-E: shows extended stream information, such date range of data

-n: skips decimation streams

-l: show layout of data streams

List Joule logs for a particular module:

Command Line:

```
$> joule module logs "NAME"
```

To copy data using Joule to another Joule node or to a NilmDB URL:

Command Line:

```
$> joule data copy [OPTIONS] SOURCE DESTINATION
```

SOURCE and DESTINATION are the Joule stream names.

Additional options include:

-s, --start: timestamp or descriptive string

-e, --end: timestamp or descriptive string

-d, --destination-node: node name or NilmDB URL

--source-url: copy from a NilmDB URL

Alternatively, data can be copied using `nilm-copy`:

Command Line:

```
$> nilm-copy -u <source URL> -U <dest URL> SOURCE DESTINATION
```

SOURCE and DESTINATION are NilmDB stream names.

Additional options include:

-s, --start: timestamp or descriptive string

-e, --end: timestamp or descriptive string

-F, force metadata changes, sometimes required if metadata in source and destination do not match

To download data using `nilmtool extract`:

Command Line:

```
$> nilmtool -u <source URL> extract [OPTIONS] path
```

Additional options include:

-s, --start: timestamp

-e, --end: timestamp

C.7 Repairing NilmDB

In the event of abrupt power loss, NilmDB will likely need to be fixed. To do so, first stop Joule and NilmDB:

Command Line:

```
$> sudo service joule stop
$> sudo service nilmdb stop
```

Next, run the `nilmdb-fsck` command as follows, where DATABASE is the location of the NilmDB server (`/opt/data` by default):

Command Line:

```
$> sudo nilmdb-fsck DATABASE -fn
```

Omitting the `-n` flag will check and fix the database row by row. This is very time-consuming and often not necessary when NilmDB is corrupted by a sudden loss of power. Once NilmDB has been fixed, start NilmDB and Joule:

Command Line:

```
$> sudo service joule start
$> sudo service nilmdb start
```

C.8 Known Issues

- If an incorrect monitor refresh rate is set, this may lead to burn-in on the external monitor used in the AIO box. Ensure that the correct refresh rate, 60 Hz, is set in Settings > Displays.

C.9 Installation Files

The following are the scripts for customizing an installable NILM Ubuntu ISO.

Listing C.1: install-iso.sh

```
1 #!/bin/bash
2
3 set -e
4
5 bash ./install.sh
6
7 echo -en "$COL_GREEN"
8 cat <<EOF
9
10 -----Install Complete-----
11
12 EOF
13 echo -e "$COL_RESET"
```

Listing C.2: install.sh

```
1 #!/bin/bash
2
3 set -e
4 # Colors
5
6 ESC_SEQ="\x1b["
7 COL_RESET=$ESC_SEQ"39;49;00m"
8 COL_RED=$ESC_SEQ"31;01m"
9 COL_GREEN=$ESC_SEQ"32;01m"
10 COL_YELLOW=$ESC_SEQ"33;01m"
11 COL_BLUE=$ESC_SEQ"34;01m"
12 COL_MAGENTA=$ESC_SEQ"35;01m"
13 COL_CYAN=$ESC_SEQ"36;01m"
14
15 # Make sure only root can run our script
16 if [[ $EUID -ne 0 ]]; then
17 echo -en "$COL_RED"
18 cat <<EOF
19 Error: This script must be run as root
20     $ sudo bash install.sh
21 EOF
```

```

22 echo -e "$COL_RESET"
23 exit
24 fi
25
26 echo -en "$COL_YELLOW"
27 cat <<EOF
28 *****
29 * MIT NILM Installer *
30 * Updated January 2022 *
31 *****
32
33 EOF
34 echo -en "$COL_RESET"
35
36 #-----
37 # Set up the system
38 #-----
39 echo -en "$COL_BLUE"
40 cat <<EOF
41 A. Add [nilm] user Configuring System
42 EOF
43 echo -en "$COL_RESET"
44 #add the nilm user if it doesn't exist
45 if ! id -u nilm > /dev/null 2>&1; then
46     useradd --shell /bin/bash --user-group --home /home/nilm -m nilm
47     echo nilm:nilm | chpasswd #set password to nilm
48     mkdir -p /home/nilm/Desktop
49     chown nilm:nilm /home/nilm/Desktop
50     usermod -a -G sudo nilm
51 fi
52
53 python3 -m pip install --upgrade pip > /dev/null
54
55 #-----
56 # install nilm config files
57 #-----
58
59 echo -en "$COL_BLUE"
60 cat <<EOF
61 B. NILM Config Files
62 EOF
63 echo -en "$COL_RESET"
64
65 # Install the stub meters.yml file
66 cp scripts/meters.yml /opt/configs/meters.yml
67
68 # Link the meters file to the desktop
69 ln -sf /opt/configs/meters.yml /home/nilm/Desktop/meters.yml
70 chown nilm:nilm /home/nilm/Desktop/meters.yml
71
72 # Install the example meters file
73 cp scripts/example_meters.yml /home/nilm/Desktop/example_meters.yml
74 chown nilm:nilm /home/nilm/Desktop/example_meters.yml
75
76 # Copy the configuration README
77 cp scripts/NILM_README /home/nilm/Desktop/NILM_README
78 chown nilm:nilm /home/nilm/Desktop/NILM_README
79
80 # Make folder for example cron jobs
81 mkdir -p /home/nilm/Desktop/example_cron_jobs
82
83 #copy cron job for check_nilmdb
84 cp scripts/check_nilmdb /etc/cron.d/
85 cp scripts/check_nilmdb.py /home/nilm/Desktop/example_cron_jobs/check_nilmdb.py
86 chown nilm:nilm /home/nilm/Desktop/example_cron_jobs/check_nilmdb.py
87
88 # copy all cron jobs to examples folder #
89 cp scripts/check_nilmdb /home/nilm/Desktop/example_cron_jobs/

```

```

90 cp scripts/mount_secondary /home/nilm/Desktop/example_cron_jobs/
91 cp scripts/rsync_external /home/nilm/Desktop/example_cron_jobs/
92 chown -R nilm:nilm /home/nilm/Desktop/example_cron_jobs
93
94 cp scripts/20_nilm /etc/sudoers.d/20_nilm
95
96 # copy NILM wallpaper and set as default
97 cp img/NILM_Wallpaper.png /usr/share/backgrounds/
98 mv /usr/share/backgrounds/warty-final-ubuntu.png /usr/share/backgrounds/warty-
    final-ubuntu-old.png
99 mv /usr/share/backgrounds/NILM_Wallpaper.png /usr/share/backgrounds/warty-final-
    ubuntu.png
100
101 #-----
102 # install nilm dashboard
103 #-----
104
105 echo -en "$COL_BLUE"
106 cat <<EOF
107 C. Install NILM Dashboard
108 EOF
109 echo -en "$COL_RESET"
110
111 # clone the source to /opt
112 git clone nilm-dashboard-v2 /opt/nilm-dashboard-v2
113 URL=$(git -C nilm-dashboard-v2 config remote.origin.url)
114 git -C /opt/nilm-dashboard-v2 config remote.origin.url $URL
115
116 groupadd dashboard
117 usermod -a -G dashboard nilm
118 usermod -a -G dashboard joule
119
120 chown -R nilm:dashboard /opt/nilm-dashboard-v2/
121
122 # install nilm-dashboard
123 cd /opt/nilm-dashboard-v2
124 pip install -r requirements.txt > /dev/null
125 python3 setup.py develop > /dev/null
126 cd - > /dev/null
127
128 cp scripts/development.yml /opt/nilm-dashboard-v2/config/development.yml
129 cp scripts/adev_vars.sh /opt/nilm-dashboard-v2/adev_vars.sh
130
131 mkdir -p /home/nilm/Desktop/dashboard-settings
132 ln -sf /opt/nilm-dashboard-v2/config/development.yml /home/nilm/Desktop/dashboard-
    settings/
133 ln -sf /opt/nilm-dashboard-v2/adev_vars.sh /home/nilm/Desktop/dashboard-settings/
134 ln -sf /opt/nilm-dashboard-v2/example/library.yml /home/nilm/Desktop/dashboard-
    settings/
135
136 touch /opt/nilm-dashboard-v2/demoDB.db
137
138 # copy example conf file #
139 mkdir -p /home/nilm/Desktop/dashboard-settings/example_module
140 cp scripts/event_detector.conf /home/nilm/Desktop/dashboard-settings/
    example_module/
141
142 chown -R joule:dashboard /opt/nilm-dashboard-v2/
143 chmod -R g+w /opt/nilm-dashboard-v2/
144 chown -R nilm:nilm /home/nilm/Desktop/dashboard-settings/
145
146 # set default Firefox pages #
147 cp scripts/syspref.js /etc/firefox/syspref.js

```

C.10 Demo Event Detector

Listing C.3: `event_detector.py`: An example event detector implemented as a Joule module and compatible with NILM Dashboard.

```
1 from joule import FilterModule
2 import asyncio
3 import numpy as np
4 import yaml
5 from scipy import signal
6 from scipy.signal import medfilt
7 import sys
8
9 from sqlalchemy import create_engine
10 from sqlalchemy.orm import Session
11 from typing import List
12
13 from dashboard.models.meta import Base
14 from dashboard.models.load import Load
15 from dashboard.models.event import Event
16 from dashboard.settings import parse_configs
17
18 from datetime import datetime
19
20 appliance_file = "/home/nilm/Desktop/dashboard-settings/library.yml"
21
22 def find_ss_on(pstream, ind_start, trans_end, ind_end, pos):
23     return np.median(pstream[trans_end:ind_end]) - np.median(pstream[ind_start:pos
24     - 1])
25
26 def find_ss_off(pstream, ind_start, ind_end, pos):
27     return np.median(pstream[ind_start:pos - 1]) - np.median(pstream[pos + 1:
28     ind_end])
29
30 def cycling_load_on(load_id: int, power: float, power_factor: float, num_runs,
31 on_times, states) -> List[Event]:
32     events = []
33     for i in range(num_runs):
34         events.append(Event(load_id=load_id, power=power[i], power_factor=
35         power_factor[i], type=Event.EventType.ON, start_time=on_times[i], state=states[i
36         ]))
37     return events
38
39 def cycling_load_off(load_id: int, power: float, power_factor: float, num_runs,
40 off_times, states) -> List[Event]:
41     events = []
42     for i in range(num_runs):
43         events.append(Event(load_id=load_id, power=power[i], power_factor=
44         power_factor[i], type=Event.EventType.OFF, start_time=off_times[i], state=states
45         [i]))
46     return events
47
48 def cycling_load_run(load_id: int, power: float, power_factor: float, num_runs,
49 times, states) -> List[Event]:
50
51     ind = np.argsort(times, axis=0)
52     power = power[ind]
53     power_factor = power_factor[ind]
54     times = times[ind]
55     states = states[ind]
56
57     for i in range(num_runs-1):
58         events = []
59         if states[i] == 1 and states[i+1] == 2:
60             events.append(Event(load_id=load_id, power=power[i+1], power_factor=
61             power_factor[i+1], type=Event.EventType.RUN, start_time=times[i], end_time=
```

```

        times[i+1]))
52     return events
53
54 class Demo_Detector(FilterModule):
55     " Detects and classifies resistive loads and outputs to sql database "
56
57     def __init__(self):
58         self.stop_requested = False
59         self.description = "Classifies resistive loads with a known steady state
power"
60         self.help = "Classifies resistive loads with a known steady state power"
61
62         with open(appliance_file, 'r') as f:
63             appliances = yaml.safe_load(f)
64
65         settings = appliances['demo']['loads']
66
67         self.num_of_phases = appliances['demo']['phases']
68         self.num_of_loads = 0
69         self.power_levels = []
70         self.index = []
71         self.load_name = []
72         for key, my_appliance in settings.items():
73             self.power_levels.append(int(my_appliance['power']))
74             self.index.append(my_appliance['index'])
75             self.load_name.append(my_appliance['name'])
76             self.num_of_loads +=1
77
78     def custom_args(self, parser):
79         parser.add_argument("-c", "--config", help="configuration file", required=
True)
80
81     async def run(self, parsed_args, inputs, outputs):
82         stream_in = inputs["input"]
83
84         config = parse_configs(parsed_args.config)
85         if config is None:
86             exit(1) # error
87
88         engine = create_engine(config.database_dsn)
89         Base.metadata.create_all(engine)
90         db = Session(bind=engine)
91
92         while(not self.stop_requested):
93             sarray_in = await stream_in.read()
94
95             time = sarray_in['timestamp']
96             data_in = sarray_in['data']
97
98             # minimum length of data block
99             if(len(sarray_in) < (300)):
100                 await asyncio.sleep(1)
101                 continue
102
103             # length of data block
104             array_length = len(sarray_in)
105             WINDOW_SIZE = 11
106
107             if self.num_of_phases == 3:
108
109                 PA = data_in[:,0]
110                 PB = data_in[:,8]
111                 PC = data_in[:,16]
112
113                 QA = data_in[:,1]
114                 QB = data_in[:,9]
115                 QC = data_in[:,17]
116

```

```

117         # median filter to smooth
118         PA_filt = medfilt(PA,WINDOW_SIZE)
119         QA_filt = medfilt(QA,WINDOW_SIZE)
120         PB_filt = medfilt(PB,WINDOW_SIZE)
121         QB_filt = medfilt(QB,WINDOW_SIZE)
122         PC_filt = medfilt(PC,WINDOW_SIZE)
123         QC_filt = medfilt(QC,WINDOW_SIZE)
124
125
126         PQA_filt = np.hstack((PA_filt[:,None],QA_filt[:,None]))
127         PQB_filt = np.hstack((PB_filt[:,None],QB_filt[:,None]))
128         PQC_filt = np.hstack((PC_filt[:,None],QC_filt[:,None]))
129
130         PQ_filt = np.hstack((PA_filt[:,None],QA_filt[:,None],PB_filt[:,
None],QB_filt[:,None],PC_filt[:,None],QC_filt[:,None]))
131
132         elif self.num_of_phases == 1:
133             P = data_in[:,0]
134             Q = data_in[:,1]
135             P_filt = medfilt(P,WINDOW_SIZE)
136             Q_filt = medfilt(Q,WINDOW_SIZE)
137
138             PQ_filt = np.hstack((P_filt[:,None],Q_filt[:,None]))
139
140         f_line = 60
141
142         # threshold for minimum P_total or Q_total
143         min_on = 10
144
145         min_ind_between = 15 #events cannot be less 0.25 seconds apart
146
147         # window width for steady-state change calculations
148         ss_med = 0.2 # seconds
149         ss_med = int(round(ss_med*f_line)) # indices
150
151         if self.num_of_phases == 3:
152
153             dPA_filt = np.diff(PA_filt)
154             dPA_filt = np.append(dPA_filt, [0])
155
156             dPB_filt = np.diff(PB_filt)
157             dPB_filt = np.append(dPB_filt, [0])
158
159             dPC_filt = np.diff(PC_filt)
160             dPC_filt = np.append(dPC_filt, [0])
161
162             dP = np.hstack((dPA_filt[:,None],dPB_filt[:,None],dPC_filt[:,None
]))
163
164             dP_max_ind = np.argmax(dP,axis = 1)
165             dP_min_ind = np.argmin(dP,axis = 1)
166
167             dP_max = np.amax(dP,axis = 1)
168             dP_min = np.amin(dP,axis = 1)
169
170             ind = np.zeros(len(dPA_filt))
171             ind[dP_max > min_on] = 1
172             ind[dP_min < -min_on] = -1
173
174             # remove extra turn on indices
175             for i, j in enumerate(ind):
176                 if j == 1 or j == -1:
177                     ind[i+1:i+min_ind_between] = 0
178
179             ind_locations = np.nonzero(ind)[0]
180             ind = ind[ind_locations]
181             dP_max_ind = dP_max_ind[ind_locations]
182             dP_min_ind = dP_min_ind[ind_locations]

```

```

183         num_of_ind = len(ind)
184
185     elif self.num_of_phases == 1:
186
187         dP_filt = np.diff(P_filt)
188         dP = np.append(dP_filt, [0])
189
190         ind = np.zeros(len(dP))
191         ind[dP > min_on] = 1
192         ind[dP < -min_on] = -1
193
194         for i, j in enumerate(ind):
195             if j == 1 or j == -1:
196                 ind[i+1:i+min_ind_between] = 0
197
198         ind_locations = np.nonzero(ind)[0]
199         ind = ind[ind_locations]
200         num_of_ind = len(ind)
201
202     # pre-allocating
203     time_new = np.zeros(num_of_ind, dtype=time.dtype)
204     index_on = np.zeros(num_of_ind)
205     index_off = np.zeros(num_of_ind)
206     data_ss = np.zeros((num_of_ind, 2), dtype=float)
207     data_peak = np.zeros((1, 2), dtype=float)
208
209     good_window = 20 # plus/minus window for steady state value
210
211     for y in range(num_of_ind):
212         x = int(ind_locations[y])
213
214         # ON EVENTS #
215         if ind[y] == 1:
216
217             ind_start = x - ss_med
218             trans_end = x + 15
219             ind_end = trans_end + ss_med
220
221             if ind_start < 0:
222                 ind_start = 0
223
224             time_new[y] = time[x]
225
226             if self.num_of_phases==3:
227                 phase_ind = dP_max_ind[y]
228
229                 data_ss[y,0] = find_ss_on(PQ_filt[:, phase_ind*2], ind_start
230 , trans_end, ind_end, x)
231                 data_ss[y,1] = find_ss_on(PQ_filt[:, phase_ind*2+1],
232 ind_start, trans_end, ind_end, x)
233
234                 for i in range(self.num_of_loads):
235                     min_thres = self.power_levels[i] - good_window
236                     max_thres = self.power_levels[i] + good_window
237
238                     # Threshold for steady state power
239                     if data_ss[y,0] > min_thres and data_ss[y,0] <
240 max_thres:
241                         index_on[y] = self.index[i] + 1
242
243                     elif self.num_of_phases==1:
244                         data_ss[y,0] = find_ss_on(PQ_filt[:,0], ind_start, trans_end
245 , ind_end, x)
246                         data_ss[y,1] = find_ss_on(PQ_filt[:,1], ind_start, trans_end
247 , ind_end, x)
248
249                 for i in range(self.num_of_loads):
250                     min_thres = self.power_levels[i] - good_window

```

```

246         max_thres = self.power_levels[i] + good_window
247
248         # Threshold for steady state power
249         if data_ss[y,0] > min_thres and data_ss[y,0] <
max_thres:
250             index_on[y] = self.index[i] + 1
251
252         # OFF EVENTS #
253         if ind[y] == -1:
254
255             ind_start = x - ss_med
256             ind_end = x + ss_med
257
258             if ind_start < 0:
259                 ind_start = 0
260
261             time_new[y]= time[x]
262
263             if self.num_of_phases==3:
264
265                 phase_ind = dP_min_ind[y]
266
267                 data_ss[y,0] = find_ss_off(PQ_filt[:,phase_ind*2],
ind_start,ind_end,x)
268                 data_ss[y,1] = find_ss_off(PQ_filt[:,phase_ind*2+1],
ind_start,ind_end,x)
269
270                 for i in range(self.num_of_loads):
271                     min_thres = self.power_levels[i] - good_window
272                     max_thres = self.power_levels[i] + good_window
273
274                     # Threshold for steady state power
275                     if abs(data_ss[y,0]) > min_thres and abs(data_ss[y,0])
< max_thres:
276                         index_off[y] = self.index[i] + 1
277
278                     elif self.num_of_phases==1:
279
280                         data_ss[y,0] = find_ss_off(PQ_filt[:,0],ind_start,ind_end,
x)
281                         data_ss[y,1] = find_ss_off(PQ_filt[:,1],ind_start,ind_end,
x)
282
283                         for i in range(self.num_of_loads):
284                             min_thres = self.power_levels[i] - good_window
285                             max_thres = self.power_levels[i] + good_window
286
287                             # Threshold for steady state power
288                             if data_ss[y,0] > min_thres and data_ss[y,0] <
max_thres:
289                                 index_off[y] = self.index[i] + 1
290
291                 P_total = data_ss[:,0]
292                 Q_total = data_ss[:,1]
293
294
295                 # only keep if the steady-state is greater than the min threshold
296                 good_ind = np.where( (abs(P_total) > min_on) | (abs(Q_total) > min_on)
)
297
298                 P_total = P_total[good_ind]
299                 Q_total = Q_total[good_ind]
300                 S_total = np.sqrt(P_total**2 + Q_total**2)
301                 power_factor = P_total / S_total
302
303                 index_on = index_on[good_ind]
304                 index_off = index_off[good_ind]
305

```



```

306         load_ids = []
307         load_names = []
308
309         # look for existing Loads in sql database #
310         for load in db.query(Load).all():
311             load_ids.append(load.id)
312             load_names.append(load.name)
313
314
315         for i in range(0,self.num_of_loads):
316
317             load_on_ind = np.where( index_on == i + 1 )
318             load_off_ind = np.where( index_off == i + 1 )
319
320             num_on_events = np.count_nonzero(index_on == i + 1)
321             num_off_events = np.count_nonzero(index_off == i + 1)
322
323             # if load from applianace file does NOT exist in sql db, add it#
324             if self.load_name[i] not in load_names:
325                 pump = Load(name=self.load_name[i], type = Load.LoadType.
CYCLING)
326
327                 db.add(pump)
328
329             # checking load ids and load names in sql database #
330             for load in db.query(Load).all():
331                 if load.id not in load_ids:
332                     load_ids.append(load.id)
333                     load_names.append(load.name)
334
335             # setting load_id #
336             if self.load_name[i] in load_names:
337                 load_id = load_ids[load_names.index(self.load_name[i])]
338
339             # adding ON events to sqldb #
340             if num_on_events > 0:
341
342                 load_time_on = time_new[load_on_ind]
343                 on_times = [datetime.fromtimestamp(time*1e-6) for time in
load_time_on]
344
345                 load_S_on = S_total[load_on_ind]
346                 load_pf_on = power_factor[load_on_ind]
347                 on_states = np.ones(num_on_events)
348                 db.add_all(cycling_load_on(load_id=load_id,
349                                     power=load_S_on,
350                                     power_factor=load_pf_on,
351                                     num_runs=num_on_events,
352                                     on_times=on_times,
353                                     states = on_states))
354
355             # adding OFF events to sqldb #
356             if num_off_events > 0:
357                 load_time_off = time_new[load_off_ind]
358                 off_times = [datetime.fromtimestamp(time*1e-6) for time in
load_time_off]
359
360                 load_S_off = S_total[load_off_ind]
361                 load_pf_off = power_factor[load_off_ind]
362                 off_states = 2*np.ones(num_off_events)
363                 db.add_all(cycling_load_off(load_id=load_id,
364                                     power=load_S_off,
365                                     power_factor=load_pf_off,
366                                     num_runs=num_off_events,
367                                     off_times = off_times,
368                                     states = off_states))
369
370         db.commit()
371
372         stream_in.consume(len(sarray_in)-60)

```

```
371         db.close()
372     def stop(self):
373         self.stop_requested = True
374     if __name__ == "__main__":
375         r = Demo_Detector()
376         r.start()
```

C.11 NerdJack Code Changes

The code changes to make NerdJack compatible with the NILM software suite are documented here. Changes were made to `nerdjack.c` to add binary output mode for Nerdjack and to update the Nerdjack packet size for 12 channel transmission. For `nerdjack.c`, the first change was to add the int variables `NERDJACK_PACKET_SIZE = 2148` and `NERDJACK_NUM_SAMPLES = 1080`. The data is also declared as, `signed short data[1080]`.

```
31  int NERDJACK_PACKET_SIZE = 2168;
32  int NERDJACK_NUM_SAMPLES = 1080;
33
34  typedef struct __attribute__((packed)) {
35      unsigned char headerone;
36      unsigned char headertwo;
37      unsigned short packetNumber;
38      unsigned short adcused;
39      unsigned short packetsready;
40      signed short data[1080];
41  } dataPacket;
42
```

Then the unsigned short is changed to a `uint16_t`. Specifically, `unsigned short dataline[numChannels]` is changed to `uint16_t short dataline[numChannels]`

```
433  double voltline[numChannels];
434
435  uint16_t dataline[numChannels];
436
437  unsigned short packetsready = 0;
438  unsigned short adcused = 0;
439  unsigned short tempshort = 0;
440  int charsread = 0;
```

Now `NERDJACK_PACKET_SIZE = numChannelsSampled * 90` and `NERDJACK_NUM_SAMPLES = NERDJACK_NUM_SAMPLES * 2 + 8`.

```
442  int numgroupsProcessed = 0;
443  double volts;
444
445  NERDJACK_NUM_SAMPLES = numChannelsSampled * 90;
446  NERDJACK_PACKET_SIZE = NERDJACK_NUM_SAMPLES * 2 + 8;
```

An unsigned short is changed to a `uint16_t`. Specifically, `(unsigned short)(datapoint - INT16_MIN)` is changed to `(uint16_t)(datapoint - INT16_MIN)`

```

550     case CONVERT_HEX:
551     case CONVERT_DEC:
552         dataline[i] =
553             (uint16_t)(datapoint -
554                 INT16_MIN);
555     break;

```

Then, the binary output mode is added. To do this, the `goto bad` statement for the condition `if (printf("\n") < 0)` is removed from all `switch (convert)` cases. Instead it is added to the cases: `case CONVERT_VOLTS`, `case CONVERT_HEX`, and `case CONVERT_DEC` (these are all the cases there originally). Then we add the `case CONVERT_BINARY`.

```

561     switch (convert) {
562     case CONVERT_VOLTS:
563         for (i = 0; i < numChannels; i++) {
564             if (printf("%lf ", voltline[i])
565                 < 0)
566                 goto bad;
567         }
568     case CONVERT_HEX:
569         for (i = 0; i < numChannels; i++) {
570             if (printf("%04hX", dataline[i])
571                 < 0)
572                 goto bad;
573         }
574     case CONVERT_DEC:
575         for (i = 0; i < numChannels; i++) {
576             if (printf("%hu ", dataline[i])
577                 < 0)
578                 goto bad;
579         }
580     default:
581     case CONVERT_BINARY:
582         if (fwrite(dataline,
583                 sizeof(uint16_t), numChannels, stdout) < 0)
584             goto bad;
585         // }
586         break;
587     }
588 }
589
590
591
592
593
594
595
596
597
598
599
600
601
602

```

Listing C.4: `nerdjack_capture.py`: Added for the NerdJack as the analogous code of `ethstream_capture.py` for the LabJack.

```
1 import asyncio
2 import numpy as np
3 import logging
4 import sys
5
6 from joule.utilities import (timestamp_to_human,
7                             seconds_to_timestamp, time_now)
8
9 from .errors import CaptureError
10 from nilm.meter import meter
11
12 class NerdjackCapture():
13
14     def __init__(self, ip_address, filter_index):
15         # tunable constants
16         self.data_ts_inc = 1e6 / 8000.0 # 8KHz sampling
17
18         self.ip_address = ip_address
19         self.stop_requested = False
20
21         if(filter_index == None):
22             self.num_channels = 6
23         else:
24             self.num_channels = 7
25             self.filter_index = int(filter_index) + 6
26
27         self.ROW_BYTES = self.num_channels * 2 # num channels, 2 byte ints
28         self.ROWS_PER_BLOCK = 8000 # read ~8 blocks per second
29         self.BLOCK_SIZE = self.ROW_BYTES * self.ROWS_PER_BLOCK
30
31     async def run(self, output,
32                 max_gap=10,
33                 align=True,
34                 nrows=0):
35         process = await self._start_meter()
36         data_pipe = process.stdout
37         rows_processed = 0
38         # convert gap to microseconds
39         self.max_gap = seconds_to_timestamp(max_gap)
40         # set up timestamps
41         self.clock_ts = time_now()
42         self.data_ts = self.clock_ts
43         try:
44             while(not self.stop_requested):
45                 if(align):
46                     self._align_clock()
47                 data = await data_pipe.readexactly(self.BLOCK_SIZE)
48                 np_data = self._parse(data)
49                 await output.write(np_data)
50                 rows_processed += len(np_data)
51                 if(nrows > 0 and rows_processed >= nrows):
52                     self.stop_requested = True
53         except asyncio.streams.IncompleteReadError:
54             if(not self.stop_requested):
55                 logging.error("short read from ethstream, exiting")
56         finally:
57             try:
58                 process.terminate()
59             except ProcessLookupError:
60                 pass # already terminated ethstream proc
61         await output.close()
62
63     def stop(self):
```

```

64     sys.stderr.flush()
65     self.stop_requested = True
66
67 async def _start_meter(self):
68
69     if(self.num_channels == 6):
70         cmd = ["ethstream",
71              "-a", self.ip_address,
72              "-r", "8000",
73              "-C", "0,1,2,3,4,5",
74              "-N", "-B", "-o"]
75
76     elif(self.num_channels == 7):
77         cmd = ["ethstream",
78              "-a", self.ip_address,
79              "-r", "8000",
80              "-C", "0,1,2,3,4,5,%i"%self.filter_index,
81              "-N", "-B", "-o"]
82
83     create = asyncio.create_subprocess_exec(
84         *cmd,
85         stdin=asyncio.subprocess.DEVNULL,
86         stdout=asyncio.subprocess.PIPE)
87
88     return await create
89
90 def _parse(self, raw):
91     if len(raw) != self.BLOCK_SIZE:
92         raise CaptureError("short read from ethstream process")
93     # parse from string: 2 byte units in big endian
94     data = np.frombuffer(raw, dtype=np.uint16)
95     rows = len(raw) // self.ROW_BYTES
96     # give the array the proper shape
97     data.shape = (rows, self.num_channels)
98     # calculate timestamps
99     top_ts = self.data_ts + rows * self.data_ts_inc
100    ts = np.array(np.linspace(self.data_ts, top_ts,
101                             rows, endpoint=False), dtype=np.uint64)
102
103    ts.shape = (rows, 1)
104    self.data_ts = top_ts # update the timestamp for the next run
105
106    # add timestamps
107    ts_data = np.hstack((ts, data))
108    # now ts_data is the form
109    # [ts, d0, d1, d2, d3, ...]
110    # [ts, d0, d1, d2, d3, ...]
111    # ...
112    return ts_data
113
114 def _align_clock(self):
115     self.clock_ts = time_now()
116     if (self.data_ts - self.max_gap) > self.clock_ts:
117         print("Data is coming in too fast: data time "
118             "is %s but clock time is only %s" %
119             (timestamp_to_human(self.data_ts),
120              timestamp_to_human(self.clock_ts)))
121         exit(1) # can't recover
122
123     if (self.data_ts + self.max_gap) < self.clock_ts:
124         print("Skipping data timestamp forward from "
125             "%s to %s to match clock time\n" % (
126             timestamp_to_human(self.data_ts),
127             timestamp_to_human(self.clock_ts)))
128         self.data_ts = self.clock_ts
129
130 def close(self):
131     pass # nothing to do

```

Bibliography

- [1] J. Rogelj, D. Shindell, K. Jiang, S. Fifita, P. Foster, V. Ginzburg, C. Handa, H. Kheshgi, S. Kobayashi, E. Kriegler, L. Mundaca, R. S  ferian, and M. Vilarino, *Mitigation pathways compatible with 1.5  C in the context of sustainable development*. Intergovernmental Panel on Climate Change, 2018, pp. 93–174.
- [2] International Energy Agency, *Net Zero by 2050: A Roadmap for the Global Energy Sector*, 2021.
- [3] J. Paris, J. S. Donnal, R. Cox, and S. Leeb, “Hunting cyclic energy wasters,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2777–2786, 2014.
- [4] H. M. Hashemian and W. C. Bean, “State-of-the-art predictive maintenance techniques,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3480–3492, Oct 2011.
- [5] J. Paris, J. S. Donnal, and S. B. Leeb, “NilmDB: The non-intrusive load monitor database,” *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2459–2467, 2014.
- [6] D. H. Green, S. R. Shaw, P. Lindahl, T. J. Kane, J. S. Donnal, and S. B. Leeb, “A multiscale framework for nonintrusive load identification,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 992–1002, 2020.
- [7] W. Wichakool, Z. Remscrim, U. A. Orji, and S. B. Leeb, “Smart Metering of Variable Power Loads,” *IEEE Trans on Smart Grid*, vol. 6, no. 1, pp. 189–198, 2015.
- [8] O. Keysan and H. B. Ertan, “Real-time speed and position estimation using rotor slot harmonics,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 899–908, 2013.
- [9] J. Paris, J. S. Donnal, Z. Remscrim, S. B. Leeb, and S. R. Shaw, “The sinefit spectral envelope preprocessor,” *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4385–4394, 2014.
- [10] A. Abouljian, D. H. Green, J. F. Switzer, T. J. Kane, G. V. Bredariol, P. Lindahl, J. S. Donnal, and S. B. Leeb, “Nilm Dashboard: A power system monitor for electromechanical equipment diagnostics,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1405–1414, 2019.

- [11] Zeifman, M. and Roth, K., “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, Feb 2011.
- [12] J. A. Mueller and J. W. Kimball, “Accurate energy use estimation for nonintrusive load monitoring in systems of known devices,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2797–2808, 2018.
- [13] J. Zhang, X. Chen, W. W. Y. Ng, C. S. Lai, and L. L. Lai, “New appliance detection for nonintrusive load monitoring,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4819–4829, 2019.
- [14] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, “Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,” *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.
- [15] G. A. Raiker, S. Reddy B., U. Loganathan, S. Agrawal, A. S. Thakur, A. K., J. P. Barton, and M. Thomson, “Energy disaggregation using energy demand model and iot-based control,” *IEEE Transactions on Industry Applications*, vol. 57, no. 2, pp. 1746–1754, 2021.
- [16] F. M. Wittmann, J. C. López, and M. J. Rider, “Nonintrusive load monitoring algorithm using mixed-integer linear programming,” *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 180–187, 2018.
- [17] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, “An energy prediction approach for a nonintrusive load monitoring in home appliances,” *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 96–105, 2020.
- [18] M. Simonov, G. Chicco, and G. Zanetto, “Event-driven energy metering: Principles and applications,” *IEEE Transactions on Industry Applications*, vol. 53, no. 4, pp. 3217–3227, 2017.
- [19] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, “NILM redux: The case for emphasizing applications over accuracy,” in *Proc. 2nd Int. Non-Intrusive Appliance Load Monitor Workshop*, June 2014, pp. 1–4.
- [20] B. Celik and J. Vanschoren, “Adaptation strategies for automated machine learning on evolving data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3067–3078, 2021.
- [21] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, “Learning under concept drift: A review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
- [22] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Comput. Surv.*, vol. 46, no. 4, Mar. 2014.

- [23] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges, "BLUED: A fully labeled public dataset for event-based nonintrusive load monitoring research," in *Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability*, 2012, pp. 12–16.
- [24] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, no. 150007, 2015.
- [25] G. V. Bredariol, "The shipboard automatic watchstander (SAW): Utilization of nonintrusive load monitoring for shipboard automation," Master's thesis, Massachusetts Institute of Technology, June 2017.
- [26] P. A. Lindahl, D. H. Green, G. Bredariol, A. Aboulain, J. S. Donnal, and S. B. Leeb, "Shipboard fault detection through nonintrusive load monitoring: A case study," *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8986–8995, 2018.
- [27] T. J. Kane, "The NILM Dashboard: Shipboard automatic watchstanding and real-time fault detection using non-intrusive load monitoring," Master's thesis, Massachusetts Institute of Technology, June 2019.
- [28] W. Wichakool, A.-T. Avestruz, R. W. Cox, and S. B. Leeb, "Modeling and estimating current harmonics of variable electronic loads," *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 2803–2811, 2009.
- [29] A. W. Langham, D. H. Green, and S. B. Leeb, "Resolution analysis for power system measurement and transient identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.
- [30] A. V. Oppenheim and R. W. Schaffer, *Discrete-time signal processing*, 3rd ed. Upper Saddle River, NJ: Pearson, 2010.
- [31] R. Skartlien and L. Oyehaug, "Quantization error and resolution in ensemble averaged data with noise," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1303–1312, 2005.
- [32] *Fundamentals of Precision ADC Noise Analysis*, Texas Instruments, September 2020. [Online]. Available: <https://www.ti.com/lit/pdf/SLYY192>
- [33] D. Kumar and F. Zare, "A comprehensive review of maritime microgrids: System architectures, energy efficiency, power quality, and regulations," *IEEE Access*, vol. 7, pp. 67 249–67 277, 2019.
- [34] A. Testa *et al.*, "Interharmonics: Theory and modeling," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2335–2348, 2007.
- [35] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, pp. 162–165, 1964.

- [36] “IEEE standard for digitizing waveform recorders,” *IEEE Std 1057-2017 (Revision of IEEE Std 1057-2007)*, pp. 1–0, 2018.
- [37] M. Loève, “Probability theory I,” *Graduate texts in mathematics*, vol. 45, p. 12, 1977.
- [38] E. O. Muñiz, “A method for deriving various formulas in electrostatics and electromagnetism using Lagrange’s trigonometric identities,” *American Journal of Physics*, vol. 21, no. 2, pp. 140–141, 1953.
- [39] “IEEE standard for terminology and test methods for analog-to-digital converters,” *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)*, pp. 1–139, 2011.
- [40] P. Sysel and P. Rajmic, “Goertzel algorithm generalized to non-integer multiples of fundamental frequency,” *EURASIP Journal on Advances in Signal Processing*, vol. 2012, 12 2012.
- [41] S. Qin, Z. Huang, and X. Wang, “Optical angular encoder installation error measurement and calibration by ring laser gyroscope,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 3, pp. 506–511, 2010.
- [42] S.-T. Wu, J.-Y. Chen, and S.-H. Wu, “A rotary encoder with an eccentrically mounted ring magnet,” *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 8, pp. 1907–1915, 2014.
- [43] R. M. Kennel, “Encoders for simultaneous sensing of position and speed in electrical drives with digital control,” *IEEE Transactions on Industry Applications*, vol. 43, no. 6, pp. 1572–1577, 2007.
- [44] Z. Gao, L. Turner, R. S. Colby, and B. Leprettre, “A frequency demodulation approach to induction motor speed detection,” *IEEE Transactions on Industry Applications*, vol. 47, no. 4, pp. 1632–1642, 2011.
- [45] A. Bellini, G. Franceschini, and C. Tassoni, “Monitoring of induction machines by maximum covariance method for frequency tracking,” *IEEE Transactions on Industry Applications*, vol. 42, no. 1, pp. 69–78, 2006.
- [46] M. Sahraoui, A. J. M. Cardoso, K. Yahia, and A. Ghoggal, “The use of the modified prony’s method for rotor speed estimation in squirrel-cage induction motors,” *IEEE Transactions on Industry Applications*, vol. 52, no. 3, pp. 2194–2202, 2016.
- [47] U. A. Orji, Z. Remscrim, C. Schantz, J. Donnal, J. Paris, M. Gillman, K. Surakitbovorn, S. B. Leeb, and J. L. Kirtley, “Non-intrusive induction motor speed detection,” *IET Electric Power Applications*, vol. 9, no. 5, pp. 388–396, 2015.

- [48] M. El Hachemi Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 984–993, 2000.
- [49] K. Lee, L. Huchel, D. H. Green, and S. B. Leeb, "Automatic power frequency rejection instrumentation for nonintrusive frequency signature tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
- [50] S. Nandi, S. Ahmed, and H. A. Toliyat, "Detection of rotor slot and other eccentricity related harmonics in a three phase induction motor with different rotor cages," *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 253–260, 2001.
- [51] Labjack Corporation, *UE9 Datasheet*, Lakewood, Colorado, 2020.
- [52] J. Donnal, "Joule: A real-time framework for decentralized sensor networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3615–3623, 2018.
- [53] J. S. Donnal, "Wattsworth: An open-source platform for decentralized sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 189–196, 2020.
- [54] J. Park, J. Candelaria, L. Ma, and K. Dunn, "Dc ring-bus microgrid fault protection and identification of fault location," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2574–2584, 2013.
- [55] S. Mudaliyar and S. Mishra, "Coordinated voltage control of a grid connected ring dc microgrid with energy hub," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1939–1948, 2019.
- [56] M. Salehi, S. A. Taher, I. Sadeghkhani, and M. Shahidehpour, "A poverty severity index-based protection strategy for ring-bus low-voltage dc microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6860–6869, 2019.
- [57] B. Stevens, A. Dubey, and S. Santoso, "On improving reliability of shipboard power system," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1905–1912, 2015.
- [58] T. S. Ustun, C. Ozansoy, and A. Ustun, "Fault current coefficient and time delay assignment for microgrid protection system with central protection unit," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 598–606, 2013.
- [59] S. A. Saleh, "Signature-coordinated digital multirelay protection for microgrid systems," *IEEE Transactions on Power Electronics*, vol. 29, no. 9, pp. 4614–4623, 2014.
- [60] M. A. Haj-ahmed and M. S. Illindala, "Investigation of protection schemes for flexible distribution of energy and storage resources in an industrial microgrid," *IEEE Transactions on Industry Applications*, vol. 51, no. 3, pp. 2071–2080, 2015.

- [61] M. W. Rose and R. M. Cuzner, "Fault isolation and reconfiguration in a three-zone system," in *2015 IEEE Electric Ship Technologies Symposium (ESTS)*, 2015, pp. 409–414.
- [62] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox, "Nonintrusive load monitoring and diagnostics in power systems," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 7, pp. 1445–1454, 2008.
- [63] A. Derviškić, R. Razzaghi, Q. Walger, and M. Paolone, "The white rabbit time synchronization protocol for synchrophasor networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 726–738, 2020.
- [64] A. Mahmood, M. I. Ashraf, M. Gidlund, J. Torsner, and J. Sachs, "Time synchronization in 5g wireless edge: Requirements and solutions for critical-mtc," *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 45–51, 2019.
- [65] M. Lévesque and D. Tipper, "A survey of clock synchronization over packet-switched networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2926–2947, 2016.
- [66] T.-C. Lin, H.-H. Wang, and C.-H. Liu, "Fault location for three-ended ring-topology power system using minimum gps-based measurements and cvt/ct sensing," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 22 019–22 031, 2021.
- [67] Y. Yuan, K. Dehghanpour, and Z. Wang, "Mitigating smart meter asynchrony error via multi-objective low rank matrix recovery," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4308–4317, 2021.
- [68] A. Alimardani, F. Therrien, D. Atanackovic, J. Jatskevich, and E. Vaahedi, "Distribution system state estimation based on nonsynchronized smart meters," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2919–2928, 2015.
- [69] X. Zeng, K. Liu, J. Ma, M. Chen, and M. Yu, "Reliability and delay trade-off analysis of unslotted ieee 802.15.4 sensor network for shipboard environment," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 2400–2411, 2021.
- [70] M. Chen, K. Liu, J. Ma, and C. Liu, "Spatio-temporal fingerprint localization for shipboard wireless sensor networks," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10 125–10 133, 2018.
- [71] B. T. Mills, D. H. Green, J. S. Donnal, and S. B. Leeb, "Power monitoring beyond radial distribution networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.
- [72] I. Badran, M. T. Lazim, and M. Zeidan, "A general solution for ring-bus distribution systems reliability," in *2008 5th International Multi-Conference on Systems, Signals and Devices*, 2008, pp. 1–4.

- [73] G. Lipardi, L. Piva, L. Piegari, E. Tironi, R. Lamedica, A. Ruvio, G. Sulligoi, and A. Vicenzutti, “Electric loads characterization in an aircraft carrier with ring-bus distribution system,” in *2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)*, 2015, pp. 1–6.
- [74] U. Orji, B. Sievenpiper, K. Gerhard, S. B. Leeb, N. Doerry, J. L. Kirtley, and T. McCoy, “Load modeling for power system requirement and capability assessment,” *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1415–1423, 2015.
- [75] “IEEE recommended practice for electrical installations on shipboard—design,” *IEEE Std 45.1-2017*, pp. 1–198, 2017.
- [76] W. C. Greene, “Evaluation of non-intrusive monitoring for condition based maintenance applications on us navy propulsion plants,” Master’s thesis, Massachusetts Institute of Technology, 2005.
- [77] J. T. Leghorn, “Modeling for ship power system emulation,” Master’s thesis, Massachusetts Institute of Technology, 2009.
- [78] T. L. Hannon, “Integrated circuit breaker protection software,” U.S. Patent US6 694 271B1, 2001.
- [79] M. Azaria and D. Hertz, “Time delay estimation by generalized cross correlation methods,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 280–285, 1984.
- [80] R. C. Schaefer, “Art of generator synchronizing,” *IEEE Transactions on Industry Applications*, vol. 53, no. 1, pp. 751–757, 2017.
- [81] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar, “Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,” *Sensors*, vol. 12, no. 12, p. 16838–16866, Dec 2012.
- [82] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Improving nonintrusive load monitoring efficiency via a hybrid programming method,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2148–2157, Dec 2016.
- [83] J. Pan, C. Li, Y. Tang, W. Li, and X. Li, “Energy consumption prediction of a cnc machining process with incomplete data,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 987–1000, 2021.
- [84] W. C. Karl, S. B. Leeb, L. A. Jones, J. L. Kirtley, and G. C. Verghese, “Applications of rank-based median filters in power electronics,” *IEEE Transactions on Power Electronics*, vol. 7, no. 3, pp. 437–443, 1992.
- [85] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York, NY, USA: McGraw-Hill, Inc., 1995.

- [86] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [87] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1545–1554.
- [88] K. Lee, S. Leeb, L. Norford, P. Armstrong, J. Holloway, and S. Shaw, "Estimation of variable-speed-drive power consumption from harmonic content," *IEEE Transactions on Energy Conversion*, vol. 20, no. 3, pp. 566–574, 2005.
- [89] E. K. Saathoff, D. H. Green, R. A. Agustin, J. W. O'Connell, and S. B. Leeb, "Inrush current measurement for transient space characterization and fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [90] P. Yin and M. B. Pate, "Impact of duct flow resistance on residential heating and cooling energy use in systems with PSC and ECM blowers," *Energy and Buildings*, vol. 130, pp. 625–636, 2016.
- [91] B. Stephens, A. Novoselac, and J. A. Siegel, "The effects of filtration on pressure drop and energy consumption in residential HVAC systems (rp-1299)," *HVAC&R Research*, vol. 16, no. 3, pp. 273–294, 2010.
- [92] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 175–188, 2015.
- [93] B. Cannas, S. Carcangiu, D. Carta, A. Fanni, and C. Muscas, "Selection of features based on electric power quantities for non-intrusive load monitoring," *Applied Sciences*, vol. 11, no. 2, 2021.
- [94] R. Chen, C. Dewi, S. Huang, and R. Caraka, "Selecting critical features for data classification based on machine learning methods," *Journal Of Big Data*, vol. 7, p. 26, 07 2020.
- [95] K. Mao, "Rbf neural network center selection based on fisher ratio class separability measure," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1211–1217, 2002.
- [96] L. Wang, "Feature selection with kernel class separability," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1534–1546, 2008.
- [97] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

- [98] R. Agustin, “A Load Identification and Diagnostic Framework for Aggregate Power Monitoring,” Master’s thesis, Massachusetts Institute of Technology, 2021.
- [99] M. M. Suarez-Alvarez, D.-T. Pham, M. Y. Prostov, and Y. I. Prostov, “Statistical approach to normalization of feature vectors and clustering of mixed datasets,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 468, no. 2145, pp. 2630–2651, 2012.
- [100] B. Wang, W. Shi, and Z. Miao, “Confidence analysis of standard deviational ellipse and its extension into higher dimensional Euclidean space,” *PLOS ONE*, vol. 10, no. 3, pp. 1–17, 03 2015.
- [101] J. M. Shapiro, G. B. Lamont, and G. L. Peterson, “An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection,” in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’05. New York, NY, USA: Association for Computing Machinery, 2005, p. 337–344.
- [102] I. T. Jolliffe, *Principal Component Analysis and Factor Analysis*. New York, NY: Springer New York, 1986, pp. 115–128.
- [103] I. Gilitschenski and U. D. Hanebeck, “A robust computational test for overlap of two arbitrary-dimensional ellipsoids in fault-detection of Kalman filters,” in *2012 15th International Conference on Information Fusion*, 2012, pp. 396–401.
- [104] —, “A direct method for checking overlap of two hyperellipsoids,” in *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2014, pp. 1–6.
- [105] B. Zhang, Y. Guo, Y. Li, Y. He, H. Wang, and Q. Dai, “Memory recall: A simple neural network training framework against catastrophic forgetting,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.
- [106] S. Liu, S. Xue, J. Wu, C. Zhou, J. Yang, Z. Li, and J. Cao, “Online active learning for drifting data streams,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.
- [107] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [108] B. Celik and J. Vanschoren, “Adaptation strategies for automated machine learning on evolving data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3067–3078, 2021.
- [109] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, “Context aware energy disaggregation using adaptive bidirectional

- LSTM models,” *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020.
- [110] Y. Wang, L. Zhang, Y. Yao, and Y. Fu, “How to trust unlabeled data instance credibility inference for few-shot learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [111] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.
- [112] X. Pu and C. Li, “Online semisupervised broad learning system for industrial fault diagnosis,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6644–6654, 2021.
- [113] R. N. Gemaque, A. F. J. Costa, R. Giusti, and E. Santos, “An overview of unsupervised drift detection methods,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, 2020.
- [114] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. a. Gama, “Machine learning for streaming data: State of the art, challenges, and opportunities,” *SIGKDD Explor. Newsl.*, vol. 21, no. 2, p. 6–22, Nov. 2019.
- [115] H. Hu, M. Kantardzic, and T. S. Sethi, “No free lunch theorem for concept drift detection in streaming data classification: A review,” *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, 2020.
- [116] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with drift detection,” in *Brazilian symposium on artificial intelligence*. Springer, 2004, pp. 286–295.
- [117] M. Baena-Garcia, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldá, and R. Morales-Bueno, “Early drift detection method,” in *Fourth international workshop on knowledge discovery from data streams*, vol. 6, 2006, pp. 77–86.
- [118] S. Wang, L. L. Minku, and X. Yao, “A systematic study of online class imbalance learning with concept drift,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4802–4821, 2018.
- [119] A. Haque, L. Khan, and M. Baron, “Sand: Semi-supervised adaptive novel class detection and classification over data stream,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. AAAI Press, 2016, p. 1652–1658.
- [120] F. A. Pinage, E. M. dos Santos, and J. Gama, “A drift detection method based on dynamic classifier selection,” *Data Mining and Knowledge Discovery*, vol. 34, pp. 50–74, 2019.

- [121] K. B. Dyer, R. Capo, and R. Polikar, “Compose: A semisupervised learning framework for initially labeled nonstationary streaming data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 12–26, 2014.
- [122] R. Capo, A. Sanchez, and R. Polikar, “Core support extraction for learning from initially labeled nonstationary environments using compose,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 602–608.
- [123] M. Umer, C. Frederickson, and R. Polikar, “Learning under extreme verification latency quickly: Fast compose,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–8.
- [124] R. Razavi-Far, E. Hallaji, M. Saif, and G. Ditzler, “A novelty detector and extreme verification latency model for nonstationary environments,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 561–570, 2019.
- [125] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *NIPS 2014 Workshop on Deep Learning*, 2014.
- [126] J. Peng, B. Tang, H. Jiang, Z. Li, Y. Lei, T. Lin, and H. Li, “Overcoming long-term catastrophic forgetting through adversarial neural pruning and synaptic consolidation,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.
- [127] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.
- [128] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li, “Memory efficient class-incremental learning for image classification,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2021.
- [129] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, “Overcoming catastrophic forgetting in neural networks,” *Proc. Nat. Acad. Sci. USA*, vol. 144, no. 13, pp. 3521–3526, 2017.
- [130] A. Mallya and S. Lazebnik, “Packnet: Adding multiple tasks to a single network by iterative pruning,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.
- [131] R. A. Agustin, “A load identification and diagnostic framework for aggregate power monitoring,” Master’s thesis, Massachusetts Institute of Technology, Feb 2021.

- [132] R. Hyde, P. Angelov, and A. MacKenzie, “Fully online clustering of evolving data streams into arbitrarily shaped clusters,” *Information Sciences*, vol. 382–383, pp. 96 – 114, 2017.
- [133] M. Tareq, E. A. Sundararajan, M. Mohd, and N. S. Sani, “Online clustering of evolving data streams using a density grid-based method,” *IEEE Access*, vol. 8, pp. 166 472–166 490, 2020.
- [134] Y.-F. Li and Z.-H. Zhou, “Towards making unlabeled data never hurt,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 175–188, 2015.
- [135] S. Makonin and F. Popowich, “Nonintrusive Load Monitoring Performance Evaluation,” *Energy Efficiency*, vol. 8, pp. 809–814, 12 2014.
- [136] O. Kramer, *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Springer Berlin Heidelberg, 2013, ch. K-Nearest Neighbors, pp. 13–23.
- [137] T. Siau and A. Bayen, *An Introduction to MATLAB Programming and Numerical Methods for Engineers*, 1st ed. USA: Academic Press, Inc., 2014.
- [138] H. S. Kang, J. Y. Lee, and D. Y. Lee, “An integrated energy data analytics approach for machine tools,” *IEEE Access*, vol. 8, pp. 56 124–56 140, 2020.
- [139] H. Liu and V. Makis, “Cutting-tool reliability assessment in variable machining conditions,” *IEEE Transactions on Reliability*, vol. 45, no. 4, pp. 573–581, 1996.
- [140] B. Gustavsen, M. Høyer-Hansen, M. Hatlo, and S. Midttveit, “Voltages and ac corrosion on metallic tubes in umbilical cables caused by magnetic induction from power cable charging currents,” *IEEE Transactions on Power Delivery*, vol. 34, no. 2, pp. 596–605, 2019.
- [141] S. Samarasinghe, C. Ekanayake, H. Ma, T. K. Saha, J. Baniya, D. Allan, and G. Russel, “A risk assessment for utilities to prevent transformer oltc failures caused by silver sulphide corrosion,” *IEEE Transactions on Power Delivery*, pp. 1–1, 2021.
- [142] J. W. O’Connell, D. H. Green, B. T. Mills, A. Moeller, S. Kidwell, K. Lee, L. Huchel, and S. B. Leeb, “Nonintrusive ventilation system diagnostics,” *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 268–19 278, 2021.
- [143] A. Singh, Y. Pandey, A. Kumar, M. K. Singh, A. Kumar, and S. C. Mukhopadhyay, “Ventilation monitoring and control system for high rise historical buildings,” *IEEE Sensors Journal*, vol. 17, no. 22, pp. 7533–7541, 2017.
- [144] A. Verma, S. Prakash, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, “Sensing, controlling, and IoT infrastructure in smart building: A review,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9036–9046, 2019.

- [145] A. Kumar, A. Singh, A. Kumar, M. K. Singh, P. Mahanta, and S. C. Mukhopadhyay, "Sensing technologies for monitoring intelligent buildings: A review," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 4847–4860, 2018.
- [146] Y. Yan, P. B. Luh, and K. R. Pattipati, "Fault diagnosis of components and sensors in HVAC air handling systems with new types of faults," *IEEE Access*, vol. 6, pp. 21 682–21 696, 2018.
- [147] P. R. Armstrong, C. R. Laughman, S. B. Leeb, and L. K. Norford, "Detection of rooftop cooling unit faults based on electrical measurements," *HVAC&R Research*, vol. 12, no. 1, pp. 151–175, 2006.
- [148] C. Laughman, "Fault detection methods for vapor-compression air conditioners using electrical measurements," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.
- [149] B. B. Daly and W. C. Osborne, *Wood's Practical Guide to Fan Engineering*. Woods Air Movement, Limited, 1978.
- [150] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.
- [151] A. Movement and C. Association, *AMCA 210: Laboratory Methods of Testing*. AMCA, 1999.
- [152] A. C. of Governmental Industrial Hygienists, *Industrial Ventilation, A Manual Of Recommended Practice*. Committee on Industrial Ventilation, 1998.
- [153] A. Khezzar, M. Y. Kaikaa, M. E. K. Oumaamar, M. Boucherma, and H. Razik, "On the use of slot harmonics as a potential indicator of rotor bar breakage in the induction machine," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4592–4605, 2009.
- [154] M. Drif and A. J. Marques Cardoso, "Discriminating the simultaneous occurrence of three-phase induction motor rotor faults and mechanical load oscillations by the instantaneous active and reactive power media signature analyses," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1630–1639, 2012.
- [155] M. Chirindo, M. A. Khan, and P. Barendse, "Analysis of non-intrusive rotor speed estimation techniques for inverter-fed induction motors," *IEEE Transactions on Energy Conversion*, vol. 36, no. 1, pp. 338–347, 2021.
- [156] H. W. Beaty and J. L. Kirtley, *Electric Motor Handbook*. McGraw-Hill, 1998.
- [157] S. J. Chapman, *Electric machinery fundamentals*. McGraw-Hill, 2012.

- [158] B. Stephens, “The impacts of duct design on life cycle costs of central residential heating and air-conditioning systems,” *Energy and Buildings*, vol. 82, pp. 563–579, 2014.
- [159] A. Khezzar, M. Y. Kaikaa, and M. Boucherma, “Analytical investigation of rotor slot harmonics in a three phase induction motor with broken rotor bars,” in *2005 European Conference on Power Electronics and Applications*, 2005, pp. 10 pp.–P.10.
- [160] G. H. Muller and C. F. Landy, “A novel method to detect broken rotor bars in squirrel cage induction motors when interbar currents are present,” *IEEE Transactions on Energy Conversion*, vol. 18, no. 1, pp. 71–79, 2003.
- [161] H. Hwang and S. Kang, “Nonintrusive load monitoring using a lstm with feedback structure,” *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2022.
- [162] G. Karniadakis, Y. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, pp. 422–440, 06 2021.
- [163] K. Li, J. Liu, Z. Wang, and B. Wei, “Strategies and operating point optimization of statcom control for voltage unbalance mitigation in three-phase three-wire systems,” *IEEE Transactions on Power Delivery*, vol. 22, no. 1, pp. 413–422, 2007.
- [164] I. M. Karmacharya and R. Gokaraju, “Fault location in ungrounded photovoltaic system using wavelets and ann,” *IEEE Transactions on Power Delivery*, vol. 33, no. 2, pp. 549–559, 2018.
- [165] A. Vukojevic and S. Lukic, “Microgrid protection and control schemes for seamless transition to island and grid synchronization,” *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 2845–2855, 2020.
- [166] D. P. Kothari and I. J. Nagrath, *Modern power system analysis*. Tata McGraw-Hill Pub. Co., 2008.
- [167] J. Willems, “Reflections on apparent power and power factor in nonsinusoidal and polyphase situations,” *IEEE Transactions on Power Delivery*, vol. 19, no. 2, pp. 835–840, 2004.
- [168] “IEEE standard definitions for the measurement of electric power quantities under sinusoidal, nonsinusoidal, balanced, or unbalanced conditions,” *IEEE Std 1459-2010 (Revision of IEEE Std 1459-2000)*, pp. 1–50, 2010.
- [169] A. Hamel, A. Gaudreau, and M. Cote, “Intermittent arcing fault on underground low-voltage cables,” *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1862–1868, 2004.

- [170] A. van Deursen, P. Wouters, and F. Steennis, "Corrosion in low-voltage distribution networks and perspectives for online condition monitoring," *IEEE Transactions on Power Delivery*, vol. 34, no. 4, pp. 1423–1431, 2019.
- [171] D. Paul, "Phase-ground fault current analysis and protection of a high-resistance grounded power system," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 3306–3314, 2020.
- [172] T. Baldwin and F. Renovich, "Analysis of fault locating signals for high-impedance grounded systems," *IEEE Transactions on Industry Applications*, vol. 38, no. 3, pp. 810–817, 2002.
- [173] U. Orji, C. Schantz, S. B. Leeb, J. L. Kirtley, B. Sievenpiper, K. Gerhard, and T. McCoy, "Adaptive zonal protection for ring microgrids," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1843–1851, 2017.
- [174] W. Liu, T. Tarasiuk, M. Gorniak, M. Savaghebi, J. C. Vasquez, C.-L. Su, and J. M. Guerrero, "Power quality assessment in shipboard microgrids under unbalanced and harmonic ac bus voltage," *IEEE Transactions on Industry Applications*, vol. 55, no. 1, pp. 765–775, 2019.
- [175] R. Yan and T. K. Saha, "Voltage variation sensitivity analysis for unbalanced distribution networks due to photovoltaic power fluctuations," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1078–1089, 2012.
- [176] M. Grotzbach and R. Redmann, "Line current harmonics of vsi-fed adjustable-speed drives," *IEEE Transactions on Industry Applications*, vol. 36, no. 2, pp. 683–690, 2000.
- [177] A. von Jouanne and B. Banerjee, "Assessment of voltage unbalance," *IEEE Transactions on Power Delivery*, vol. 16, no. 4, pp. 782–790, 2001.
- [178] C. F. Nascimento, E. H. Watanabe, O. Diene, A. B. Dietrich, A. Goedel, J. J. C. Gyselinck, and R. F. S. Dias, "Analysis of noncharacteristic harmonics generated by voltage-source converters operating under unbalanced voltage," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 951–961, 2017.
- [179] V. V. Sourabh Dash, "Challenges in the industrial applications of fault diagnostic systems," *Computers and Chemical Engineering*, vol. 24, pp. 785–791, 2000.
- [180] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, Inc., 2009.
- [181] S. Rao, *Reliability-Based Design*. McGraw-Hill, 1992.
- [182] P. Embrechts and M. Hofert, "A note on generalized inverses," *Mathematical Methods of Operations Research*, vol. 77, no. 3, pp. 423–432, 2013.

- [183] G. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [184] M. D. Gillman, J. S. Donnal, J. Paris, S. B. Leeb, M. A. H. E. Sayed, K. Wertz, and S. Schertz, “Energy accountability using nonintrusive load monitoring,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1923–1931, 2014.
- [185] P. A. Lindahl, M. T. Ali, P. Armstrong, A. Aboulian, J. Donnal, L. Norford, and S. B. Leeb, “Nonintrusive load monitoring of variable speed drive cooling systems,” *IEEE Access*, vol. 8, pp. 211 451–211 463, 2020.
- [186] S. B. Kidwell, “Shipboard fault detection, load transient exploration, and power simulation,” Master’s thesis, Massachusetts Institute of Technology, May 2020.
- [187] J. W. O’Connell, “Shipboard fault detection, marine micro-grid power diagnostics and vessel ventilation monitoring,” Master’s thesis, Massachusetts Institute of Technology, June 2021.
- [188] B. T. Mills, “Solving time-alignment challenges in shipboard non-intrusive load monitoring,” Master’s thesis, Massachusetts Institute of Technology, June 2021.
- [189] T. Deeter, D. H. Green, S. Kidwell, T. J. Kane, J. S. Donnal, K. Vasquez, B. Sievenpiper, and S. B. Leeb, “Behavioral modeling for microgrid simulation,” *IEEE Access*, vol. 9, pp. 35 633–35 645, 2021.
- [190] C. A. Reusser, H. A. Young, J. R. Perez Osses, M. A. Perez, and O. J. Simmonds, “Power electronics and drives: Applications to modern ship propulsion systems,” *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 106–122, 2020.
- [191] J. Ge, F. Yu, T. Tomizawa, H. Song, and N. Yusa, “Inspection of pitting corruptions on weld overlay cladding using uniform and rotating eddy current testing,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [192] D. C. Sweeney, A. M. Schrell, and C. M. Petrie, “Pressure-driven fiber-optic sensor for online corrosion monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [193] J. R. Gallion and R. Zoughi, “Millimeter-wave imaging of surface-breaking cracks in steel with severe surface corrosion,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 10, pp. 2789–2791, 2017.
- [194] D. Green, T. Kane, S. Kidwell, P. Lindahl, J. Donnal, and S. Leeb, “NILM Dashboard: Actionable feedback for condition-based maintenance,” *IEEE Instrumentation and Measurement Magazine*, vol. 23, no. 5, pp. 3–10, 2020.
- [195] “Intergranular stress corrosion cracking of copper in nitrite solutions,” *Corrosion Science*, vol. 47, no. 9, pp. 2099–2124, 2005.

- [196] M. B. Kannan and P. Shukla, "Stress corrosion cracking (scc) of copper and copper-based alloys," in *Stress Corrosion Cracking*, ser. Woodhead Publishing Series in Metals and Surface Engineering, V. Raja and T. Shoji, Eds. Woodhead Publishing, 2011, pp. 409–426.
- [197] S. H. Crandall, N. C. Dahl, and T. J. Lardner, *An Introduction to the Mechanics of Solids*, 2nd ed. McGraw-Hill, 1999.
- [198] R. D. Blevins, *Formulas for Natural Frequency and Mode Shape*, 1st ed. Van Nostrand Reinhold Co, 1977.
- [199] L. Huchel, T. C. Krause, T. Ługowski, S. B. Leeb, and J. Helsen, "Chasing the cut: A measurement approach for machine tool condition monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [200] A. Omekanda, Z. Tang, and P. Pillay, "Vibration prediction in switched reluctance motor with transfer function identification from shaker and force hammer tests," *IEEE Transactions on Industry Applications*, vol. 39, no. 4, pp. 978–985, 2003.