Modern Control Methods for Chemical Process Systems

by

Joel Anthony Paulson

B.S., University of Texas at Austin (2011)
M.S.CEP, Massachusetts Institute of Technology (2013)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

Author ..... Signature redacted ....
Department of Chemical Engineering
September 16, 2016

Certified by ..... Signature redacted .....
Richard D. Braatz
Edward R. Gilliland Professor in Chemical Engineering
Thesis Supervisor

Certified by . Signature redacted .....
Michael S. Strano
Carbon P. Dubbs Professor in Chemical Engineering
Thesis Supervisor

Accepted by . Signature redacted .....
Daniel Blankschtein
Herman P. Meissner Professor in Chemical Engineering
Chairman, Committee for Graduate Students

Modern Control Methods for Chemical Process Systems

by

Joel Anthony Paulson

Submitted to the Department of Chemical Engineering
on September 16, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Strong trends in chemical engineering have led to increased complexity in plant design and operation, which has driven the demand for improved control techniques and methodologies. Improved control directly leads to smaller usage of resources, increased productivity, improved safety, and reduced pollution.

Model predictive control (MPC) is the most advanced control technology widely practiced in industry. This technology, initially developed in the chemical engineering field in the 1970s, was a major advance over earlier multivariable control methods due to its ability to seamlessly handle constraints. However, limitations in industrial MPC technology spurred significant research over the past two to three decades in the search of increased capability. For these advancements to be widely implemented in industry, they must adequately address all of the issues associated with control design while meeting all of the control system requirements including:

- The controller must be insensitive to uncertainties including disturbances and unknown parameter values.
- The controlled system must perform well under input, actuator, and state constraints.
- The controller should be able to handle a large number of interacting variables efficiently as well as nonlinear process dynamics.
- The controlled system must be safe, reliable, and easy to maintain in the presence of system failures/faults.

This thesis presents a framework for addressing these problems in a unified manner. Uncertainties and constraints are handled by extending current state-of-the-art MPC methods to handle probabilistic uncertainty descriptions for the unknown parameters and disturbances. Sensor and actuator failures (at the regulatory layer) are handled using a specific internal model control structure that allows for the regulatory control layer to perform optimally whenever one or more controllers is taken offline due to failures. Non-obvious faults, that may lead to catastrophic system failure if not detected

early, are handled using a model-based active fault diagnosis method, which is also able to cope with constraints and uncertainties. These approaches are demonstrated on industrially relevant examples including crystallization and bioreactor processes.

Thesis Supervisor: Richard D. Braatz
Title: Edward R. Gilliland Professor in Chemical Engineering

Thesis Supervisor: Michael S. Strano
Title: Carbon P. Dubbs Professor in Chemical Engineering

Acknowledgments

I am extremely grateful for my Ph.D. advisors Richard Braatz and Michael Strano for their continued support and guidance. I also thank them for allowing me the freedom to explore many different research topics in addition to exposing me to a number of interesting projects in the fields of modeling, simulation, and control. In particular, I spent countless hours discussing a plethora of topics with my main advisor Professor Richard Braatz. He not only taught me about many facets of chemical engineering and control theory, but also how to be a better communicator, presenter, and researcher. We often discussed the broader areas of academia, industry, and life from which I learned so much and will be forever grateful.

I would also like to thank my thesis committee members, Bassam Alfeeli, George Stephanopoulos, and William Tisdale, for their insightful comments and encouragement. They pushed me to pursue my research interests and often suggested ways to frame my work so that it was accessible to a chemical engineering audience.

I thoroughly enjoyed the many courses that I took during my graduate studies. I thank Martin Bazant and William Deen for teaching me transport phenomena, Arup Chakraborty and Brad Olsen for classical and statistical thermodynamics, Paul Barton and Joe Scott for numerical methods, George Stephanopoulos and Richard Braatz for systems engineering and control theory, Jeffrey Shapiro and Alan Oppenheim for communication and signal processing, and Peter Hagelstein for signals and systems. I would also like to thank Richard Braatz, Elizabeth Lee, and James Swan for their support, advice, and patience while I was a teaching assistant for the graduate numerical methods course. Without their help, I would not have won the School of Engineering Graduate Student Award for Extraordinary Teaching and Mentoring.

Thanks to the Braatz group members for their help and support over the past five years with special thanks to those I worked with personally including Lucas Foguth, Eranda Harinath, Amos Lu, Ali Mesbah, Mark Molaro, and Xiaoxiang Zhu. I thank Darin Bellisario and Youngwoo Sun (from the Strano group) for involving me in their work on modeling carbon nanotube solar cells and photoconductive atomic

force microscopy, respectively, which were interesting projects not discussed in this thesis. Thanks to the many collaborators throughout the globe that I have worked with during my thesis: Rolf Findeisen, Sergio Lucia, Roberto Marseglia, Tillmann Mühlpfordt, Davide Raimondo, Stefan Strief, Marcello Torchio, and Mengling Wang. Basketball has always been one of my favorite pastimes. I would like to thank all of the members of the MIT basketball community for the many wonderful pickup games over the past few years. I will fondly remember my teammates on the Chemical Engineering Basketball Team, and I cannot thank them enough for their support and drive during our surprise run to the intramural finals.

I gratefully acknowledge the funding agencies that have supported my work: the National Science Foundation Graduate Research Fellowship, the Novartis-MIT Center for Continuous Manufacturing, and the Defense Advanced Research Project Agency.

Lastly, I would like to thank my entire family for being a constant source of moral support. My biological parents Diana Raley and Scott Paulson and my stepfather Michael Raley have been a source of love, wisdom, and encouragement since the day I was born, which has shaped the man I am today in more ways than I will ever know. They also laid the foundation for my education, which made this work possible. Words cannot express my gratitude for them. I would also like to offer my most heartfelt thanks and gratitude to my wife You Peng who has directly and indirectly contributed to this thesis in so many ways that I cannot even begin to enumerate. To my unborn child, I dedicate this thesis to you. Your mother and I already love you so much and I hope you get a chance to read this work in the future.

Contents

I Introduction 22

1 Introduction 23

II Uncertainty Quantification and Propagation 31

2 Control of Self-assembly in Micro- and Nano-scale Systems 33

2.1 Introduction 33

2.2 Challenges 34

2.3 Promising Research Directions 39

2.4 Outlook for Future Research 58

2.5 Importance of Uncertainty: An Illustration 61

3 Polynomial Chaos Framework 69

3.1 Introduction to Probability Theory 69

3.2 Quantification of Uncertainty from Data 72

3.3 Power Series Methods 77

3.4 Wiener–Hermite Polynomial Chaos 81

3.5 Generalized Polynomial Chaos 88

III Constrained Predictive Control of Large-scale Systems 91

4 Optimization Methods for Fast Model Predictive Control 93

4.1 Introduction 93

4.2 Problem Formulation 95
4.3 Cases of Model Predictive Control 99
4.4 Optimization Methods for Quadratic Programming 105
4.5 Sparse and Condensed Formulations of MPC 111
4.6 Explicit MPC 118
4.7 Example: Quadratic Dynamic Matrix Control 121
4.8 Concluding Remarks 128

5 Plant-wide Control for Continuous Pharmaceutical Manufacturing 133
5.1 Introduction 133
5.2 Process Description 136
5.3 Control-relevant Process Modeling 140
5.4 Application to Integrated Continuous Pharmaceutical Pilot Plant . . 146
5.5 Conclusions 153

IV Stochastic Model Predictive Control 156

6 Fast Model Predictive Control of High-dimensional Systems with Probabilistic Uncertainty 157
6.1 Introduction 157
6.2 Can Parameter Uncertainty Lead to Instability in MPC? 160
6.3 Problem Formulation 161
6.4 Galerkin Projection for DAE Systems 164
6.5 Fast MPC with Probabilistic Parameter Uncertainty 168
6.6 Example: End-to-end Continuous Pharmaceutical Manufacturing . . 173
6.7 Conclusions 176

7 Optimal Risk Allocation for Disturbance Rejection 177
7.1 Introduction 177
7.2 Problem Formulation 180
7.3 Feedback Parametrization of the Controller 182

7.4 Joint State Chance Constraints 185
7.5 Optimizing Feedback and Risk Allocation Simultaneously 188
7.6 Example: Continuous Bioreactor Process 193
7.7 Conclusions 197

8 Stability in Stochastic Receding Horizon Control 199
8.1 Introduction 199
8.2 Problem Formulation 201
8.3 Deterministic Surrogate 204
8.4 Tractable Stochastic Model Predictive Control Algorithm 206
8.5 Stability Analysis for the Unconstrained Case 211
8.6 Example: Van de Vusse Reactor 218
8.7 Conclusions 220

9 Improved Output Feedback with Bayesian Learning 223
9.1 Introduction 223
9.2 Failure of Chance Constraints in Receding Horizon Control 225
9.3 Recursive Bayesian Estimation using Polynomial Chaos 230
9.4 Output Feedback MPC with Bayesian Learning 234
9.5 Example: Reactors in Series 235
9.6 Conclusions 236

V Fault/Failure Tolerant Process Control 239

10 Multi-objective Failure Tolerant Controller Design 241
10.1 Introduction 241
10.2 Systematic Design of Multi-objective Controllers 243
10.3 Design of Failure-tolerant Controllers 250
10.4 Design of Multi-objective Controllers for Multi-loop Systems 253
10.5 H_2-optimal Controllers for SISO Systems 257
10.6 Example: Continuous Thin-film Dryer 260

10.7 Conclusions 267

11 Active Fault Diagnosis for Nonlinear Systems **275**

11.1 Introduction 275

11.2 Problem Formulation 277

11.3 Convex Relaxations 281

11.4 Optimal Robust Separating Input 283

11.5 Example: Two-tank Benchmark Problem 287

11.6 Conclusions 292

VI Conclusions and Suggestions for Future Work **293**

12 Conclusions and Future Outlook **295**

12.1 Summary of Contributions 295

12.2 Suggestions for Future Work 297

List of Figures

1-1 Block diagram illustrating the structure of modern "advanced" model-based controllers where $u^\star$ are control inputs calculated as the optimizers of the optimal control probleml, y are the measured outputs of the process, n is a measurement error (e.g., noise or bias), and r is the reference value the optimizer (e.g., the setpoint for the outputs y) . . 25

1-2 General approach for the combining first-principles models with measured data to develop an accurate model for design and control. . . . 26

1-3 Illustration of the layered control structure typically utilized in plant-wide control. The regulatory layer typically connects to the physical hardware in the plant and should be designed to reject fast disturbances in the process. The supervisory layer commonly utlizes dynamic constrained optimization (such as MPC) to drive the process to target values. These target values are often chosen by a real time optimizer that optimizes the economics based on a steady-state model. 30

2-1 Active self-assembly of patchy particles enables bottom-up construction of sophisticated structures from smaller particles. Red and green regions are surface patches positioned on particles to govern directional interactions between particles and particle specificity. The surface patches resemble the arrangements of bonds around atoms. Reprinted by permission from Macmillan Publishers Ltd: Nature [117], Copyright 2012. 41

2-2 A schematic of a hypothetical system of hexagonal prisms assembled from DNA-functionalized nanoparticles, which represents a type of active self-assembly. Double-stranded DNA is attached to the gold nanoparticles on the surface. A small strand of self-complementary single-stranded DNA acts as a sticky patch at the end of the DNA linker to which the DNA on other nanoparticles is attached. Reprinted by permission from Macmillan Publishers Ltd: Nature Materials [84], Copyright 2010. 42

2-3 Active self-assembly of millimeter-scale wires using nano-scale transport. Surface-adhered kinesin motor proteins enable transport of biotinylated microtubules, which form microtubule bundles due to cross-linkage of streptavidin. Reprinted (adapted) with permission from [111]. Copyright 2011 American Chemical Society. 43

2-4 Illustration of the experimental apparatus used drive the assembly of paramagnetic colloids via toggling the external magnetic field. A pair of Helmholtz coils are used to generate the magnetic field (whose field lines are normal to the gravitational field) and a DSLR camera was used to generate images. The suspension is deposited on a glass slide with a coverslip so that the colloids can only move in the horizontal plane perpendicular to gravity due to their strong sedimentation. Reproduced from [243] with permission from The Royal Society of Chemistry. . . 44

2-5 The time evolution of a suspension of paramagnetic colloid particles (dispersed in ultra-pure water with a volume fraction of 0.5%) under the influence of a toggled magnetic field that is turned on and off at various frequencies (denoted toggling frequency). The magnetic field strength is 1500 A/m. Each image is 3.2 × 2.1 mm in dimension. All suspensions appear to reach their final/slowly evolving state after ~2000 seconds. There is a clear region near 0.66 Hz in which the suspension condenses into large crystalline domains (favored state). At high toggling frequencies ($\geq$ 5 Hz), the suspension remains trapped in the unfavorable entangled/disordered state. At 0.33 Hz, the kinetics of chain-breaking appear to slow significantly so that it will take a long time to reach the favored state of large crystalline domains. This represents an open-loop control policy for magnetic toggling frequency that was computed heuristically. Reproduced from [243] with permission from The Royal Society of Chemistry. 46

2-6 Open-loop control policy used to drive the system to a desired arrangement of particles. Please refer to the text in Section 2.3 for detailed descriptions of this figure and the underlying optimization problems that must be solved to derive the control law. (a) Dynamic path that restricts the system to progressively smaller subsets of the system phase space (denoted Ω_α^i for the ith stage). (b) A one-dimensional example of the phase space restriction. Reprinted (adapted) with permission from [234]. Copyright 2010 American Chemical Society. 49

2-7 Experimental demonstration that clathrin self-assembly is robust to changes in pH due to weak specific interactions. This methodology can be useful for designing open-loop control policies to be more robust. (a and c) Cryo transmission electron microscopy (TEM) image of clathrin self-assembly at pH = 6.0 after 20 minutes and 4 days. (b and d) Cryo TEM image of clathrin self-assembly at pH = 5.1 after 20 minutes and 4 days. Representative spherical cages and disordered aggregates of clathrin are outlined in blue and red, respectively. (e) Dot plot showing the major axis distribution of ellipses fit to the clarthrin assemblies at pH = 6.0 (circles) and pH = 5.1 (triangles) after 20 minutes (open) and 4 days (filled). Solid lines represent the mean value (n = 63, 169, 64, and 54). Reproduced from [224] with permission from The Royal Society of Chemistry. 52

2-8 Image of a microfluidic platform that uses evaporation to induce nucleation in microliter droplets (see [245, 246]). The evaporation rate in each droplet is specified by the partial pressure of water at the droplet surface, the area and length of the channel that connects the droplet to external air, and the humidity of the external air. This apparatus can be used for open- or closed-loop control based on manipulation of the evaporation rate. Reprinted from [250] with permission from Elsevier. 54

2-9 Comparison of closed-loop (feedback) and open-loop control policies for microliter droplet crystallization for the system in Fig. 2-8. The output is the number of crystals at the final time in the simulation while the manipulated variable (input) is the volume of the droplet. A proportional controller was used to incorporate feedback with the desired number of crystals (i.e., setpoint) set to 1. The open-loop input trajectory was taken to be the average of the input trajectories for the 1000 closed-loop simulations. Other intuitive open-loop input trajectories (e.g., those corresponding to fastest/slowest induction time in the 1000 simulations) resulted in even worse performance. 55

2-10 Experimental apparatus used for closed-loop control of colloidal assembly. (a) Experimental setup of gold film quadrupole electrodes on a glass slide microscope slide and an o-ring container that houses an aqueous dispersion of silica colloidal particles. (b) Example image taken using optical microscopy used as a sensor for system measurements. Theoretical Cartesian coordinate-based potential energy wells were computed, using the electrode center as a reference, at (c) $V = 4$ V, $\omega = 1$ MHz and (d) $V = 4$ V, $\omega = 0.1$ MHz. Reprinted (adapted) from [118] with permission from John Wiley and Sons. 56

2-11 Feedback controlled self-assembly of a colloidal crystal containing 130 particles using the control law in (2.8). Optical microscopy measurements with $\langle C_6 \rangle_{SP}$ equal to (a) 2, (b) 3, (c) 4, (d) 5, and (e) 6. (f) Feedback-controlled self-assembly (0 to 360 seconds) and disassembly (480–720 seconds). The top pane shows $\langle C_6 \rangle_{SP}$ (solid blue line) and $\langle C_6 \rangle_{PV}$ (blue points) versus time. The bottom pane shows the manipulated variables voltage V (green line) and frequency ω (orange line) versus time computed using (2.8). Reprinted (adapted) from [118] with permission from John Wiley and Sons. 57

2-12 Optimal temperature profile for the crystallizer problem in Section 2.5. The temperature trajectory was approximated with a piecewise linear function with 8 discretization intervals. Also, an industrially common two-stage linear cooling profile is shown for comparison. 66

2-13 Nucleated to seed mass ratio distributions for the optimal temperature profile. The distribution at the final time has significant variation due to uncertainty in the nucleation and growth parameters, which indicates that this "optimal" profile is not robust to uncertainty. . . . 66

2-14 Nucleated to seed mass ratio distributions for the two-stage linear cooling profile (commonly type of profile implemented in industry). Although the objective is 15% higher than the optimal profile when the system is at nominal parameter values, the profile has significantly less variation due to parametric uncertainties. 67

3-1 Basic description of the polynomial chaos framework and how to systematically select the order to achieve a desired level of accuracy. . . 90

4-1 Critical regions Π_i for a randomly generated example. 120

4-2 Solution to (4.40) for a unit step in the input. The output values, and thus step response coefficients, are directly computable from the $E(1/2, t)$ line on this plot. 129

4-3 Closed-loop response of the hyperbolic PDE system output. See Section 4.7 for the details of the simulation. 129

4-4 Input profile, computed using the QDMC algorithm, supplied to the hyperbolic PDE system and corresponds to the output profile shown in Figure 4-3. See Section 4.7 for the details of the simulation. 130

4-5 Comparison of CPU times (of a single optimization problem averged over 20 runs) versus number of states in the true system for a naive implementation of sparse state-space MPC and QDMC. The algorithms were coded in Matlab® and run on a laptop PC (Intel i7, 2.7 GHz, 8 GB RAM). 130

5-1 Integrated continuous pharmaceutical manufacturing pilot plant equipped with a stabilizing control layer. R reactor, S separator, C crystallizer, M mixer, W washing/filtering unit, D dilution tank, E extruder, MD molding unit, P pump, CC concentration control, FC flow control, LC level control, SP setpoint. 137

5-2 Synthetic reactions from intermediate **1** to active pharmaceutical ingredient **6**. 139

5-3 Dynamic sensitivity analysis of the critical quality attributes with respect to the potential critical process parameters (the streams are ordered as in Figure 5-1). 141

5-4 Validation of the identified and linearized models with respect to an independent data set generated using the plant simulator. 144

5-5 Illustration of the receding-horizon implementation of the plant-wide QDMC system. 148

5-6 Closed-loop control of the ICM pilot plant with the plant-wide MPC and regulatory control systems in the presence of parametric uncertainties in the intermediate and API synthesis reaction kinetics (reactors **R1** and **R2** in Figure 5-1). 150

5-7 Closed-loop control of the ICM pilot plant using the plant-wide MPC and regulatory control systems in the presence of persistence disturbance in the filtration units (filter units **W1** and **W2** in Figure 5-1). 152

5-8 Closed-loop control of the ICM pilot plant using the plant-wide MPC and regulatory control systems in the presence of temporary disturbance in the purity level of the intermediate compound **1** (stream **1** in Figure 5-1). 153

5-9 Dynamic response of the ICM pilot plant to a step increase in the production rate. The pilot plant is in closed-loop operation with the plant-wide MPC and regulatory control systems. 154

6-1 Illustration that LQG optimal controller does not provide guaranteed stability margins when there is process gain uncertainty. The output of interest is $y(t) = x_1(t)$. 162

6-2 Dynamic response of production rate (normalized) for 200 closed-loop simulations of a setpoint change in the production rate. 174

6-3 Dynamic response of API dosage (normalized) for 200 closed-loop simulations of a setpoint change in the production rate. 175

6-4 Histograms of the API dosage (normalized) at various times based on 200 closed-loop simulations of a setpoint change in the production rate. 175

7-1 Probability distributions of butanol at process times 1, 10, 20, 30, and 40 hr; the black dashed line shows the setpoint. A 10% change in the butanol setpoint is applied at time 0 hr. 197

7-2 Joint probability distribution of acetate and butyrate at process time 1 hr; the black dashed lines show the bounds of the state constraints. 198

8-1 Histograms of x_1 at different times obtained from 100 closed-loop simulations of the receding-horizon SMPC (blue) and MPC (red). The proposed SMPC approach leads to smaller mean and variance of x_1 in the presence of probabilistic uncertainties and process noise. 220

8-2 Time profiles of x_2 obtained from 100 closed-loop simulations of the proposed SMPC approach and nominal MPC. The red-dashed line represents the state constraint. Nominal MPC results in violation of the constraint in 46% of the cases. 221

9-1 Cascade of two CSTRs with reaction $A \rightarrow B \rightarrow C$ and nonadiabatic flash with purge and recycle, e.g., [254]. 236

9-2 Histograms of PDF ($M = 1000$) of parameters $\tilde{\theta}_1$ and $\tilde{\theta}_2$ for cascaded CSTRs and flash over time t with $\theta_1^* = \theta_2^* = 2.5$ depicted as solid line. 237

9-3 Stochastic MPC inputs (as deviations from steady state) to cascaded CSTRs and flash. 237

9-4 States (as deviations from steady state) of cascaded CSTRs and flash under inputs from Figure 9-3. Measurements depicted as dots. 238

10-1 General classical feedback control structure. 244

10-2 General control structure for a stable LTI system. 245

10-3 Cascade control system. 255

10-4 Classical parallel cascade structure. 268

10-5 Coordinated control system. 269

10-6 Dynamic behavior of the single-loop control system for a step change in the reference r, the measured load disturbance l_m, the measured output disturbance d_m, and the unmeasured output disturbance d_u at $t = 0$, 150, 200, and 250 s, respectively. 270

10-7 Dynamic behavior of the single-loop control system during various failures in the sensors of the measurable variables for a step change in the reference r, the measured load disturbance l_m, the measured output disturbance d_m, and the unmeasured output disturbance d_u at $t = 0$, 150, 200, and 250 s, respectively. 271

10-8 Dynamic behavior of the series and parallel control systems for a step change in the reference r, the unmeasured output disturbance in the primary loop d_{u1}, the measured output disturbance in the primary loop d_{m1}, the unmeasured output disturbance in the secondary loop d_{u2}, the measured load disturbance in the primary loop l_{m1}, and the measured load disturbance in the secondary loop l_{m2} at $t = 0$, 100, 150, 200, 300, and 350 s, respectively. 272

10-9 Dynamic behavior of the coordinated control system for a step change in the reference r, the measured load disturbance for the slow input l_{m2}, the measured load disturbance for the fast input l_{m1}, the measured output disturbance d_m, the unmeasured output disturbance d_u, and the input setpoint u_r at $t = 0$, 50, 100, 150, 200, and 300 s, respectively. . 273

11-1 Sequential two-tank system. 287

11-2 1000 sampled outputs are shown by circles from the nominal and faulty models when the optimal separating input, calculated from the t-LP relaxation, is injected. $f^{[0]}$, $f^{[1]}$, $f^{[2]}$, and $f^{[3]}$ are represented by the blue, red, green, and yellow circles, respectively. The convex hulls (black lines) in the lower-right panel were drawn for clearer illustration. 291

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

2.1 Parameters used in the batch cooling crystallization optimal control case study. The parameters and model were adapted from [171], which is based on an industrial crystallization of potassium nitrate from water. 65

3.1 Correspondence of Wiener–Askey polynomial chaos basis to the distribution of the random input 90

5.1 The effective CPPs used for plant-wide identification and control (the streams are ordered as in Figure 5-1). 142

6.1 Variance comparisons of API dosage (normalized) at various times based on 200 closed-loop simulations of a setpoint change in the production rate. 175

7.1 Settings of the SMPC problem in the continuous bioreactor case study. 196

10.1 Multi-objective controllers for the single-loop control system. 262
10.2 Multi-objective controllers for the series cascade control system. . . . 264
10.3 Multi-objective controllers for the parallel cascade control system. . . 265
10.4 Multi-objective controllers for the coordinated control system. 266

11.1 Comparison of the proposed method at different levels of convex relaxation. By including the additional constraints in (11.4), t-LP is a tighter relaxation than LP. Computations performed on a Desktop PC (Intel i7, 2.7GHz, 8 GB RAM) running Windows 7 (64-bit) using a single core. 292

Part I

Introduction

Chapter 1

Introduction

Issues in Chemical Process Control Most chemical products are manufactured through a number of interconnected unit operations whose underlying phenomena mainly involve chemical reaction networks, fluid mechanics, and heat and mass transfer. As the design and operation of chemical plants have become more sophisticated, the control of chemical processes has become increasingly difficult, which has driven the demand for improved control techniques and methodologies.

Industrial model predictive control (MPC) is the most advanced control technology widely practiced in industry [174]. This technology initially developed in the chemical engineering field in the 1970s was a major advance over earlier multivariable control technologies such as decoupling control, mainly in its ability to explicitly handle actuator, state, and output constraints. Industrial MPC technology has its limitations, so significant research has been carried out over the past two to three decades to develop advanced control methods that have some increased capability compared to past control methods. For such "advanced control" to be widely implemented in industry, its capabilities must be a major step beyond current industrial MPC technology. The main reason for the limited application of current "advanced" control techniques (that have been developed/explored in academia) is that they do not adequately address all of the requirements of industrial control systems [32]:

1) The controlled system must be insensitive to the presence of uncertainties and

plant/model mismatch including unknown parameter values, neglected dynamics, unmeasured or poorly modeled disturbances, and sensor noise.

2) The controlled system must perform well under input, actuator, and state constraints.

3) The controller should be able to handle a large number of interacting variables (due to, for example, interrelation of sequential processing steps as a result of recycles, heat integrations, etc.) and nonlinearity in the process dynamics.

4) The controlled system must be safe, reliable, and easy to maintain in the presence of system faults.

Methods for designing controllers for single-input single-output (SISO) plants that are insensitive to plant/model mismatch and unknown disturbances were developed in the 1940s. The extension to multivariable plants was found to not be as straightforward. The theory of "optimal" control (e.g., Linear Quadratic Gaussian control) was developed for multivariable linear systems in the 1960s [61]. Soon after, however, it was shown that optimal controllers could be very sensitive to plant/model mismatch (i.e., small perturbations in the model can lead to very poor performance/instability when applying the controller to the real system) [179, 181].

Most modern advanced control techniques are model based and look to apply mathematical optimization tools to optimize the performance based on future model predictions. The necessary components for this framework are (i) a dynamic model, (ii) an estimator that converts measured process variables into estimates of unmeasured states and/or parameters, and (iii) a control algorithm that computes the optimal control action based on model predictions from the state estimate for a given objective function and constraint set [216]. A typical block diagram of this procedure is shown in Figure 1-1.

Clearly, a model of the process must be developed when using this type of control strategy. In this case, control performance is strongly related to the accuracy of this model, which is intimately tied to how the model was developed. The field of system

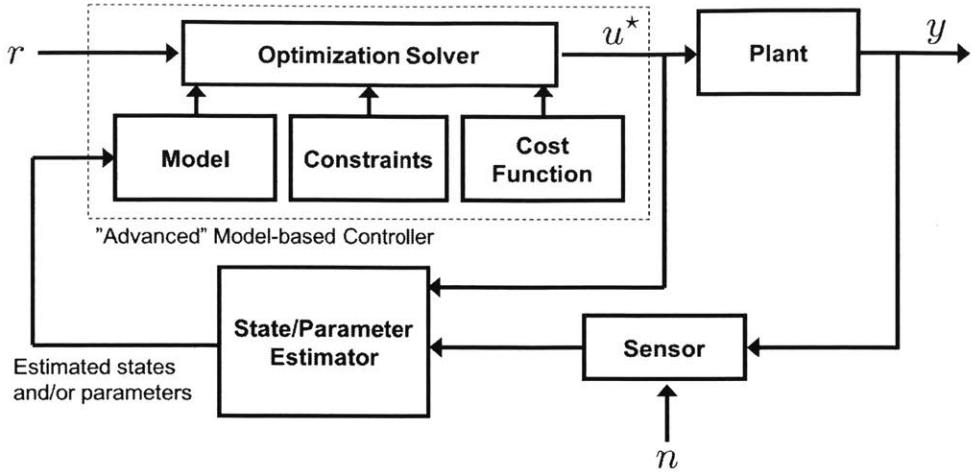


Figure 1-1: Block diagram illustrating the structure of modern "advanced" model-based controllers where $u^\star$ are control inputs calculated as the optimizers of the optimal control probleml, y are the measured outputs of the process, n is a measurement error (e.g., noise or bias), and r is the reference value the optimizer (e.g., the setpoint for the outputs y)

identification uses statistical methods to build models of dynamical systems from measured data. Most system identification algorithms (including those used in current industrial control technology) are black-box in nature, meaning that no prior model is assumed and the model is developed entirely from input-output data. Although these purely data-based methods are general and relatively easy to implement on complex systems, data-based methods have many limitations including that (1) most methods restrict the model to be linear, (2) the methods require a very large amount of data (that is not always available) to provide any statistical guarantees, and (3) the resulting uncertainty descriptions are not necessarily accurate or related to physical quantities (that is, are hard to interpret). On the other hand, white-box models are developed entirely from first principles meaning the internal structure is assumed to be known *exactly* [56]. For most chemical process systems, typically there is much first-principles knowledge of the system when the plant was designed, but all the system parameters may not be known (for example, a chemical reactor may be known to operate approximately in plug flow, but with many unknown rate constants).

The combination of data-based and first-principles models is commonly referred to

as *gray-box* modeling. A commonly used approach for developing accurate gray-box models, that has been applied in its simplest form since the early 1970s, is summarized in Figure 1-2. This approach, which designs experiments so as to maximize model accuracy, has much lower data requirements than constructing black-box models of similar accuracy. This approach also has a much higher computational cost, which is strongly correlated with model complexity. For sufficiently complex systems, development and optimization over first-principles models can become quite time-consuming and expensive, which is likely the reason that the current state-of-the-art plant-wide system identification algorithms used in industry are geared toward black-box models developed purely from data. The first part of this thesis discusses ways to the bridge this gap through the use of first-principles modeling for a variety of complex chemical systems. Certain mathematical tricks can be used to simplify the models (including how to characterize and propagate their uncertainty descriptions) needed in optimization, design, and dynamic control.

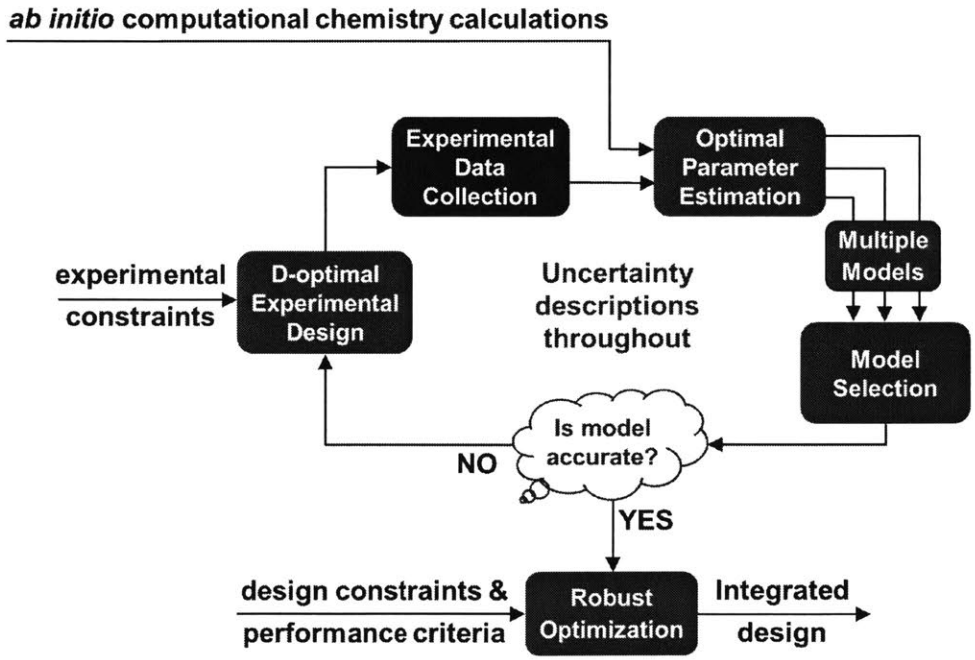


Figure 1-2: General approach for the combining first-principles models with measured data to develop an accurate model for design and control.

The next major part of this thesis aims to develop a control methodology that

meets all the requirements listed above. Very few industrial control algorithms explicitly account for uncertainties during controller design. Instead, robustness is achieved through tuning a large number of parameters in the controller, which can be quite cumbersome to select as many iterations may be needed to achieve a desired level of robustness. Furthermore, it is difficult to systematically trade off robustness and performance in an optimal manner. Using gray-box models (with rigorously quantified uncertainty descriptions), control formulations, algorithms, and theory can be developed to provide robustness guarantees with minimal-to-no tuning.

Thesis Overview First, a detailed discussion of the modeling and control of micro- and nano-scale self-assembling systems is provided. This is done, both to familiarize the reader with an important family of chemical processes (that includes crystallization which is one of the most challenging chemical separation processes to control), and to illustrate the importance of handling uncertainty within model-based control methods using a well-known batch crystallization example.

Next, a detailed discussion on uncertainty quantification and of polynomial chaos methods for uncertainty propagation is provided. The relationship of polynomial chaos to other commonly used uncertainty propagation methods is also explored and summarized. Polynomial chaos' ability to handle a wide-range of probability distributions (e.g., Gaussian, uniform, and Beta distributions), as well as its computational efficiency relative to these other methods, motivates the core of this thesis which is the pairing of the polynomial chaos framework with MPC.

The state-of-the-art MPC formulations and solution methods are overviewed and elaborated upon next. The main focus is on the various methods that have been developed for so-called *fast MPC*, in which the resulting optimization problem can be solved most efficiently. Fast MPC methods fall into two categories, mainly the choice of optimization method and the structure of the formulated MPC optimization problem. It is shown how the quadratic dynamic matrix control (QDMC) algorithm (the most commonly applied variant of model-based control applied in the chemical process industry) relates to the more modern state-space MPC, and the features of

the algorithm that make it so effective in the chemical process industry are discussed. These concepts are illustrated on systems that can be modeled by a second-order hyperbolic partial differential equation.

The QDMC framework is then used to develop a plantwide controller for a continuous pharmaceutical manufacturing pilot plant. Integrated continuous manufacturing offers ample opportunities for efficient and cost effective production of pharmaceutical drugs. In the pharmaceutical industry, the ability to directly handle constraints is a key requirement in order to meet the stringent regulatory requirements on critical quality attributes (CQAs). One of the biggest challenges in plantwide control of such processes is highly interactive dynamics of the integrated units, which if not properly accounted for during controller design can lead to unacceptable performance. It is demonstrated that QDMC effectively handles both of these challenges in the presence of a number of uncertainties/disturbances.

This directly leads to the next part of the thesis, which explores stochastic control methods for handling probabilistic parameter uncertainty and disturbances as a way to reduce the conservatism of their worst-case counterparts. First, we develop a method to handle probabilistic time-invariant parameter uncertainty in MPC for high-dimensional systems by applying polynomial chaos methods within the QDMC algorithm. We show how this approach is more robust to parametric uncertainty when compared to nominal QDMC using continuous pharmaceutical manufacturing as the motivating case study.

Next, stochastic MPC of linear systems subject to arbitrary (possibly unbounded) stochastic disturbances is explored. The proposed method tackles the stochastic disturbance rejection MPC problem, which is one of the other main sources of uncertainty encountered in chemical processes. Joint state chance constraints are incorporated into the method using the idea of risk allocation. These two ideas are then brought together to develop a stochastic MPC method that is capable of handling both probabilistic time-invariant parameter uncertainty and additive stochastic disturbances. Stochastic closed-loop stability of this approach is also explored using theory for Markov processes.

Next, it is proved (by counterexample) that chance constraints are not guaranteed to be fulfilled by the closed-loop system when applying a receding-horizon controller to systems with time-invariant uncertainty. This motivates the development of algorithms that can adapt the parameter distribution at each step based on measurements of the system. This concept of Bayesian learning is then discussed within the context of stochastic MPC, and an algorithm for solving this problem using polynomial chaos is introduced.

The final part of this thesis explores methods for failure and fault-tolerant process control. First, a general internal model control structure (with multiple degrees-of-freedom) for stable systems is presented. This control structure, which could be placed at the regulatory layer below that of MPC (see Figure 1-3 for brief illustration), enables designing independent controllers that each can be optimized according to a particular objective. The global optimality of the multi-objective control system is shown to remain intact when one or more controllers is taken offline. This circumvents the need to redesign the control system whenever one or more of the manipulations and/or sensors fails, which suggests tolerance to failures. Next, a model-based fault diagnosis method is developed for uncertain nonlinear systems. The approach is referred to as *active* as it looks to determine input actions that guarantee that the reachable sets of measurements do not overlap between the nominal and faulty models. This is useful as it allows for early detection of non-obvious faults in the system to avoid catastrophic failure. This is followed a summary of ideas for future research.

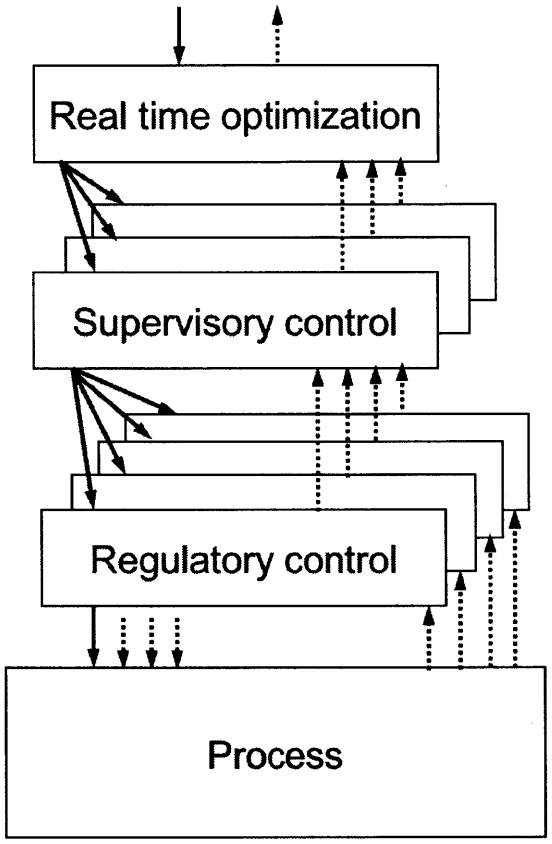


Figure 1-3: Illustration of the layered control structure typically utilized in plant-wide control. The regulatory layer typically connects to the physical hardware in the plant and should be designed to reject fast disturbances in the process. The supervisory layer commonly utlizes dynamic constrained optimization (such as MPC) to drive the process to target values. These target values are often chosen by a real time optimizer that optimizes the economics based on a steady-state model.

Part II

Uncertainty Quantification and Propagation

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Control of Self-assembly in Micro- and Nano-scale Systems

2.1 Introduction

Self-assembly is a process in which particles spontaneously arrange into complex patterns or organized superstructures [265]. Systems with self-organizing characteristics are commonly encountered in nature and engineered technologies, where particles can be of all scales ranging from molecules in a crystal to cells in a tissue to planets in a galaxy [264]. Bottom-up engineering of self-assembly systems enables manufacturing materials and devices with novel optical, mechanical, and electronic properties. The innovative applications of self-assembly at the micro- and nano-scales have sparked interest in understanding the physics, dynamics, and implementation of self-organizing systems. Control of self-assembly processes is key to the manufacture of materials with unique properties.

This chapter aims to provide an overview on the recent progress of controlling self-assembly of micro- and nano-scale systems. Controlled self-assembly implies promoting or accelerating the organization of particles towards desired structures. Intervention is expected in a self-organizing process, for example, by changing the particle interactions [84, 117] or by manipulating the environment (i.e., global system variables) in which self-assembly takes place [118, 235, 243]. This idea is often called

"directed self-assembly" [45, 93]. The concept should be distinguished from "directed assembly", which refers to the precise manipulation of particles one-by-one during the construction of the structure (like a mason building a brick wall) [118]. Directed assembly at the microscale is now considered standard manufacturing technology such as in three-dimensional printing (see e.g., [141, 49, 154] and the citations therein). In addition, directed assembly was recently demonstrated at the nanoscale by moving particles using the tip of an atomic force microscope [147, 133]. Directed assembly has a very strong bottleneck from a manufacturing point of view due to the limiting speeds at which you can manipulate the building blocks of the system. For example, a reasonable estimate for the printing speed of a three-dimensional printer with resolutions on the micro- to macro-scale is ~ 1 cm/s [173]. Assuming nanometer resolution is attainable with this printing speed, it would take $\sim 10^{14}$ seconds to print a device of 1 cm^3 volume with nanometer precision. Although directed assembly is a very active area of research, the topic is beyond the scope of this work.

Organization In the attempt to control self-assembly systems, many practical difficulties arise that are associated with the small-scale characteristics of the systems, which limit current technology and practice. This chapter outlines the major challenges in the control of self-assembly systems. Promising research directions in the areas of active self-assembly, open-loop control, and closed-loop control are motivated using examples from the literature. The chapter concludes with perspectives on the research outlook of control of self-assembly systems, and an illustration of the importance of uncertainty analysis using a batch crystallization example.

The majority of this chapter was published in *Journal of Process Control* [200].

2.2 Challenges

High-dimensional Stochastic Nonlinear Dynamics

In macroscopic systems, measured variables (i.e., outputs) typically are stochastic due to sensor noise and unknown disturbances arising from environmental fluctuations in

variables (e.g., temperature) acting on the system. Isolated stochastic terms can be included in deterministic models to account for this behavior on the macroscale [13]. In other words, the measured outputs of macroscale systems are most often deterministic in the absence of noise and unknown disturbances.

Micro- and nano-scale systems are different in that their underlying phenomena are *inherently stochastic* so that repeated experiments can produce different results even if the system has no noise or unknown disturbances [250]. This inherent stochasticity can greatly impact the self-assembly of particles at these scales. For example, self-assembly of colloidal particles (at fixed conditions) can require excessively long periods of waiting time before initiation of the first step of the process (e.g., nucleation) needed to make a product, due to the first step having a high-energy activation barrier [118]. Another example is a microfluidic platform that uses evaporation to induce crystal nucleation of organic compounds such as amino acids and proteins [245, 246]. The measured output for a single droplet is the *induction time* (i.e., the time at which the first crystal nucleates) and is best represented as an induction time *distribution* due to the stochastic nature of the system.

Stochastic dynamics with continuous states are typically described by Langevin equations [50, 284], which describe the time evolution of a group of variables that change slowly relative to other variables in the system. The original Langevin equation was derived as a modification to Newton's equations of motion to include Brownian motion and frictional drag due to collisions of particles (slow variables) with the solvent (fast variables). This system can be formulated as a stochastic differential equation (SDE) of the form [191]

$$d\mathbf{X}_t = \boldsymbol{\mu}(\mathbf{X}_t, t)dt + \boldsymbol{\sigma}(\mathbf{X}_t, t)d\mathbf{W}_t, \tag{2.1}$$

where $\mathbf{X}_t \in \mathbb{R}^n$ denotes an n-dimensional stochastic process, $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_N)$ denotes the drift vector, $\mathbf{W}_t$ denotes an m-dimensional Weiner process (i.e., Brownian motion), and $\boldsymbol{\sigma} = [\sigma_{ij}]$ is directly related to the diffusion tensor $\mathbf{D} = [D_{ij}]$ with

elements

$$D_{ij}(\mathbf{X}_t, t) = \frac{1}{2}\sum_{k=1}^{m} \sigma_{ik}(\mathbf{X}_t, t)\sigma_{jk}(\mathbf{X}_t, t). \tag{2.2}$$

This SDE is nonlinear and difficult to solve directly. Methods such as Monte Carlo simulation or Molecular Dynamics (MD) are available for obtaining time-averaged quantities of interest while avoiding direct simulation of (2.1). However, these methods are very computationally expensive and are only able to simulate complex systems for a very short period of time.

The dynamics of self-assembling systems involve the evolution of hierarchical components at different time scales due to their architecture (e.g., atoms make up proteins, proteins make up capsomers, and capsomers make up viral capsids [95]). Equation (2.1) alone cannot describe this behavior; instead, a multiscale approach is required (e.g., see [195]). However, multi-scale modeling approaches are very computationally expensive, taking on the order of days to simulate a relatively small self-assembly system (a system consisting of ~ 50 particles) using standard personal computers.

For systems with a discrete number of possible states, the stochastic dynamics are described by the Master equation [70, 121]

$$\frac{dP_\sigma}{dt} = \sum_{\sigma'} w_{\sigma' \to \sigma}(t) P_{\sigma'}(t) - \sum_{\sigma'} w_{\sigma \to \sigma'}(t) P_\sigma(t), \tag{2.3}$$

where $P_\sigma(t)$ denotes the probability that the system is in configuration σ at time t and $w_{\sigma' \to \sigma}(t)$ denotes the rate of transition from configuration σ' to configuration σ at t. The overall system is described by writing (2.3) (i.e., conservation equation for probability of configuration σ) for every possible configuration of the system. The probabilities can be stacked into a state vector $x(t)$ and the transition rates collected into a matrix $A(t, u(t); p)$ so that (2.3) can be written in the state-space form

$$\frac{dx}{dt} = A(t, u(t); p)x(t) \tag{2.4}$$

where $A(t, u(t); p)$ depends on time-varying variables (e.g, temperature), system in-

puts (i.e., manipulated variables) $u(t)$, and model parameters p such as chemical kinetic rate constants, diffusion coefficients, and thermodynamic properties of the system.

The main challenge in implementing control systems for processes modeled by (2.4) is that the number of states is usually very large (usually much greater than 10^{10}) for processes of practical importance [250]. Kinetic Monte Carlo (KMC) simulations are commonly used to approximate the solution of (2.4) by computing specific realizations of the Master equation. This approach uses calls from a random number generator to select a specific event to occur from a queue of all possible events, along with its corresponding time step, so that the time simulated in the KMC algorithm corresponds to real time [70]. Although this approach is usually much faster than solving (2.4) directly, KMC simulations can still take in the order of days for realistic systems. If state or output distributions are required for control, then a large number of KMC simulations are needed, which makes real-time control infeasible even for relatively simple systems. If the control objective depends only on coarse statistics of the distribution, then one approach is to develop low-order "equation free" models (e.g., [124, 123]) by fitting the KMC simulation results; however, the relationship between the manipulated variables and the system states in these models will no longer be transparent, making control less intuitive and more black box in nature [250].

Limited Sensors for Real-time Measurements

Controlling self-assembly systems at the micro- and nano-scale requires the acquisition of real-time information about the system status. This requirement leads to the needs of advanced real-time sensing techniques, while traditional self-assembly systems often rely on imaging or other characterization techniques performed after the assembly process to measure the local properties (e.g., using transmission electron micrographs to inspect the morphology in a self-assembling block copolymer system [128]).

Several factors result in real-time sensing in self-assembly systems being a challenge, including the small length scales, the slow and invasive nature of most observation techniques, and the limited variables that can be used to quantify the system

status. For crystallization in a nanoliter droplet implemented in a microfluidic platform [245, 246, 87], the small scale of the system inhibits implementing conventional methods of probing the solute concentration. While visual observation of the dynamics in self-assembled systems could be accessible through advanced microscopes (such as the fluorescent imaging technique used to track the real-time movement and clustering of Janus particles [46]), such information has to be translated into a variable that can represent the assembly status for control.

Limited Actuation for Control

Another challenge that naturally arises in controlling self-assembly systems is the limited availability of actuators. For a micro- or nano-scale self-assembly system, localized manipulation by *local actuators* to influence the assembly of the particles is constrained. Instead, controlling the self-assembly process often relies on manipulating macroscopic variables in the system (denoted *global actuators*). The underlying principle is that changing the global properties can change the dynamic pathways of assembly. For example, during crystallization, optimized temperature control can improve the yield of one particular crystal polymorph over another [101, 169]. The number of global variables that can be useful for altering the state of assembly are somewhat limited, which often includes temperature, pressure, concentration, composition, and, for some systems, external fields such as electric and magnetic fields.

The use of the macroscopic variables as actuators acting globally on the system implies an impact on all particles—no matter what the particle association state is. This non-specific action may lead to disruption of already formed structures. On the other hand, identifying the relationship between actuation and system response, as well as specifying the amount of actuation in order to drive the system towards the desired state of assembly, is not trivial. The task may become even more challenging when a system has complicated dependency on the variables used as the actuators. For example, the formation of snowflakes from water vapor is very sensitive to the particular changes in temperature and pressure [143].

Kinetic Traps in the Energy Landscape

Self-assembling systems typically have a range of stable or metastable configurations that can have vastly different structures and properties [85, 55]. A common but practically important example of multiple configurations is crystal polymorphism, which is the ability of a compound to crystallize as one or more distinct crystal species. Different polymorphs of a compound have different molecular structures and usually very different properties such as solubility, melting point, density, hardness, vapor pressure, optical properties, and electronic properties [96].

Although most self-assembling systems have one lowest energy state (i.e., the state that is thermodynamically favored), systems can become trapped in "kinetically arrested" states associated with local minima in the free energy landscape, giving rise to various metastable configurations (e.g., glasses, gels, polycrystals) [118]. For example, suspensions of micrometer-sized paramagnetic colloidal spheres form disordered entangled chain-like structures in a steady magnetic field at high field strengths, where thermodynamic calculations indicate that the formation of well-ordered crystallization domains is favored. The thermodynamically most stable crystalline state is not observed experimentally because the system is trapped in the entangled state and the energy barrier is too high for the system to overcome in a reasonable amount of time [243]. Avoiding kinetically arrested states is key to consistently achieving the thermodynamically most stable state and is an important issue to account for when designing control laws for self-assembling systems.

2.3 Promising Research Directions

This section describes some promising approaches for addressing the aforementioned challenges (see Section 2.2) in the control of micro- and nano-scale self-assembly systems. These approaches can be grouped into three broad categories: active self-assembly (particle design), open-loop control, and closed-loop control. The advantages and disadvantages of these different approaches are illustrated through the use of examples.

Sufficiently general systems engineering methods for multiscale systems can be applied to self-assembly systems, which have been discussed in detail in past reviews (e.g., see [250, 34, 35] and the citations therein). These methods are well-known and will not be discussed in this chapter, mainly because multiscale system techniques applied to small-scale self-assembling systems are computationally expensive and do not take advantage of underlying structure that can facilitate control tasks such as parameter estimation and feedback control [250].

Active Self-assembly

In active self-assembly [127, 273, 4], particles interact to reach a *consensus* regarding the desired global behavior of a self-organizing system, which depends on the state of all particles. The interactions among the particles are governed by a *consensus protocol* that specifies the information exchange between a particle and all of its surrounding particles. The primary challenge in active self-assembly is how to task the individual particles so that the global behavior of a self-organizing system is engineered (e.g., a desired structure emerges) with high probability, despite the stochastic nature of the system. This tasking requires the ability to design *local rules* for individual particles, which ensure the unique convergence to a pre-specified global behavior with the fastest convergence rate [273].

Active self-assembly provides opportunities for *bottom-up engineering* of micro- and nano-scale systems such as proteins in cells and nanoscale molecular machines. Self-assembly of *patchy particles* [117] has emerged as an effective approach to actively control structures that are assembled from small particles. Placing sticky patches on particles can cause the particles to interact only along certain directions and, therefore, alleviates the lack of specificity of particles (see Fig. 2-1). Generation of directional interactions between patchy particles mimics atomic bonding in molecules, and can significantly increase the structural complexity of a self-organizing system (see e.g., [86, 63] and the citations therein for detailed discussions on patchy particles/colloids and how their self-assembled patterns can be theoretically predicted). For example, Wang et al. [262] synthesized micrometer-sized particles with symmet-

rically arranged sticky patches of DNA on their surfaces to enable DNA-mediated interactions. The single-stranded DNA molecules attached to the patches mediate interparticle binding through hybridization with complementary DNA strands attached to patches on neighboring patches. In such a particle, the location of patches governs the directionality, and the sequence-dependent binding of DNA dictates specificity.

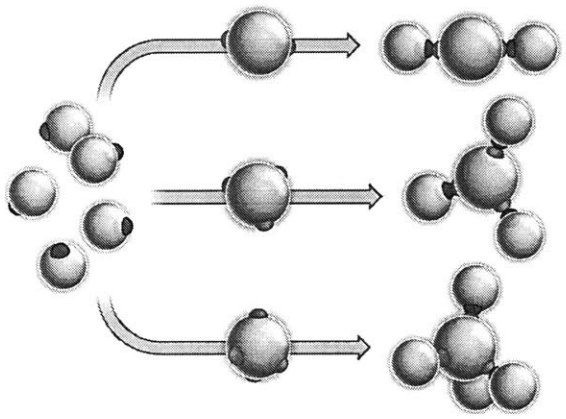


Figure 2-1: Active self-assembly of patchy particles enables bottom-up construction of sophisticated structures from smaller particles. Red and green regions are surface patches positioned on particles to govern directional interactions between particles and particle specificity. The surface patches resemble the arrangements of bonds around atoms. Reprinted by permission from Macmillan Publishers Ltd: Nature [117], Copyright 2012.

Patchy particles have been used to synthesize artificial molecules by combining mixtures of particles that have matched directional interactions and complementary DNA strands, such as DNA-functionalized gold nanoparticles shown in Fig. 2-2 [84]. In addition, an energy source can be used to accelerate the movement of particles, while modifying the particle interactions using sticky patches. Fig. 2-3 shows the assembly of biotinylated microtubules partially coated with streptavidin into linear bundles [111]. The movement of the biotinylated microtubules is facilitated by gliding them on a surface coated with kinesin motor proteins, which can result in gliding velocities up to 1 μm/s. The streptavidin coating enables microtubules to cross-link into microtubule bundles.

Similarly, *Janus particles* that combine incompatible elements in the same unit

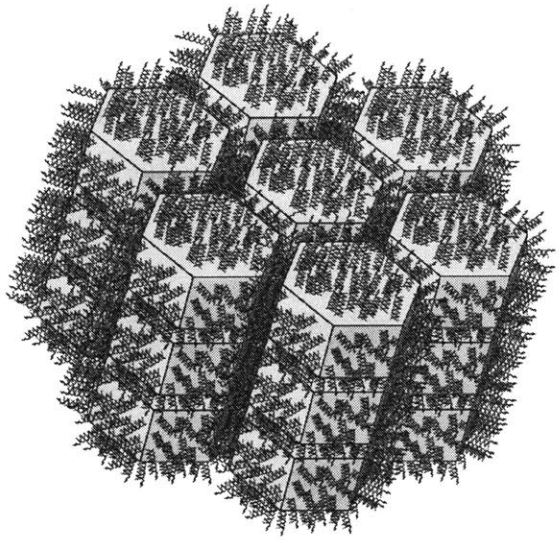


Figure 2-2: A schematic of a hypothetical system of hexagonal prisms assembled from DNA-functionalized nanoparticles, which represents a type of active self-assembly. Double-stranded DNA is attached to the gold nanoparticles on the surface. A small strand of self-complementary single-stranded DNA acts as a sticky patch at the end of the DNA linker to which the DNA on other nanoparticles is attached. Reprinted by permission from Macmillan Publishers Ltd: Nature Materials [84], Copyright 2010.

structure lead to the formation of persistent and defect-free superstructures [90]. This *ambivalence principle* is ubiquitously used in nature to create complex self-organizing structures. Examples are the formation of biological membranes from self-assembly of phospholipid molecules with polar head groups and hydrophobic tails, or the formation of a DNA strand from nucleotides that consist of a part capable of forming hydrogen bonds and an inert part. Applications of Janus particles are emerging for bottom-up manufacturing of complex matter through spontaneous self-assembly. It has been shown that, when immersed in an aqueous salt solution, spherical Janus particles (e.g., symmetric micelle structures) that are hydrophobic on one hemisphere and polar on the other polymerize into elongated strings and branched anisotropic structures [106].

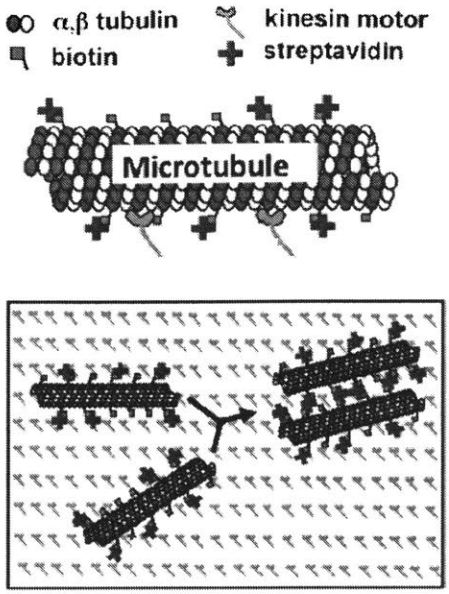


Figure 2-3: Active self-assembly of millimeter-scale wires using nano-scale transport. Surface-adhered kinesin motor proteins enable transport of biotinylated microtubules, which form microtubule bundles due to cross-linkage of streptavidin. Reprinted (adapted) with permission from [111]. Copyright 2011 American Chemical Society.

Open-loop Control

The term *open-loop control* is used to describe a system whose input profiles are set *a priori* (not informed by measurements/feedback). Open-loop input profiles can be computed heuristically, based on experimental observations and intuition, or using a model-based optimization, which requires a mathematical formulation of the system and control objective. For complex systems, a hybrid approach can be used to embed heuristic knowledge of the system in the optimization procedure and, therefore, reduce the complexity of the problem.

Open-loop control strategies are independent of measurements, which is the only option for micro- and nano-scale self-assembling systems that do not have real-time sensors available (see Section 2.2). However, the reduced information comes at the expense of robustness (i.e., the ability of the controller to reject disturbances, model imperfections, etc. by accounting for the current measured system state). The impact of feedback control using real-time measurements is explored in Section 2.3. As

discussed in Section 2.2, actuators that drive micro- and nano-scale self-assembly can be introduced either globally (changes felt throughout the system) or locally (changes felt only in the local environment). Global actuators act on the macroscopic scale which is easy to implement the manipulation, but do not provide the controllability enabled by local manipulations. Local actuators are more challenging to implement experimentally, but enable more effective actuation at the micro- and nano-scale.

A recent example of an open-loop control procedure developed *heuristically* using a single *global actuator* is the self-assembly of paramagnetic colloids into well-ordered crystalline domains using toggled magnetic fields [243]. The experimental setup is shown in Fig. 2-4. When no magnetic field is applied, the colloidal suspension remains dispersed in solution. In a constant magnetic field, the colloidal suspension forms chains parallel to the magnetic field lines, which start to aggregate laterally. The aggregation arrests the motion of the particles and leaves them in an entangled/disordered state that is at higher energy than the thermodynamically favored state of well-ordered crystalline domains.

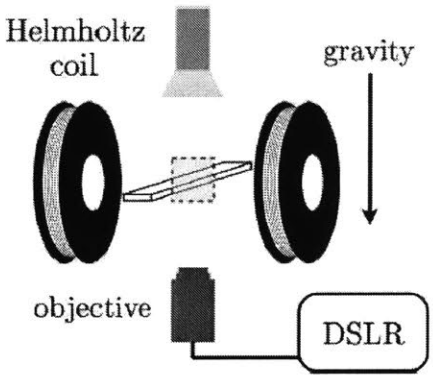


Figure 2-4: Illustration of the experimental apparatus used drive the assembly of paramagnetic colloids via toggling the external magnetic field. A pair of Helmholtz coils are used to generate the magnetic field (whose field lines are normal to the gravitational field) and a DSLR camera was used to generate images. The suspension is deposited on a glass slide with a coverslip so that the colloids can only move in the horizontal plane perpendicular to gravity due to their strong sedimentation. Reproduced from [243] with permission from The Royal Society of Chemistry.

Toggling the magnetic field on and off at particular frequencies enables the suspen-

sion as a whole to quickly enter its lowest energy state (well-ordered crystals). This open-loop control strategy is effective because, when the magnetic field is turned off, the attractive interactions between the colloids are suppressed allowing the particles to relax and reconfigure (thus breaking out of the kinetically trapped state) and reach the desired state in a much shorter period of time. In a constant magnetic field, the entangled chains are unlikely to break apart due to the attractive forces between the colloids, which creates a high energy barrier that traps the suspension in this disordered state.

The toggling frequency is the key parameter in such systems since the time that the field remains off must be long enough for the suspension structure to rearrange but not long enough for the colloids to diffuse very far and redisperse. That is, the toggling frequency should be similar to the characteristic relaxation rate of the suspension. The time evolution of the suspension at different toggling frequencies measured in [243] is shown in Fig. 2-5, where the dark regions represent the particle-rich phase. All frequencies form the entangled/disordered (unfavorable) state initially. As time increases, the chains start to merge and the system begin to "coarsen." The near-optimal toggling frequency is $\sim$0.5–1 Hz, where the coarse chains start to break apart at $\sim$500 seconds and form the desired state of well-ordered crystals (the favored state) after $\sim$2000 seconds.

The open-loop approach of cycling manipulated variables has proved useful in a variety of self-assembly processes, where the particles in the system are indistinguishable. Another related example is the use of temperature cycling in crystallization to change crystal shape, enhance crystal size uniformity, and increase polymorphic purity [7, 115].

An interesting alternative to these methods, which can increase controllability in small-scale self-assembling systems, is to introduce small external controls (e.g., electric charges, magnets). The external controls introduce attractive or repulsive interactions to the system potential energy to drive the system towards a particular configuration. Two approaches have been proposed for the design of these external controls:

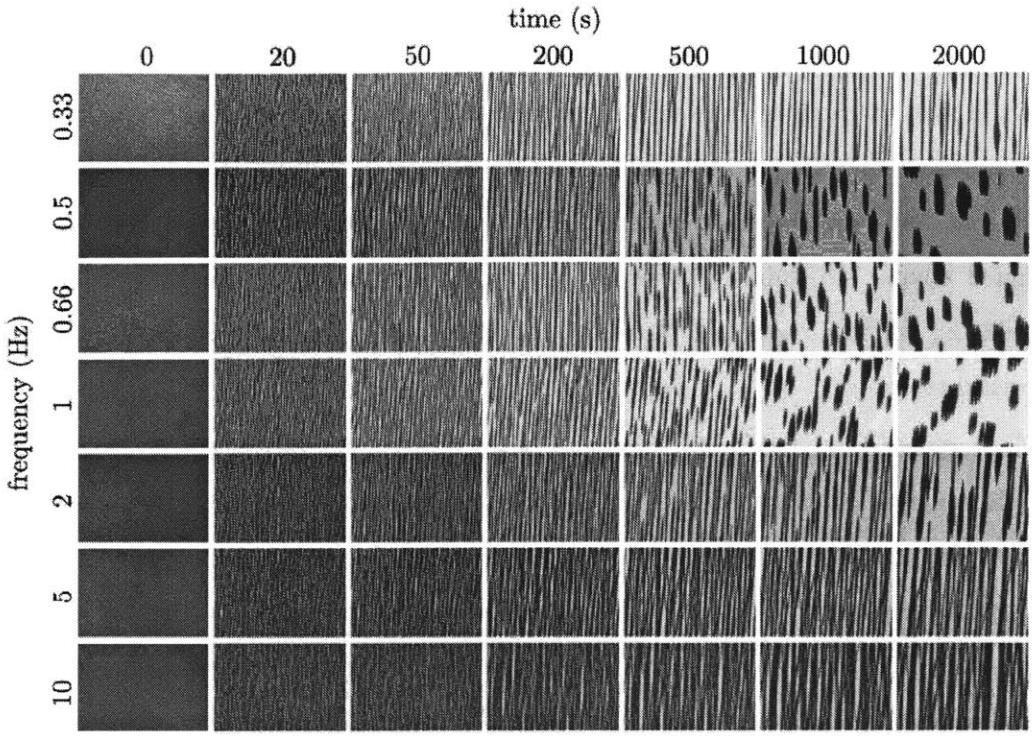


Figure 2-5: The time evolution of a suspension of paramagnetic colloid particles (dispersed in ultra-pure water with a volume fraction of 0.5%) under the influence of a toggled magnetic field that is turned on and off at various frequencies (denoted toggling frequency). The magnetic field strength is 1500 A/m. Each image is 3.2×2.1 mm in dimension. All suspensions appear to reach their final/slowly evolving state after $\sim$2000 seconds. There is a clear region near 0.66 Hz in which the suspension condenses into large crystalline domains (favored state). At high toggling frequencies ($\geq$ 5 Hz), the suspension remains trapped in the unfavorable entangled/disordered state. At 0.33 Hz, the kinetics of chain-breaking appear to slow significantly so that it will take a long time to reach the favored state of large crystalline domains. This represents an open-loop control policy for magnetic toggling frequency that was computed heuristically. Reproduced from [243] with permission from The Royal Society of Chemistry.

1. Robust Static Structures - What are the optimal external controls (number, locations, and strengths) so that the self-assembled micro- or nano-structure is stable to a desired degree of robustness (i.e., obtained with a sufficiently high probability in spite of system stochasticity)?

2. Robust Dynamic Paths - What external control profiles over time are needed to ensure that, with a high probability, the system evolves to the desired structure

from any initial distribution of particles?

The theoretical challenges associated with these systems-level questions are explored in detail in a two-part series (Robust Static Structures in [233] and Robust Dynamic Paths in [234]). This work falls into the open-loop control category of optimal *local actuation* computed using *model-based optimization.*

Algorithms for computing energy landscapes such that the system remains in the desired configuration with a sufficiently high probability are proposed in [233] (please refer to [233] for a detailed discussion of these algorithms and their results as they are only summarized here). An energy functional is proposed of the form [233]

$$E(\mathbf{z}) = \underbrace{\sum_{i=1}^{V}\sum_{k=1}^{N_d} z_i H_{i,k} s_k}_{E_{\text{ext}}(\mathbf{z})} + \underbrace{\sum_{i<j} z_i J_{ij} z_j}_{E_{\text{int}}(\mathbf{z})} = \mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{s} + \mathbf{z}^{\mathrm{T}}\mathbf{J}\mathbf{z} \tag{2.5}$$

where $\mathbf{z} \in \{0,1\}^V$ denotes the system configuration with $z_i = 0$ indicating an empty lattice site and $z_i = 1$ indicating the presence of a particle, V denotes the system volume (i.e., number of available lattice sites), N_d denotes the number of external field controls, $\mathbf{s} \in \mathbb{R}^{N_d}$ denotes the configuration of the external field controls with s_k being the strength of the kth external field, $H_{i,k}$ denotes the interaction between the ith lattice site and the kth external field, $J_{i,j}$ denotes the interaction between the ith and jth lattice sites, and the superscript T refers to the vector transpose. The first sum $E_{\text{ext}}(\mathbf{z})$ represents the external field's contribution to the system energy while the second sum $E_{\text{int}}(\mathbf{z})$ represents the interactions between particles in the system. Higher-order interaction terms can easily be incorporated into this model if necessary (depending on the underlying physics of the system).

The first step of the static problem [233] is to *qualitatively* shape the energy landscape by determining the number N_d and locations of the required well- and barrier-forming point conditions (external controls) to keep the desired self-assembled structure in place (termed the *minimum tiling problem*). The goal of the minimum tiling problem is to reduce the number of degrees of freedom in the system to be practical both from an implementation and optimization point of view. Given the

number N_d and locations of the point-conditions, the second step is to *quantitatively* shape the energy landscape by determining the strengths of these point conditions to maximize the probability of the desired configuration, i.e.,

$$\max_{\mathbf{s}\in\mathbf{S}} p(\mathbf{z}_d, \mathbf{s}) = \max_{\mathbf{s}\in\mathbf{S}} \frac{e^{-\beta E(\mathbf{z}_d,\mathbf{s})}}{\sum_{\mathbf{z}_j\in\Omega_\alpha} e^{-\beta E(\mathbf{z}_j,\mathbf{s})}} \tag{2.6}$$

where $p(\mathbf{z}_d, \mathbf{s})$ denotes the probability of desired state $\mathbf{z}_d$ given external control field $\mathbf{s}$, and $\mathbf{S}$ denotes the set of all possible point condition strength values. The canonical Boltzmann probability distribution function (constant particle number, volume, and temperature) is used here to calculate $p(\mathbf{z}_d, \mathbf{s})$ based on the energy function (2.5). The parameter β determines the "flow" of system through the system phase space (low β implies more accessibility of the states to the system). The denominator represents the *partition function* and is defined as a sum over all configurations Ω_α (see [194] for a detailed discussion on the use of partition functions in the quantitative prediction of self-assembly). Numerous ways to solve this problem are discussed in detail in [233] and the resulting optimal control policy $\mathbf{s}^*$ is applied once the system reaches the desired configuration.

To dynamically force the system to this desired configuration, a method was developed that progressively restricts the system phase space (i.e., allowable configurations) to smaller and smaller regions of the physical domain (see Fig. 2-6) [234]. The basic idea is to systematically break the phase space up into *components* (i.e., subsets of phase space in which all configurations in the subset are accessible from any other configuration in the same subset). This decomposition occurs during distinct time periods denoted as "stages" where the starting stage 0 comprises the entire phase space. At the end of each stage, the starting component will be decomposed into two subsets—one of which will be processed further and another that is neglected. Take, for example, a system described in [234] shown in Fig. 2-6b. Stage 0 shows that the 6 particles can be in any combination of the 16 lattice sites. During stage 1, the objective is to drive the system to have 5 particles in the left half of the domain and 1

particle in the right half of the domain. Although a number of configurations satisfy this requirement, the target configuration is restricted to be within a particular subset of phase space. During each stage, the system is further restricted until the desired configuration is reached in the final stage.

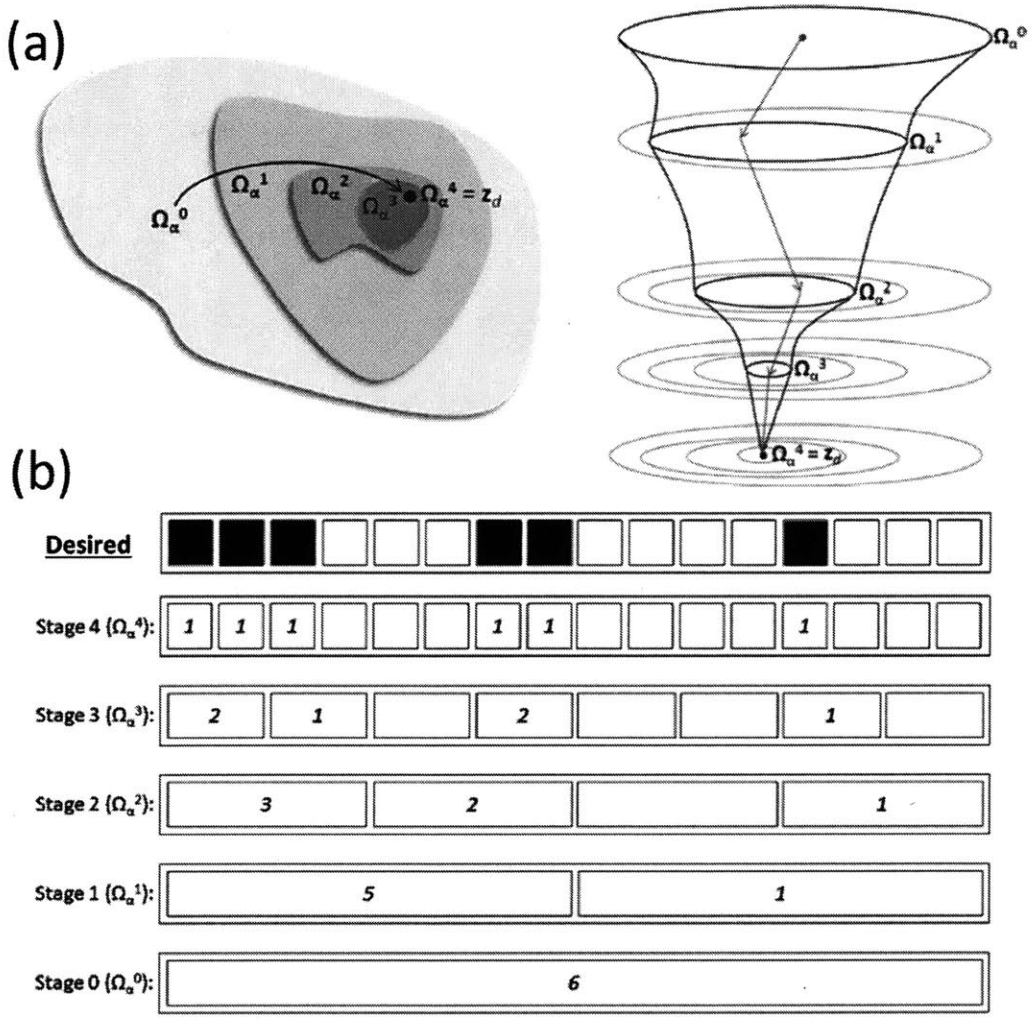


Figure 2-6: Open-loop control policy used to drive the system to a desired arrangement of particles. Please refer to the text in Section 2.3 for detailed descriptions of this figure and the underlying optimization problems that must be solved to derive the control law. (a) Dynamic path that restricts the system to progressively smaller subsets of the system phase space (denoted Ω_α^i for the ith stage). (b) A one-dimensional example of the phase space restriction. Reprinted (adapted) with permission from [234]. Copyright 2010 American Chemical Society.

To achieve this progressive restriction of phase space, Solis et al. [234] proposed solving a series of pseudostatic optimization problems over time until the system reaches equilibrium, with the optimization at stage k being

$$\max_{\mathbf{s}^{(k)} \in \mathbf{S}} p(\Omega_\alpha^{(k)}, \mathbf{s}^{(k)}) = \max_{\mathbf{s}^{(k)} \in \mathbf{S}} \frac{\sum_{\mathbf{z}_j \in \Omega_\alpha^{(k)}} e^{-\beta E(\mathbf{z}_j, \mathbf{s}^{(k)})}}{\sum_{\mathbf{z}_j \in \Omega_\alpha^{(k-1)}} e^{-\beta E(\mathbf{z}_j, \mathbf{s}^{(k)})}}, \tag{2.7}$$

which is essentially maximizing the probability of the system moving into the restricted set of configurations $\Omega_\alpha^{(k)}$ from the set of configurations of the previous stage $\Omega_\alpha^{(k-1)}$ (as the domain is halved at each stage, the resolution of the $\Omega_\alpha^{(k)}$ is $1/2^k$). This optimization looks very similar to that of the static problem (2.6) and the same minimum tiling ideas can be used to determine the number and locations of the external controls $\mathbf{s}^{(k)}$ at each stage. This maximization is a nonlinear nonconvex optimization with a combinatorial number of potential configurations. A "genetic algorithm" is described in [234] for solving this complex problem; however, convergence and global optimality of the solution are not guaranteed.

The lack of measurements in the latter approaches implies that the (stochastic) system evolution is unknown since the positions of the particles are never observed. This no-measurements situation naturally gives rise to stochastic models for describing the system dynamics such as (2.1) and (2.3). In [137], a Master equation model was used in conjunction with the aforementioned methods of [233, 234]. It is shown that open-loop control schemes can drive the self-assembly of the system (in simulations) to a targeted configuration with high probability. The main challenges for this task are estimating the parameters in the Master equation model and designing the physical actuators. These techniques provide a mathematical framework for controlling the self-assembly process and have the potential of constructing systems of higher complexity from simple particles than what is attainable by manipulating global field variables.

Strong specific interactions between particles can be used to achieve self-assembly

of specifically tailored structures (see Section 2.3), however, these design strategies can result in kinetically trapped irregular/undesirable structures that are unable to reorganize on relevant time scales. The prevalence of specific interactions that are weak and reversible, on the other hand, allows dynamic reorganization that can enable systems to explore a wider range of geometric configurations thereby increasing the probability that the thermodynamically favored self-assembly structure will eventually form [224]. Moreover, certain particles have shown the ability to combine local specific order-determining interactions with delocalized non-specific interactions (attraction or repulsion) to achieve robust self-assembly.

Recently, this behavior was studied with the three-legged protein clathrin, which plays an important role in reshaping the cell membrane during endycytosis [224]. Self-assembly of clathrin is stabilized by multiple (specific) weak leg-leg interactions [258]. Three main states were observed during clathrin assembly: monomer, assembled cages, and disordered aggregates. Experimental results from [224] indicate that two distinct kinetic routes occur in clathrin self-assembly. At pH values above the isoelectric point (IEP) of clathrin (pH = 5.8), the assembly proceeds monotonically from monomers to cage structures. At pH values below the IEP, the protein quickly forms disordered aggregates in solution that subsequently form cage structures over time due to large-scale remodelling of the clathrin aggregates. Cryo transmission electron microscopy images of clathrin assemblies at pH = 6.0 and pH = 5.1 from [224] clearly show these distinct paths (Fig. 2-7).

Both of these kinetic routes lead to similar final self-assembled states, which suggests that the mechanism of clathrin assembly has evolved to be robust. Brownian dynamics simulations suggest that stronger non-specific interactions in the system, which are likely caused by clathrin becoming increasingly hydrophobic at pH below the IEP, result in aggregation [224]. These results suggest that non-specific interactions, such as hydrophobic condensation due to local pH changes, can be used as additional degrees of freedom for robustly controlling self-assembly in an open-loop setting (without measurements), and that the use of multiple weak specific interactions can allow the system to escape disordered kinetically trapped states while still

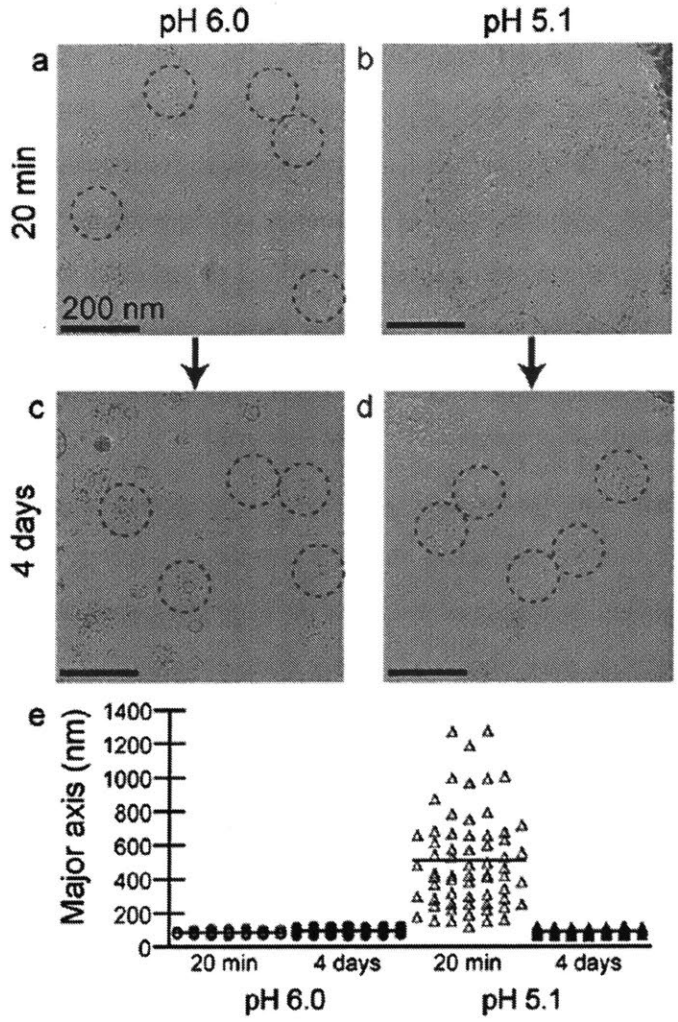


Figure 2-7: Experimental demonstration that clathrin self-assembly is robust to changes in pH due to weak specific interactions. This methodology can be useful for designing open-loop control policies to be more robust. (a and c) Cryo transmission electron microscopy (TEM) image of clathrin self-assembly at pH = 6.0 after 20 minutes and 4 days. (b and d) Cryo TEM image of clathrin self-assembly at pH = 5.1 after 20 minutes and 4 days. Representative spherical cages and disordered aggregates of clathrin are outlined in blue and red, respectively. (e) Dot plot showing the major axis distribution of ellipses fit to the clarthrin assemblies at pH = 6.0 (circles) and pH = 5.1 (triangles) after 20 minutes (open) and 4 days (filled). Solid lines represent the mean value (n = 63, 169, 64, and 54). Reproduced from [224] with permission from The Royal Society of Chemistry.

forming a desired ordered structure. These phenomena have been shown to be important in a number of other systems including, for example, the formation of a kagome

lattice structure from colloids [45] and two-dimensional S-layer protein assembly on supported lipid bilayers [263].

Closed-loop Control

In *closed-loop* control, system output measurements are used to compute the input profiles repeatedly based on the current state of the self-organizing system. Measurement feedback enables effective handling of system stochasticity, as the incorporation of measurements in the control input synthesis robustifies the inputs to uncertainties to a large extent. Such a control approach is capable of coping with the inherent stochastic nature of self-assembly, and facilitates driving the system towards a desired structure in the presence of uncertainties.

Feedback control techniques can improve the performance compared to their open-loop counterparts. To illustrate this point, consider the nucleation of organic compounds (e.g., amino acids, proteins, and active pharmaceutical ingredients) within droplets of solution using a high-throughput microfluidic platform (see Fig. 2-8). The crystallization of crystals in small volumes requires the use of stochastic models (e.g., [87]). The nucleation and growth processes for the microfluidic platform in Fig. 2-8 can be modeled by a Master equation (2.3). When only the number of crystals is considered (i.e., nucleation events), the Master equation can be solved analytically as a function of time [87]. However, when the state is extended to include the length of all crystals (i.e., nucleation and growth), the number of Master equations becomes infeasible to construct/solve directly and a KMC strategy must be used to generate approximate statistical results from the model.

The input to this system is the volume of the droplet, which can be manipulated directly by changing the properties of a local reservoir that sets the evaporation rate (see [246]), and the output of interest is the number of crystals produced in the droplet. The rate expressions and parameter values used in all model simulations are given in [246, 87] for lysozome. For all simulations, the droplet initially has a total volume of 5 μL, a protein loading (C_{protein}) of 18 g/L, a salt concentration (C_{salt}) of 0.36 M, and a solubility curve defined by $C_{\text{protein-eq}} = 1.57C_{\text{salt}}^{2.94}$ g/L, where $C_{\text{protein-eq}}$

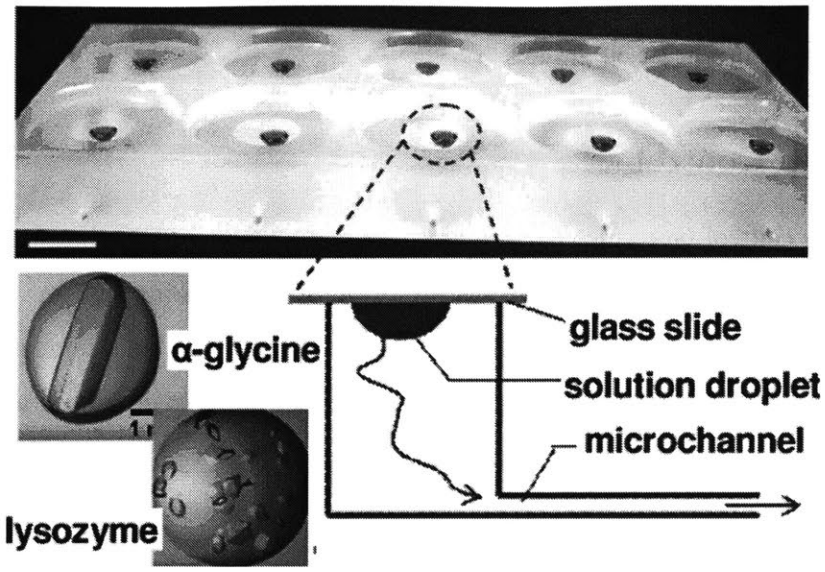


Figure 2-8: Image of a microfluidic platform that uses evaporation to induce nucleation in microliter droplets (see [245, 246]). The evaporation rate in each droplet is specified by the partial pressure of water at the droplet surface, the area and length of the channel that connects the droplet to external air, and the humidity of the external air. This apparatus can be used for open- or closed-loop control based on manipulation of the evaporation rate. Reprinted from [250] with permission from Elsevier.

denotes the equilibrium (saturated) protein concentration.

The desired number of crystals at the final time was chosen to be 1 with a time horizon of 99 hr for all simulations. A proportional controller was used to determine the input applied to the system based on measurements. In the simulations, measurements of the number of crystals in the droplet were taken every 0.1 hr where the crystals could only be observed once they grew larger than 0.1 μm. Fig. 2-9 compares the distribution of the system output for closed- and open-loop operation. The open-loop input profile results in the formation of zero crystals approximately 30% of the time, which is extremely undesirable since that result produces no crystals for analysis and would require repeating the experiment. On the other hand, closed-loop control is able to compensate for stochastic fluctuations in the system (e.g., in the induction time) by altering the droplet volume in response to observing a crystal form, which results in the precipitation of at least one crystal in every simulation.

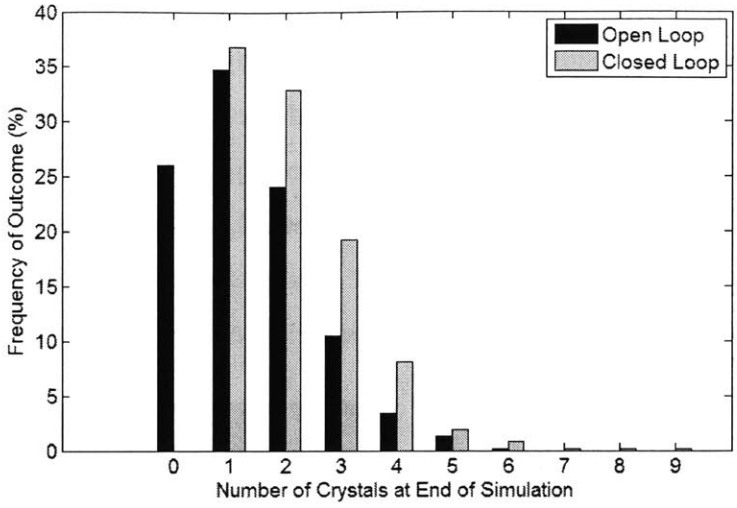


Figure 2-9: Comparison of closed-loop (feedback) and open-loop control policies for microliter droplet crystallization for the system in Fig. 2-8. The output is the number of crystals at the final time in the simulation while the manipulated variable (input) is the volume of the droplet. A proportional controller was used to incorporate feedback with the desired number of crystals (i.e., setpoint) set to 1. The open-loop input trajectory was taken to be the average of the input trajectories for the 1000 closed-loop simulations. Other intuitive open-loop input trajectories (e.g., those corresponding to fastest/slowest induction time in the 1000 simulations) resulted in even worse performance.

In another example, the use of feedback to control the self-assembly of silica colloids was recently demonstrated experimentally [118], with the configuration shown in Fig. 2-10a. The colloid positions are sensed in real time with an optical microscope (Fig. 2-10b) while the electric potential is manipulated to control the crystal assembly process, which is directly tunable from its voltage and frequency dependence (Fig. 2-10cd). A simple proportional control law

$$[V, \omega] = \begin{cases} [-K\Delta\langle C_6\rangle,\ 0.1\ \text{MHz}]; & \Delta\langle C_6\rangle < -0.25 \\ [K\Delta\langle C_6\rangle,\ 1\ \text{MHz}]; & \Delta\langle C_6\rangle \geq -0.25 \end{cases} \tag{2.8}$$

was used to compute the system inputs [118], where V denotes voltage, ω denotes frequency, $K = 4$ V denotes the proportional gain, and $\Delta\langle C_6\rangle = \langle C_6\rangle_{\text{SP}} - \langle C_6\rangle_{\text{PV}}$ where SP denotes setpoint and PV denotes process value (i.e., measurement). The

variable $\langle C_6 \rangle$ is an *order parameter* [118] used to quantify the degree of crystallinity for any configuration of colloidal particles. Specifically, $\langle C_6 \rangle$ denotes the number of hexagonal close-packed (hcp) neighbors around each particle averaged over all particles in a given configuration. The results from applying the feedback law (2.8) to the system in Fig. 2-10a are shown in Fig. 2-11, which demonstrates the controlled assembly and disassembly of a colloidal crystal. The use of feedback allows defects in the crystal to be repaired through partial disassembly, repair, and then re-assembly.

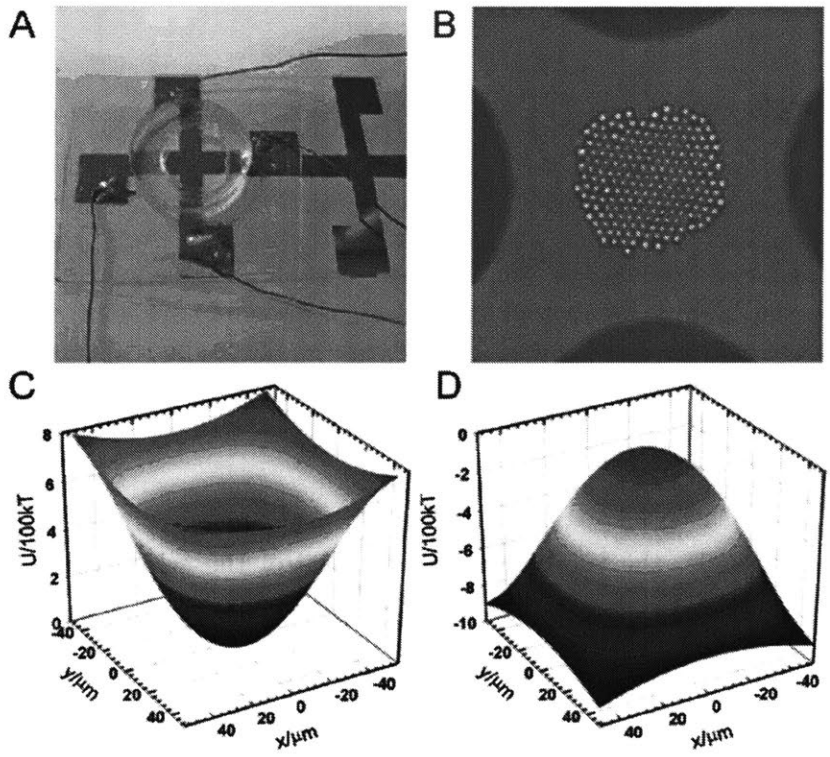


Figure 2-10: Experimental apparatus used for closed-loop control of colloidal assembly. (a) Experimental setup of gold film quadrupole electrodes on a glass slide microscope slide and an o-ring container that houses an aqueous dispersion of silica colloidal particles. (b) Example image taken using optical microscopy used as a sensor for system measurements. Theoretical Cartesian coordinate-based potential energy wells were computed, using the electrode center as a reference, at (c) $V = 4$ V, $\omega = 1$ MHz and (d) $V = 4$ V, $\omega = 0.1$ MHz. Reprinted (adapted) from [118] with permission from John Wiley and Sons.

The use of order parameters in the latter example enables collapsing the massive amount of information in the measurement of these systems (position of all parti-

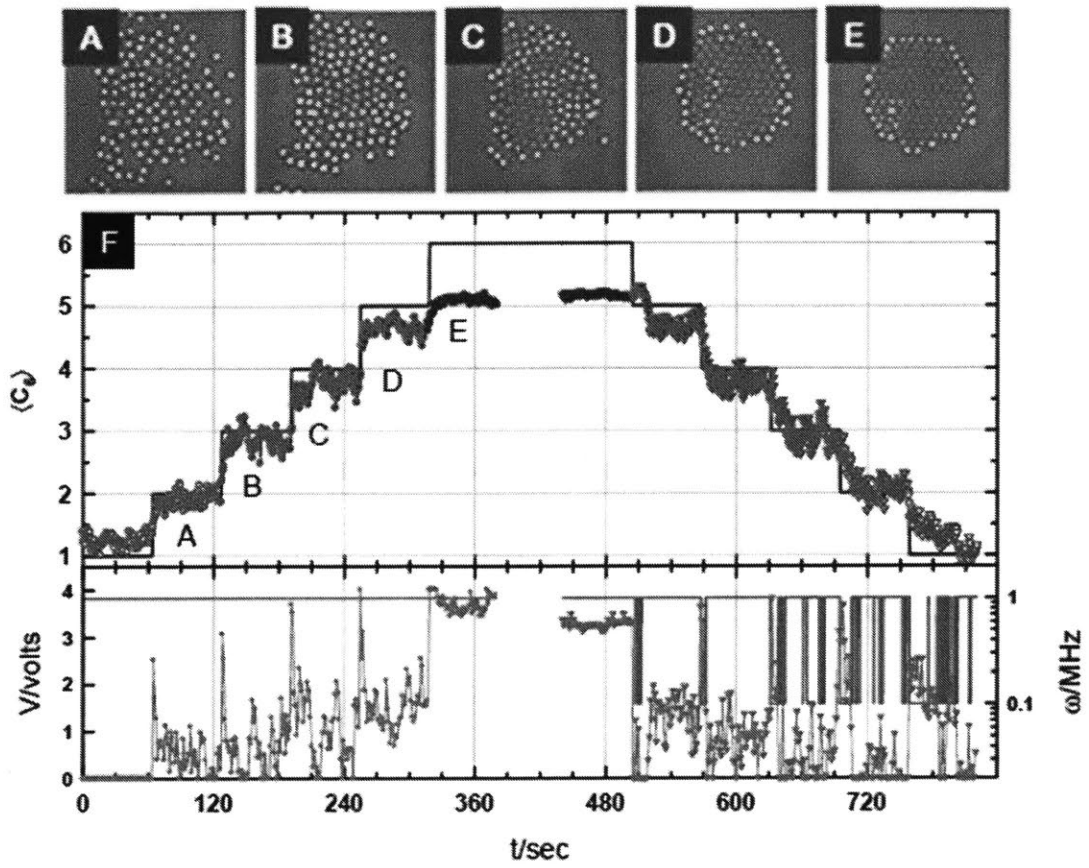


Figure 2-11: Feedback controlled self-assembly of a colloidal crystal containing 130 particles using the control law in (2.8). Optical microscopy measurements with $\langle C_6 \rangle_{SP}$ equal to (a) 2, (b) 3, (c) 4, (d) 5, and (e) 6. (f) Feedback-controlled self-assembly (0 to 360 seconds) and disassembly (480–720 seconds). The top pane shows $\langle C_6 \rangle_{SP}$ (solid blue line) and $\langle C_6 \rangle_{PV}$ (blue points) versus time. The bottom pane shows the manipulated variables voltage V (green line) and frequency ω (orange line) versus time computed using (2.8). Reprinted (adapted) from [118] with permission from John Wiley and Sons.

cles) to a single variable, which simplifies the control strategy at the cost of losing information about the current system state. However, this control strategy may not always be effective, depending on the type of system and order parameter selected. For example, as noted by [118], $\langle C_6 \rangle$-like order parameters may not be useful for condensation processes that terminate in amorphous states because they only take on finite values once geometric ordering occurs.

Advanced model-based control approaches can be implemented in a closed-loop

fashion using available real-time measurements to optimize the process of self-assembly. An example of such control has been demonstrated in [247] where model predictive control was used to drive the self-assembly of the quadrapole system in Fig. 2-10a instead of the proportional control strategy in (2.8). The simulation results indicate that the computed input trajectory is able to accelerate crystallization of the colloidal particles. However, this acceleration is achieved at the price of the computational cost of optimizing over the Langevin-based model. This example demonstrates that the tradeoff between controller complexity and performance can be an important issue in the context of controlling micro- and nano-scale self-assembly.

2.4 Outlook for Future Research

The assembly of a large number of small particles into complex ordered structures can aid in the bottom-up engineering of devices with novel characteristics. The primary challenges in the control of micro- and nano-scale self-organizing systems are high-dimensional stochastic dynamics, lack of sensing, limited actuation, and the formation of kinetically trapped configurations. This chapter provides an overview of the recent developments in active self-assembly and open- and closed-loop control of small-scale self-assembly systems.

A promising research direction is the design of particles that enables active control of particle interactions and increases particle specificity toward forming a desired structure. The design of particles for active self-assembly induces what can be interpreted as *internal feedback* where each particle responds to its local environment by exchanging information between different components of the system. Internal feedback allows the particles to reach a consensus about the desired structure of the self-organizing system based on the state of all particles. Such an approach offers vast opportunities for sensing and actuation at the micro- and nano-scale.

Actuators can be introduced either globally (changes felt throughout the system) or locally (changes felt only in a local environment) to drive the system toward the desired ordered structure. Simple open-loop control policies have been developed

to facilitate self-assembly for a variety of systems by manipulating global fields of such variables as temperature [243] and pH [224]. Examples of these strategies were discussed in detail in Section 2.3. A major drawback of these open-loop strategies is their lack of robustness such that small perturbations during the self-assembly process can lead to undesired structures (e.g., kinetically trapped states). Recent work on the self-assembly of the protein clathrin suggests that these open-loop strategies can be made more robust by replacing strong specific interactions between particles with multiple weak specific interactions [224]. These weak interactions allow the system to escape kinetically trapped states during the assembly process while still forming the desired self-assembled structure.

An alternative method to create more robust open-loop control policies is to introduce local actuation at the micro- and nano-scale. Local manipulation of fields (e.g., temperature, concentration, pH, electric field, and magnetic field) has been shown in simulations to drive the system toward a particular desired configuration with a high probability [233, 234, 137]. Methods for qualitative and quantitative shaping of the energy landscape through manipulation of these fields were reviewed in Section 2.3.

Currently, it is common to apply only one or two of these approaches at a time. However, integrating global and local actuators with "optimally-designed" particles could enable precise control of extremely complex structures during the self-assembly process. We believe this is a key direction for future research in controlled self-assembly.

Another promising research direction is the use of model-based control to optimize the self-assembly process through manipulation of local and/or global system inputs. In particular, model predictive control can handle complex multivariate dynamics and competing sets of objectives while satisfying system constraints. The control performance largely depends on the quality of the system model, which indicates that the development of compact models for complex system dynamics in self-assembly is a key need. As discussed throughout this chapter, common models for these systems include the Langevin equation, the Master equation, and partition functions, whose dimensionality/complexity limit their applicability for many systems of practical im-

portance.

The use of feedback (measurements) in the control policy is another way to make the system more robust to stochasticity and disturbances by counteracting this uncertainty with available measurements from real-time sensors. These closed-loop approaches require the development of advanced sensing technologies to monitor local system properties (e.g., positions of all particles). Real-time estimation of the system state is required when implementing most advanced model-based control strategies. A key challenge in this space is being able to utilize the system measurement within a complex model of the self-assembly process in real-time. One alternative, discussed in Section 2.3, is the use of order parameters to collapse the large amount of detailed information within the measurement to a relatively small number of variables. In addition, models in terms of these order parameters (see e.g., [247]) can be much cheaper to evaluate than their full-information counterparts. Another alternative is to employ advanced control strategies whose online computations are based purely on input-output models used in concert with output estimation. Control algorithms that employ this approach for systems with stochastic uncertainties [169] may be promising for application to self-assembly.

The development and use of nano-scale sensors for control of micro- and nano-scale self-assembly is an important route to explore. Some recent examples of nano-scale sensors include: (1) single-walled carbon nanotubes (SWCNTs), which exhibit discrete changes in the fluorescence signal when molecules adsorb and desorb from the SWCNT surface [51], (2) asymmetric catalytic particles that use chemical reactions for self-propulsion so that the speed of the catalyst is indicative of the local concentration [119], and (3) gold nanoshells with a pH-sensitive adsorbate that can be used as a standalone all-optical nano-scale pH meter [25]. Experimental verification of advanced control methods utilizing these types of nano-scale sensors remains largely unexplored in the area of controlled self-assembly.

2.5 Importance of Uncertainty: An Illustration

Uncertainties, such as unmeasured disturbances, unmodeled dynamics, faults, and unknown parameter values, are important in a large number of self-assembly processes as they represent a diverse set of applications. In particular, unknown parameter values in first principles models (for example, unknown rate constants in a reaction network that is carried out in a continuously stirred tank reactor) are the main focus of this thesis. To illustrate how parametric uncertainty can greatly impact optimal control strategies, a batch crystallization example (i.e., a type of self-assembly process) is presented below.

Moment Model for Seeded Crystallization

Crystallization is an industrially important unit operation for attaining high purity separation. Control of the crystal size distribution (CSD) can be extremely important for efficiency of downstream processes and product quality (e.g., tablet stability, dissolution rate, activity). A large portion of batch crystallizers are seeded to better control the CSD. The most common control problem is to determine the temperature profile that minimizes the ratio of nucleated to seed crystal mass subject to meeting a mass yield constraint at the end of the batch. The CSD can be modeled using a population balance equation (PBE) approach, which is a integro-algebraic partial differential equation (PDE). When the growth rate is size-independent, the PBE can be replaced by a set of ordinary differential equations (ODEs) using the method of moments. As shown in [48], the set of ODEs can be written as

$$\dot{x}(t) = f(x(t), u(t), \theta), \tag{2.9}$$

where $x = [\mu_0, \mu_1, \mu_2, \mu_3, C, \mu_{\text{seed},0}, \mu_{\text{seed},1}, \mu_{\text{seed},2}, \mu_{\text{seed},3}]^\top$ is the state, μ_j is the j^{th} moment of the CSD, C is the solute concentration, and $\mu_{\text{seed},j}$ is the j^{th} moment of

the seed CSD initially put into the crystallizer. The nonlinear function is given by

$$f(x(t), u(t), \theta) = \begin{bmatrix} B \\ G\mu_0 \\ 2G\mu_1 \\ 3G\mu_2 \\ -\rho_c k_v h(3G\mu_2) \\ G\mu_{\text{seed},0} \\ 2G\mu_{\text{seed},1} \\ 3G\mu_{\text{seed},2} \end{bmatrix}, \tag{2.10}$$

where ρ_c is the density of the crystal, k_v is a volumetric shape factor, and h is a units conversion factor. Many different models for the crystal growth rate (G) and nucleation rate (B) have been proposed in the literature. The most common kinetic models when nuclei form from existing crystals (secondary nucleation) are

$$G = k_g S^g, \tag{2.11}$$

$$B = k_b k_v S^b \mu_3, \tag{2.12}$$

where k_g, g, k_b, and b are the kinetic parameters for growth and nucleation, respectively, $S = (C - C_{\text{sat}})/C_{\text{sat}}$ is the relative supersaturation, and C_{sat} is the saturation concentration assumed to be fit to a quadratic model $A_0 + A_1 T(t) + A_2 T^2(t)$ where $u(t) = T(t)$ is the crystallizer temperature (the manipulated variable in this problem).

Characterizing Parameter Uncertainty

This case study uses parameters for an industrial-scale process involving the crystallization of potassium nitrate from water, as reported in [171] (see Table 2.1). As typically done in the literature, assume that only the growth and nucleation parameters need to be estimated from data, i.e., define $\theta = [g, \ln(k_g), b, \ln(k_b)]^\top$. Due to measurement noise, model errors, and having a finite amount of data, the estimates of these parameters will not be exact. The majority of parameter estimation algorithms

quantify uncertainty in terms of a multivariate normal distribution of the form

$$f_\theta(\theta) = \frac{1}{(2\pi)^{n_\theta/2}|V_\theta|^{1/2}} \exp\left(-\frac{1}{2}\left[(\theta-\hat{\theta})^\top V_\theta^{-1}(\theta-\hat{\theta})\right]\right), \tag{2.13}$$

where $\hat{\theta}$ is the nominal parameter vector (values shown in Table 2.1) and V_θ is the positive definite covariance matrix. The uncertainty of the nucleation and growth kinetics is characterized (from experimental data) by the covariance matrix [171]

$$V_\theta^{-1} = \begin{bmatrix} 102873 & -21960 & -7509 & 1445 \\ -21960 & 4714 & 1809 & -354 \\ -7509 & 1809 & 24225 & -5198 \\ 1445 & -354 & -5198 & 1116 \end{bmatrix}. \tag{2.14}$$

Optimal Temperature Profile

The optimal control problem of interest is given by

$$\begin{aligned} \min_{T(t)} \quad & J = \frac{\text{nucleated crystal mass}}{\text{seed crystal mass}} = \frac{\mu_3(t_f) - \mu_{\text{seed},3}(t_f)}{\mu_{\text{seed},3}(t_f)}, \\ \text{subject to:} \quad & \text{model equations (2.9)}, \ \forall t \in [0, t_f] \\ & T_{\min} \le T(t) \le T_{\max}, \ \forall t \in [0, t_f] \\ & R_{\min} \le \frac{dT(t)}{dt} \le R_{\max}, \ \forall t \in [0, t_f] \\ & C(t_f) \le C_{\text{final,max}}, \end{aligned} \tag{2.15}$$

where t_f is the time of the batch, $T_{\min}$, $T_{\max}$, $R_{\min}$, and $R_{\max}$ are the minimum and maximum temperatures and temperature ramp rates, respectively, and $C_{\text{final,max}}$ defines the minimum yield requirement. To obtain a finite number of decision variables in the optimization problem, $T(t)$ was parametrized as a piecewise affine function with eight discretization intervals. The optimal temperature profile, obtained using a standard nonlinear programming solver with $\theta = \hat{\theta}$, is shown in Figure 2-12. For the nominal system, this profile results in a nucleated to seed mass ratio of 4.75, which is a decrease of more than 15% when compared to a common industrial profile

consisting of two stages of linear cooling with different rates.

Although the optimal nominal temperature profile decreases J compared to the two-stage linear profile when the parameters are at the nominal values, it is important to see how the system responds to realistic uncertainty in the nucleation and growth parameters as defined by the probability density function (PDF) (2.13). This analysis involves propagating this PDF through the nonlinear dynamics to the objective of interest. For this example, polynomial chaos is used as the uncertainty propagation tool (see Chapter 3 for details). The PDFs of the nucleated to seed crystal mass ratio over time for the optimal nomimal temperature trajectory and the two-stage linear cooling profile are shown in Figures 2-13 and 2-14, respectively. The optimal nominal temperature profile (designed for nominal parameter values) produces a significant spread in J, i.e., the sample standard deviation of J is 0.52. Compare this value to the two-stage linear cooling profile typical of industrial practice, which has a larger nominal objective value of 5.65, but a smaller standard deviation of only 0.13.

This analysis indicates that the optimal profile may only *nominally* give better performance. The system under the optimal nominal design may be very sensitive to uncertainties in the parameter values, and could produce worse performance than not doing optimization at all. This example motivates the development of methodology that can robustly account for uncertainties directly in the formulation of the optimal control problem. Such a methodology would allow for one to trade off performance and robustness in a systematic way based on actual uncertainties in the model. This methodology, developed throughout the subsequent chapters of this thesis, is one of the main contribution of this work.

Table 2.1: Parameters used in the batch cooling crystallization optimal control case study. The parameters and model were adapted from [171], which is based on an industrial crystallization of potassium nitrate from water.

Process Parameters	Symbol	Value	Units
growth exponent	g	1.31	–
growth coefficient	k_g	6.5682×10^{-3}	$\frac{\text{m}}{\text{min}}$
nucleation exponent	b	1.84	–
nucleation coefficient	k_b	3.5321×10^{13}	$\frac{\#}{\text{m}^3\text{min}}$
crystal density	ρ_c	2.11×10^{6}	$\frac{\text{g(solute)}}{\text{m}^3}$
volume shape factor	k_v	1	–
conversion factor	h	1.5062×10^{-6}	$\frac{\text{m}^3\text{(slurry)}}{\text{g(solvent)}}$
saturation parameters	A_0	1.286×10^{-1}	$\frac{\text{g(solute)}}{\text{g(solvent)}}$
	A_1	5.88×10^{-3}	$\frac{\text{g(solute)}}{\text{g(solvent)}^\circ\text{C}}$
	A_2	1.721×10^{-4}	$\frac{\text{g(solute)}}{\text{g(solvent)}^\circ\text{C}^2}$
Initial Conditions	Symbol	Value	Units
initial concentration	C_0	0.5	$\frac{\text{g(solute)}}{\text{g(solvent)}}$
initial moments	$\mu_{0,0}$	3×10^{6}	$\frac{\#}{\text{m}^3}$
	$\mu_{1,0}$	1×10^{2}	$\frac{\text{m}}{\text{m}^3}$
	$\mu_{2,0}$	1×10^{-2}	$\frac{\text{m}^2}{\text{m}^3}$
	$\mu_{3,0}$	1×10^{-6}	$\frac{\text{m}^3}{\text{m}^3}$
time to startup	t_f	160	min
Constraint Parameters	Symbol	Value	Units
maximum final concentration	$C_{\text{final,max}}$	0.4292	$\frac{\text{g(solute)}}{\text{g(solvent)}}$
minimum temperature	T_{min}	28	°C
maximum temperature	T_{max}	32.4132	°C
minimum temperature ramp rate	R_{min}	-0.4	$\frac{^\circ\text{C}}{\text{min}}$
maximum temperature ramp rate	R_{max}	0.4	$\frac{^\circ\text{C}}{\text{min}}$

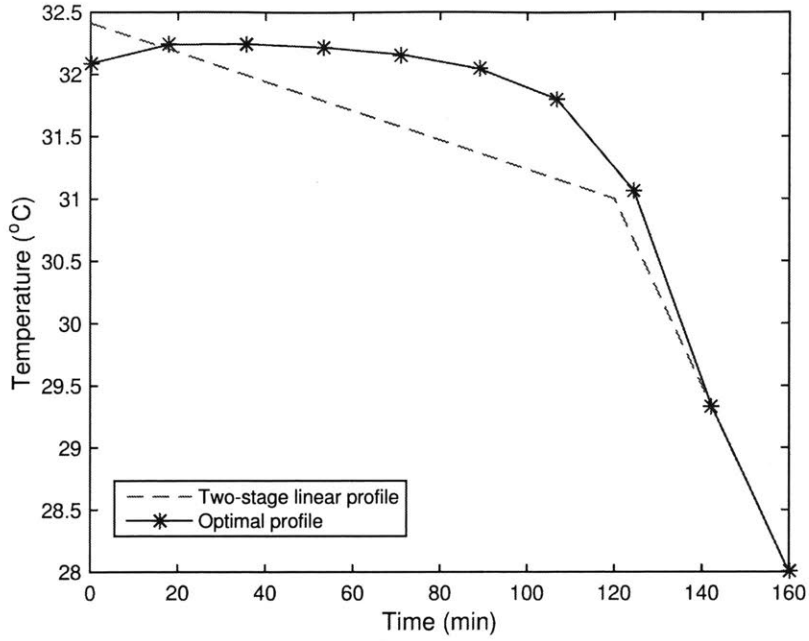


Figure 2-12: Optimal temperature profile for the crystallizer problem in Section 2.5. The temperature trajectory was approximated with a piecewise linear function with 8 discretization intervals. Also, an industrially common two-stage linear cooling profile is shown for comparison.

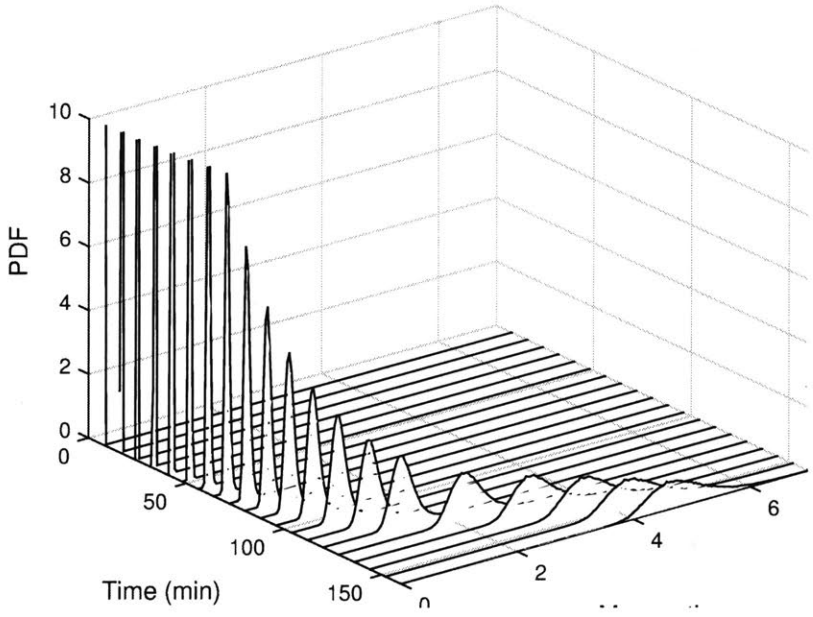


Figure 2-13: Nucleated to seed mass ratio distributions for the optimal temperature profile. The distribution at the final time has significant variation due to uncertainty in the nucleation and growth parameters, which indicates that this "optimal" profile is not robust to uncertainty.

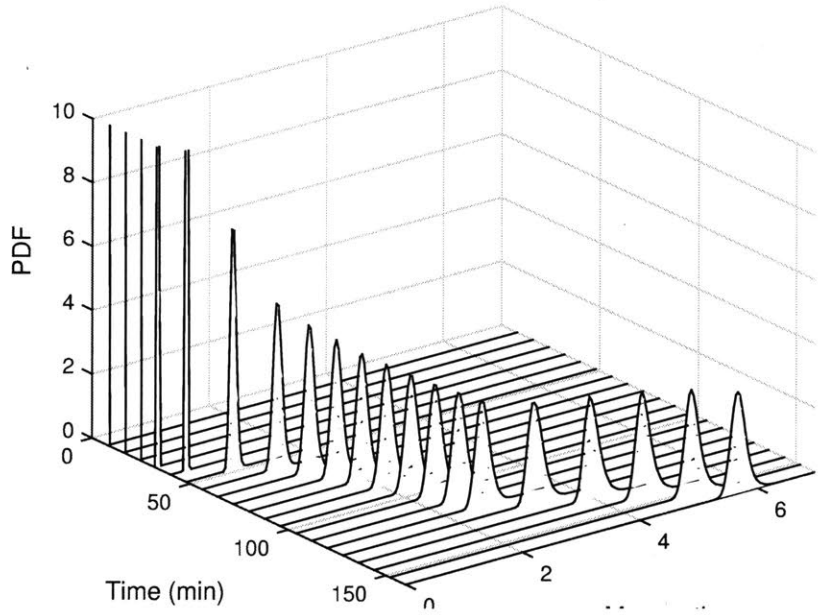


Figure 2-14: Nucleated to seed mass ratio distributions for the two-stage linear cooling profile (commonly type of profile implemented in industry). Although the objective is 15% higher than the optimal profile when the system is at nominal parameter values, the profile has significantly less variation due to parametric uncertainties.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Polynomial Chaos Framework

3.1 Introduction to Probability Theory

Random Variables

A random variable $X : \Omega \to \mathcal{S}$ is a measurable function from the set of all possible outcomes Ω to a set $\mathcal{S}$. A formal (axiomatic) way to define random variables can be done using *measure theory*, which is beyond the scope of this thesis. It suffices to say that a *probability space* (aka probability triple) is the mathematical construct used to model real-world processes/experiments. Probability spaces consist of three parts:

- A sample space, Ω, which is the set of all possible outcomes of the experiment.
- A set of events, $\mathcal{B}$, where each event in this space contains zero or more outcomes (for example, the sample space is an element of the event space $\Omega \in \mathcal{B}$). This is needed since more complex events can be described by groups of individual outcomes. The collection of all such events is a σ-algebra $\mathcal{B}$.
- The measure function, $P : \mathcal{B} \to [0, 1]$, which specifies the likelihood/probability of each event happening.

For a set $A \subseteq \mathcal{S}$, the notation $\mathbb{P}(X \in A)$ is used as a shorthand for $P(\{\omega \in \Omega : X(\omega) \in A\})$. When X is a real-valued random variable, the set $\mathcal{S}$ is equal to $\mathbb{R}$.

By recording the probability of every output range of a real-value random variable X, we can construct its probability distribution. The probability distribution effectively "forgets" about the underlying probability space used to define X and instead pushes the measure P on Ω to a measure F_X on $\mathbb{R}$ where F_X is the cumulative distribution function

$$F_X(x) = \mathbb{P}(X \leq x), \quad \forall x \in \mathbb{R}. \tag{3.1}$$

The probability density function (PDF) is commonly used to characterize continuous random variables and describes how likely X takes on a value in a particular interval

$$\mathbb{P}(a \leq X \leq b) = \int_a^b f_X(x)dx, \tag{3.2}$$

where the PDF is defined by

$$f_X(x) = \frac{d}{dx}F_X(x). \tag{3.3}$$

These concepts extend straightforwardly to random vectors. Interested readers are referred to [22, 271] for further details on probability and measure theory.

Moments

One of the most common ways to describe a PDF is in terms of its moments. These can be used to characterize/parametrize a distribution as done in, for example, Gaussian random variables which are uniquely described by their first and second moment. Moments are used heavily throughout this thesis in both estimation and control as they represent a compact way to describe process variation without the need for the full PDF. Additionally, often statistical algorithms can be greatly simplified by considering propagation of only moments of distributions. A well-known example is the Kalman filter, which describes the evolution of an estimated state vector based on process measurements in the presence of Gaussian white noise [120].

The n^{th} order (raw) moment of a random variable X (with support $\mathcal{S}$ and PDF

$f_X(x)$) is defined as

$$\mathbb{E}[X^n] = \int_S x^n f_X(x) dx, \tag{3.4}$$

where $\mathbb{E}[\cdot]$ is the expectation operator. The first moment ($m = 1$) is commonly referred to as the mean/expectation of X. The variance (or second central moment) is defined as

$$\mathbf{Var}[X] = \int_S (x - \mathbb{E}[x])^2 f_X(x) dx. \tag{3.5}$$

Joint, Marginal, and Conditional Probability Distributions

Given two or more random variables $X_1, \cdots, X_n$, defined on a shared probability space, the joint probability distribution describes the probability that each $X_1, \cdots, X_n$ falls in a particular set of values, which can be described mathematically by a joint cumulative distribution function

$$F_{X_1,\cdots,X_n}(x_1, \cdots, x_n) = \mathbb{P}(X_1 \leq x_1, \cdots, X_n \leq x_n), \tag{3.6}$$

or a joint probability density function

$$f_{X_1,\cdots,X_n}(x_1, \cdots, x_n) = \left. \frac{\partial^n F}{\partial x_1 \cdots \partial x_n} \right|_{x_1,\cdots,x_n}. \tag{3.7}$$

The PDF of each individual random variable $f_{X_i}(x_i)$ for $i = 1, \cdots, n$ is termed the *marginal density function* and is calculated by integrating over all other values of the random variables

$$f_{X_i}(x_i) = \int f_{X_1,\cdots,X_{i-1},X_{i+1},\cdots,X_n}(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_n, \tag{3.8}$$

where the multidimensional integral is over the proper support. The conditional PDF of X given the occurrence of a value of the random variable $Y = y$ is defined as

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}, \tag{3.9}$$

for any $f_Y(y) > 0$. Two random variables are *independent* if and only if the conditional distribution of X given Y is equal to the marginal distribution for all possible realizations of Y, i.e., $f_{X|Y}(x|y) = f_X(x)$ for all possible x and y.

Bayes' Theorem

Bayes' theorem describes how two conditional PDFs are related and can be stated mathematically as

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)}. \tag{3.10}$$

This expression can easily be derived by substituting the definitions of $f_{X|Y}(x|y)$ and $f_{Y|X}(y|x)$ from (3.9) into each other. The denominator of this expression is a normalization constant that is simply the marginal PDF of Y defined by (3.8). This concept easily extends to random vectors of arbitrary length (see, e.g., [88] for details). One of the most common applications of Bayes' theorem is the quantification of parameter uncertainty from data using a statistical model of the process. This is described in the following section as a precursor to uncertainty propagation. Note that an algorithm for online recursive Bayesian estimation is presented in Chapter 9 of this thesis.

3.2 Quantification of Uncertainty from Data

Preliminaries

In this section, capital letters will denote random variables/vectors. Let $f_X(\cdot)$ denote the probability density function (PDF) of a random variable X and $f_{X|Y}(x|y)$ denote

the conditional PDF of random variable X given information Y (where X and Y can be vectors). A vector of random variables $X \in \mathbb{R}^n$ follows a multivariate normal distribution with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n \times n}$ if its PDF has the form

$$f_X(x) = (2\pi)^{n/2}|\Sigma|^{-1/2} \exp\left[-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right], \tag{3.11}$$

where $|\cdot|$ denotes the determinant of a square matrix.

Bayesian Estimation Framework

While some parameters are known or can be directly measured easily with currently available sensor technology (e.g., mass or density), other parameters must be estimated from experimental data based on a model of the system. Some examples include kinetic parameters in a chemical reaction network, growth and nucleation kinetic parameters in a crystallizer, and transition probabilities in a self-assembling colloidal suspension. Determining the best possible values for the unknown parameters is a problem commonly referred to as *parameter estimation*, which has been an important topic in the statistics field for many decades.

In Bayesian statistics, unknown model parameters $\mathbf{P} \in \mathbb{R}^{n_p}$ are viewed as random variables that are related to experimental measurements/data $\mathbf{Y} \in \mathbb{R}^{n_m}$ (commonly a vector composed of repeatedly measured output values at various sample times) where n_p is the number of parameters and n_m is the number of measured data points. Using Bayes' theorem, we can write the *a posteriori* distribution as a function of the *a priori* distribution $f_{\mathbf{P}}(\mathbf{p})$ and the likelihood function $f_{\mathbf{Y}|\mathbf{P}}(\mathbf{y}|\mathbf{p})$ [13]

$$f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y}) = \frac{f_{\mathbf{Y}|\mathbf{P}}(\mathbf{y}|\mathbf{p}) f_{\mathbf{P}}(\mathbf{p})}{f_{\mathbf{Y}}(\mathbf{y})}, \tag{3.12}$$

where $f_{\mathbf{Y}}(\mathbf{y})$ is a normalization constant to ensure the *a posteriori* distribution integrates to one. The *a posteriori* distribution is defined by $f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y})$, which gives the PDF of the parameters based on all experimentally observed data. This captures all available information about the parameters. To compute the *a posteriori* distribu-

tion, one must select the prior $f_{\mathbf{P}}(\mathbf{p})$ (based on previous experimental runs and/or knowledge of the process) and the likelihood $f_{\mathbf{Y}|\mathbf{P}}(\mathbf{y}|\mathbf{p})$ that essentially describes how well a particular set of parameter values matches a particular set of observations, which should be derived from a model of the process.

For realistic problems, (3.12) cannot be solved analytically and the problem must be approached numerically. Markov chain Monte Carlo (MCMC) is a sampling-based approach for constructing the *a posteriori* distribution numerically. See, for example, [102] for a discussion on MCMC methods for estimation of kinetic parameters in polymorphic crystallization.

Maximum A Posteriori Estimation

MCMC (and similar sampling-based methods) can be quite computationally expensive as they require one to simulate a model of the system many different times for a number of different parameter values. To reduce the computational burden, often engineers are interested in *point estimates* which represent the "best estimate" of the unknown parameters. One of the most common point estimates is typically referred to as the maximum a posteriori (MAP) estimate, denoted as $\mathbf{p}^\star$ in this chapter, which is the parameter value that maximizes $f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y})$. The MAP estimate is one of the modes of $f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y})$ such that it satisfies [94]

$$\nabla_{\mathbf{p}} f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y})\big|_{\mathbf{p}^\star} = 0. \tag{3.13}$$

where $\nabla_x f(x)$ denotes the Jacobian of the function $f(x)$ with respect to x. The MAP estimate can then be derived by solving a (possibly) nonlinear optimization problem.

MAP Estimation in the Presence of Gaussian White Noise

The objective function of the MAP problem can be greatly simplified when certain (very common) assumptions are made about the system as detailed below. Let $g(\mathbf{x}, \mathbf{P})$ be a model of the process data usually derived from the physics of the problem where $\mathbf{x}$ are the independent variables (e.g., initial conditions and manipulated input values).

Experimental errors $\mathbf{E}$ are assumed to be additive and normally distributed with zero-mean and covariance $\Sigma_{\mathbf{e}}$ such that

$$\mathbf{Y} = g(\mathbf{x}, \mathbf{P}) + \mathbf{E}. \tag{3.14}$$

The model $g(\mathbf{x}, \mathbf{P})$ can be derived from, for example, ordinary or partial differential equations that describe the state evolution of the process. Assuming that there are no errors in g or $\mathbf{x}$ (meaning that the structure of the model is correct), the likelihood function corresponds to that of a normal PDF

$$f_{\mathbf{Y}|\mathbf{P}}(\mathbf{y}|\mathbf{p}) = (2\pi)^{n_m/2}|\Sigma_{\mathbf{e}}|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{y} - g(\mathbf{x}, \mathbf{p}))^{\top}\Sigma_{\mathbf{e}}^{-1}(\mathbf{y} - g(\mathbf{x}, \mathbf{p}))\right]. \tag{3.15}$$

It is also assumed that the prior distribution is a normal PDF

$$f_{\mathbf{P}}(\mathbf{p}) = (2\pi)^{n_p/2}|\Sigma_{\mathbf{p}}|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}})^{\top}\Sigma_{\mathbf{p}}^{-1}(\mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}})\right], \tag{3.16}$$

where $\boldsymbol{\mu}_{\mathbf{p}}$ and $\Sigma_{\mathbf{p}}$ are the mean and covariance matrix of the prior parameter distribution, respectively. Lastly, it is assumed that the parameters and measurements are uncorrelated, that is $\text{cov}(\mathbf{P}, \mathbf{E}) = 0$.

Using the property that the maximizer of a function is also a maximizer of any monotonic transformation of that function, we can write

$$\mathbf{p}^{\star} = \arg\max_{\mathbf{p}} f_{\mathbf{P}|\mathbf{Y}}(\mathbf{p}|\mathbf{y}) = \arg\max_{\mathbf{p}} \left[\ln f_{\mathbf{Y}|\mathbf{P}}(\mathbf{y}|\mathbf{p}) + \ln f_{\mathbf{P}}(\mathbf{p})\right].$$

Substituting (3.15) and (3.16) into this expression, the optimization problem to determine the MAP estimates reduces to the following form (whenever $\Sigma_{\mathbf{e}}$ and $\Sigma_{\mathbf{p}}$ are independent of $\mathbf{p}$)

$$\min_{\mathbf{p}\in\mathscr{P}} \left\{(\mathbf{y} - g(\mathbf{x}, \mathbf{p}))^{\top}\Sigma_{\mathbf{e}}^{-1}(\mathbf{y} - g(\mathbf{x}, \mathbf{p}) + (\mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}})^{\top}\Sigma_{\mathbf{p}}^{-1}(\mathbf{p} - \boldsymbol{\mu}_{\mathbf{p}})\right\}, \tag{3.17}$$

where $\mathscr{P}$ represent (optional) constraints on the possible parameter values. Ideally, these constraints (such as requiring the diffusion coefficient to be positive) would

be enforced through the definition of the prior; however, often the prior is selected to be a normal distribution for simplicity, as done in this analysis, which has an unbounded support. In the absence of prior information (such that $\Sigma_{\mathbf{p}}^{-1} = 0$), the MAP problem reduces to the well-known maximum likelihood estimation problem that is fully captured by the first term of the objective function.

Quantifying Uncertainty in Parameter Estimates

The accuracy of the estimated parameter values $\mathbf{p}^\star$ can be quantified in a number of different ways. Due to the stochastic fluctuations associated with measurements, the parameter estimates are also stochastic variables with probability distributions. Ideally, we would capture the full *a posteriori* PDF using, e.g., a MCMC method; however, as mentioned previously, this can be computationally prohibitive in certain applications. An alternative method is to use multivariate statistics applied to a local linearization of the model accurate in the vicinity of the estimates $\mathbf{p}^\star$ such that [13]

$$g(\mathbf{x}, \mathbf{p}) \approx g(\mathbf{x}, \mathbf{p}^\star) + \mathbf{S}(\mathbf{p} - \mathbf{p}^\star). \tag{3.18}$$

where $\mathbf{S} = \nabla_{\mathbf{p}} g(\mathbf{x}, \mathbf{p})|_{\mathbf{p}^\star}$ is the sensitivity matrix of the model with respect to the parameters, which can either be calculated analytically or estimated numerically by finite differences. When the measurement errors are independent and normally distributed, the covariance matrix of the parameter estimates can be approximated from the linearized model by [13]

$$\Sigma_{\mathbf{p}^\star}^{-1} \approx \mathbf{S}^\top \Sigma_{\mathbf{e}}^{-1} \mathbf{S} + \Sigma_{\mathbf{p}}^{-1}. \tag{3.19}$$

The approximate $100(1-\alpha)\%$ confidence region is the hyperellipsoid defined by

$$\mathscr{P}_{\text{ellipsoid}}^\star = \left\{ \mathbf{p} \in \mathbb{R}^{n_p} \mid (\mathbf{p} - \mathbf{p}^\star)^\top \Sigma_{\mathbf{p}^\star}^{-1} (\mathbf{p} - \mathbf{p}^\star) \leq \chi_{n_p}^2(\alpha) \right\}, \tag{3.20}$$

where $\chi_{n_p}^2(\alpha)$ is the chi-squared distribution with n_p degrees of freedom for a given $100(1-\alpha)\%$ confidence region. Note that the eigenvectors of $\Sigma_{\mathbf{p}^\star}^{-1}$ give the direction

while the eigenvalues give the length of the axes of the hyperellipsoid. Also, note that the sensitivity values in the matrix **S** can be used to determine what parameters have the largest effect on the output. This is intimately tied to experimental design and uncertainty management and evaluation (illustrated in Figure 1-2).

Regardless of the method used to approximate/characterize the parameter uncertainty, this uncertainty must be propagated through the system dynamics to understand its effect on important states/outputs of the system. Furthermore, this information must be incorporated into the model predictions utilized in model-based control in order to manage the control actions taken based on all currently available knowledge. This concept was illustrated on a self-assembly process in Section 2.5.

The rest of this chapter focuses on various uncertainty propagation techniques, in which a parameter PDF (characterized by an uncertainty quantification method run using a particular set of data) is propagated through the model equations to obtain PDFs of the important model outputs. A number of uncertainty propagation tools/techniques have been developed by various communities (e.g., computational fluid dynamics [183] and CO_2 sequestration [280]). Since the main focus of this thesis is on real-time process control, a particular emphasis is given to polynomial chaos methods as they can be more computationally efficient than traditional Monte Carlo methods, which are extremely expensive in certain applications, in addition to having high accuracy and nice theoretical properties.

3.3 Power Series Methods

For the remainder of this chapter, the PDF of a random variable x will be denoted as $f(x)$, so that no distinction is made between the PDF function argument and the random variable itself. Also, there will be no particular nomenclature used to denote a variable as random, and each defined variable should be understood to be random (as opposed to just a particular realization of the random variable) by context.

Define a perturbed model parameter vector of dimension n_θ as

$$\theta = \hat{\theta} + \delta\theta \tag{3.21}$$

where $\hat{\theta} \in \mathbb{R}^{n_\theta}$ is a nominal/mean of the model parameters and $\delta\theta$ is the perturbation about $\hat{\theta}$. As discussed in the previous section, the parameter estimates are random variables due to stochastic fluctuations in the process measurements. From now on, it is assumed that the uncertainty in these estimates has already been quantified.

Let $\hat{y}$ be some scalar output of interest in the process with the nominal parameter values, y is its value for the perturbed parameters θ, and $\delta y = y - \hat{y}$ is their difference. The power series approach expands δy as a Taylor series in $\delta\theta$

$$\delta y = L\delta\theta + \frac{1}{2}\delta\theta^\top M\delta\theta + \cdots, \tag{3.22}$$

where the elements of the Jacobian (aka sensitivity) vector $L \in \mathbb{R}^{1\times n_\theta}$ and Hessian matrix $M \in \mathbb{R}^{n_\theta \times n_\theta}$ are

$$L_i = \left.\frac{\partial y}{\partial \theta_i}\right|_{\theta=\hat{\theta}}, \qquad i = 1, \ldots, n_\theta, \tag{3.23}$$

$$M_{i,j} = \left.\frac{\partial^2 y}{\partial \theta_i \partial \theta_j}\right|_{\theta=\hat{\theta}}, \qquad i, j = 1, \ldots, n_\theta. \tag{3.24}$$

Worst-case Uncertainty Analysis

Parameter uncertainty is represented as sets in worst-case approaches. A fairly general uncertainty representation is given by the Hölder p-norm defined for a general vector $x \in \mathbb{R}^n$ as

$$\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}, \quad p \geq 1,$$
$$\|x\|_\infty = \max_i |x_i|.$$

This description can be used to represent hyper-ellipsoidal sets (such as the confidence region (3.20)) and general box sets as discussed in [179]. The set of possible parameter

values can then be represented as

$$\mathscr{P} = \left\{\theta : \theta = \hat{\theta} + \delta\theta,\ \|W_\theta \delta\theta\|_p \leq 1\right\}, \tag{3.25}$$

where W_θ is a specified positive definite weight matrix of dimension $(n_\theta \times n_\theta)$. The worst-case deviation in a particular process output is then defined by

$$\delta y_{w.c.} = \max_{\theta \in \mathscr{P}} |\delta y|. \tag{3.26}$$

First-order Expansion Consider the case where a first-order expansion, of the form (3.22), is used to approximate the output deviation, i.e., $\delta y \approx L\delta\theta$. Then, (3.26) reduces to

$$\delta y_{w.c.} = \max_{\|W_\theta \delta\theta\|_p \leq 1} |L\delta\theta|, \tag{3.27}$$

whose solution can be derived analytically for all $p \geq 1$ as done in [151]. The results are provided here for completeness:

- For $p = 1$

$$\delta y_{w.c.} = \|LW_\theta^{-1}\|_\infty.$$

- For finite $p > 1$

$$\delta y_{w.c.} = \left(\sum_{k=1}^{n_\theta} (|LW_\theta|_k)^{p/(p-1)}\right)^{(p-1)/p}.$$

- For $p = \infty$

$$\delta y_{w.c.} = \|LW_\theta^{-1}\|_1.$$

Second-order Expansion Consider the case where a second-order expansion, of the form (3.22), is used to approximate the output deviation, i.e., $\delta y \approx L\delta\theta +$

$(1/2)\delta\theta^\top M\delta\theta$. This problem can no longer be solved analytically, but can be rewritten in terms of the mixed structured singular value μ as described in [179]. Upper and lower bounds for this problem can be computed by iterative μ-computation and are typically tight enough for engineering purposes. Higher-order expansions can be handled in a similar manner.

Drawbacks The main drawback of worst-case approaches are that they do not capture the likelihood of each event occurring in the output space based on the likelihood of particular parameters being realized. If we have some knowledge of the probability distribution of the parameters, this can be used to understand how likely each scenario will occur (relative to the other possibilities). If only the worst-case possibility is considered in isolation, standard control methods can lead to quite conservative solutions that are not desired in practice. Less conservative solutions can be achieved by determining the PDF of the output values, which can also be done (in certain simplified cases) using power series methods as the distributional propagation tool as discussed next. The focus of this thesis is on stochastic propagation methods as they capture more information than their worst-case counterparts (i.e., worst-case approaches attempt to characterize the range/support of the PDF of the output values whereas probabilistic approaches determines the range/PDF support as well as the likelihood of each possibility).

Distributional Uncertainty Analysis

When the distribution of the parameters $f(\theta)$ has been estimated from data, a power series can be used to estimate the distribution of each output y of interest. When a first-order expansion $\delta y = L\delta\theta$ is used, the output PDF can be derived as

$$f(\delta y) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f\left(\delta\theta_1, \cdots, \delta y - \sum_{i=1}^{n_\theta - 1} L_i \delta\theta_i\right) d(\delta\theta_1) \cdots d(\delta\theta_{n_\theta}), \tag{3.28}$$

where the PDF inside the integral is that of the parameter perturbation $f(\delta\theta)$. This integral must often be solved numerically; however, it greatly simplifies when the

PDF of the parameters is assumed to be a multivariate normal distribution since the sum of normal random variables is also a normal random variable. This implies that y will also be normally distributed with mean $\hat{y}$ and variance $LV_\theta L^\top$ where V_θ is the variance of the parameter distribution (see [179] and the references therein for further details).

Analytic expressions for the output distribution cannot be obtained for higher order series expansions. Under this situation, y is related nonlinearly to θ such that the PDF must be computed numerically (oftentimes using some sort of Monte Carlo sampling method). For highly nonlinear processes, first-order expansions may not be accurate enough to capture the true system behavior. Additionally, (3.28) is not a simple expression that can utilized directly except in the case of, e.g., Gaussian uncertainties. As such, an alternative approach to power series based techniques (referred to as polynomial chaos methods) can be useful since they can be quite efficient and can handle a broader range of distributions. These methods are overviewed in the following sections.

3.4 Wiener–Hermite Polynomial Chaos

The term *polynomial chaos* was introduced by Norbet Wiener in his 1938 paper [266], in which he applied his generalized harmonic analysis to a mathematical formulation of statistical mechanics (where Brownian motion is the main motivation). The basic concept of Wiener-Hermite polynomial chaos is to represent a (finite variance) random variable by a series of Hermite polynomials in a countable sequence of independent Gaussian random variables. These exact representations can then be truncated to an expansion of finite order to be used as approximations.

Notation Let X and X_n be a real-valued random variables defined on a probability space $(\Omega, \mathcal{B}, P)$. For some set A, the notation $\mathbb{P}(X \in A)$ is used as a shorthand for $P(\{\omega \in \Omega : X(\omega) \in A\})$. Let the cumulative distribution functions (CDFs) of X and X_n be denoted by F and F_n, respectively. The mean or expectation of a random

variable X is denoted by $\mathbb{E}[X]$.

Definition 3.1 (L_p Space) *Let $F(x) = \mathbb{P}(X \leq x)$ be the CDF of random variable X. The random variable $X \in L_p$ for $p \geq 1$ if $\mathbb{E}[|X|^p] < \infty$. Note that for the special cases $p = 1$ and $p = 2$, we have $\mathbb{E}[|X|] < \infty$ and $\mathbb{E}[X^2] < \infty$ such that X has finite mean and finite variance, respectively.*

Modes of Convergence

Depending on how the sequence of random variables $\{X_n,\ n \geq 1\}$ is defined, this sequence may or may not converge to X is a number of different ways depending on the way in which their difference is measured. These are often referred to as modes of convergence, and are briefly reviewed below.

Definition 3.2 (Almost Sure Convergence) *The sequence is said to converge almost surely to X, denoted by $X_n \xrightarrow{a.s.} X$, if*

$$\lim_{n\to\infty} X_n(\omega) = X(\omega), \quad \forall \omega \in \Omega \setminus A, \tag{3.29}$$

for any event A with $P(A) = 0$.

Definition 3.3 (Convergence in Probability) *The sequence is said to converge in probability to X, denoted by $X_n \xrightarrow{i.p.} X$, if*

$$\lim_{n\to\infty} \mathbb{P}(|X - X_n| > \epsilon) = 0, \tag{3.30}$$

for any choice of $\epsilon > 0$.

Definition 3.4 (Convergence in Distribution) *The sequence is said to converge in distribution to X, denoted by $X_n \xrightarrow{d.} X$, if*

$$\lim_{n\to\infty} F_n(x) = F(x), \tag{3.31}$$

for all $x \in \mathbb{R}$ such that $F(x)$ is continuous.

Definition 3.5 (Convergence in L_p Sense) *For any $p \geq 1$, the sequence is said to converge to X in the L_p sense (or simply $\{X_n,\ n \geq 1\}$ exhibits L_p convergence), denoted by $X_n \xrightarrow{L_p} X$, if*

$$\lim_{n\to\infty} \mathbb{E}\left[|X - X_n|^p\right] = 0. \tag{3.32}$$

This is commonly referred to as mean-square convergence for the special case of $p = 2$.

Connection between Probabilistic Convergence Modes Almost sure convergence is the strongest mode of convergence and implies convergence in probability. Convergence in probability implies convergence in distribution, which can be shown using the definition of the CDF. Convergence in L_p for any $p \geq 1$ implies convergence in probability, which can be shown using the Markov inequality. Convergence in L_p also leads to the sequence converging in the L_r sense for any $1 \leq r < p$. The chain of implications between these different modes of convergence can then by compactly written using arrow notation [252]:

$$\begin{array}{ccccc} X_n \xrightarrow{L_p} X & \underset{p>r\geq 1}{\Longrightarrow} & X_n \xrightarrow{L_r} X & & \\ & & \Downarrow & & \\ X_n \xrightarrow{a.s.} X & \Longrightarrow & X_n \xrightarrow{i.p.} X & \Longrightarrow & X_n \xrightarrow{d.} X \end{array}$$

Representation of Wiener Polynomial Chaos Expansions

In his original paper, Wiener represented a general (finite variance) random variable $X(\omega)$, viewed as a function of the random event $\theta \in \mathbb{R}^n$, in the following way [272]

$$\begin{aligned} X(\theta) = & a_0 H_0 \\ & + \sum_{i_1=1}^{n} a_{i_1} H_1(\xi_{i_1}(\theta)) \\ & + \sum_{i_1=1}^{n} \sum_{i_2=1}^{i_1} a_{i_1,i_2} H_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta)) \\ & + \sum_{i_1=1}^{n} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1,i_2,i_3} H_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta)) \\ & + \cdots, \end{aligned} \tag{3.33}$$

where $H_j(\xi_{i_1}, \cdots, \xi_{i_j})$ represent the orthogonal Hermite polynomials of order j (for all $j \geq 0$ with $H_0 = 1$ being a constant) and $(\xi_{i_1}, \cdots, \xi_{i_j})$ denotes the multidimensional independent Gaussian random variables with zero mean and unit variance. The general expression for these polynomials is given by

$$H_j(\xi_{i_1}, \cdots, \xi_{i_j}) = e^{\frac{1}{2}(\xi_{i_1}^2 + \cdots + \xi_{i_j}^2)} (-1)^j \frac{\partial^j}{\partial \xi_{i_1} \cdots \xi_{i_j}} e^{-\frac{1}{2}(\xi_{i_1}^2 + \cdots + \xi_{i_j}^2)}. \tag{3.34}$$

For example, the one-dimensional hermite polynomials are

$$\begin{aligned} H_0 &= 1, \\ H_1 &= \xi, \\ H_2 &= \xi^2 - 1, \\ H_3 &= \xi^3 - \xi, \\ &\vdots \\ H_n &= \xi H_{n-1} - (n-1) H_{n-2}, \\ &\vdots \end{aligned}$$

For completeness, the two-dimensional version of the expansion (3.33) is expanded as

$$X(\theta) = a_0 H_0 + a_1 H_1(\xi_1) + a_2 H_1(\xi_2) + a_{1,1} H_2(\xi_1, \xi_1) \\ + a_{1,2} H_2(\xi_2, \xi_1) + a_{2,2} H_2(\xi_2, \xi_2) + \cdots . \tag{3.35}$$

For notational convenience, we can rewrite (3.33) as

$$X(\theta) = \sum_{j=0}^{\infty} \hat{a}_j \Psi_j(\boldsymbol{\xi}), \tag{3.36}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$. There is a one-to-one correspondence between the functions $H_j(\xi_{i_1}, \cdots, \xi_{i_j})$ and $\Psi_j(\boldsymbol{\xi})$. To illuminate this correspondence, the two-dimensional version of (3.36) is also expanded as

$$\begin{aligned} X(\theta) &= \hat{a}_0 \Psi_0 + \hat{a}_1 \Psi_1 + \hat{a}_2 \Psi_2 + \hat{a}_3 \Psi_3 + \hat{a}_4 \Psi_4 + \hat{a}_5 \Psi_5 + \cdots \\ &= \hat{a}_0 H_0 + \hat{a}_1 H_1(\xi_1) + \hat{a}_2 H_1(\xi_2) + \hat{a}_3 H_2(\xi_1, \xi_1) + \hat{a}_4 H_2(\xi_2, \xi_1) + \hat{a}_5 H_2(\xi_2, \xi_2) + \cdots \\ &= \hat{a}_0 + \hat{a}_1 \xi_1 + \hat{a}_2 \xi_2 + \hat{a}_3 (\xi_1^2 - 1) + \hat{a}_4 (\xi_1 \xi_2) + \hat{a}_5 (\xi_2^2 - 1) + \cdots \end{aligned} \tag{3.37}$$

The Wiener–Hermite polynomial chaos forms a complete orthogonal basis in the L_2-space, i.e.,

$$\langle \Psi_i, \Psi_j \rangle = \langle \Psi_i^2 \rangle \delta_{ij} \tag{3.38}$$

where δ_{ij} is the Kronecker delta (takes on the value zero when $i \neq j$ and the value one with $i = j$), and $\langle \cdot, \cdot \rangle$ is the inner product in the Hilbert space of Gaussian random variables, given by

$$\langle f(\boldsymbol{\xi}), g(\boldsymbol{\xi}) \rangle = \int f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) W(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{3.39}$$

where the weighting function is

$$W(\boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^2}} e^{-\frac{1}{2}\boldsymbol{\xi}^\top \boldsymbol{\xi}}. \tag{3.40}$$

The main difference the Wiener–Hermite from other possible complete sets of basis expansions is that the (Hermite) polynomials are orthogonal with respect to the zero mean and unit variance Gaussian weighting function [272].

The orthogonality condition can be used to find the corresponding coefficients of each term in the expansion (3.36)

$$\hat{a}_j = \frac{\langle \Psi_j, X \rangle}{\langle \Psi_j^2 \rangle}. \tag{3.41}$$

This projection requires the stochastic process $X(\theta)$ to share the same probability space with the Gaussian measure. When X is directly a function of independent Gaussian random variables (for example, $X = e^{\xi_1} \sin(\xi_2)$), these can be used as the basic random variable and no transformation is required. Otherwise, we must transform X and the basic random variables $\boldsymbol{\xi}$ into the same probability space.

Wiener–Hermite Polynomial Chaos for non-Gaussian Uncertainties One possible method, based on an idea proposed in [219], is to transform both to a uniform distribution. Let $U^n(0,1)$ denote a n-dimensional multivariate uniform distribution on the hypercube $[0,1]^n$ (with each element be uniformly and independently distributed on $[0,1]$). The notation $\sim$ implies "distributed as." Denote the CDF of random vector $X = (X_1, \cdots, X_n)$ as $F(x_1, \cdots, x_n)$. The random vector $\boldsymbol{\xi} \sim W$ where W is the multivariate Gaussian PDF of zero mean and unit variance defined in (3.40).

Define the transformation $z = (u_1, \cdots, u_n) = Tx$

$$\begin{aligned} z_1 &= F_1(x_1), \\ z_2 &= F_2(x_2|x_1), \\ &\vdots \\ z_n &= F_n(x_n|x_{n-1}, \cdots, x_1). \end{aligned}$$

As shown in [219], the random vector $Z = TX \sim U^n(0,1)$ (i.e., is composed of independent uniform random variables). Using a similarly defined transformation, one can write $Z = N\boldsymbol{\xi}$. Combining these results, we can write $X = T^{-1}N\boldsymbol{\xi}$ where T^{-1} is the inverse of T. Now, X and $\boldsymbol{\xi}$ have been mapped to the same probability space such that the integration in (3.41) can be performed by substituting $X = T^{-1}N\boldsymbol{\xi}$. See [12, Example 2.3.2] for an example of this method being applied to a random variable whose PDF is described by a Gaussian mixture model.

Convergence via the Cameron–Martin Theorem

For practical reasons, the Wiener–Hermite polynomial chaos expansion must be truncated to a finite number of terms. For notational simplicity, only a scalar random variable is considered here. The n-term polynomial chaos expansion approximation of X can then be stated as

$$X_n(\xi) = \sum_{j=0}^{n-1} a_j H_j(\xi). \tag{3.42}$$

Cameron and Martin [39] studied the convergence behavior of this sequence to Xas $n \to \infty$. This can be encapsulated in the following theorem, which states that the Hermite polynomial chaos representation of any functional $X : \mathbb{R} \to \mathbb{R}$ of a Gaussian random variable $\xi : \Omega \to \mathbb{R}$ converges in the mean-square sense.

Theorem 3.1 (Cameron–Martin Theorem [39]) *The Wiener–Hermite polynomial chaos expansion* (3.42) *of any second-order functional* $X(\xi)$ *converges in the*

mean-square sense to X. *This implies that if* $X \in L_2$, *then* $X_n \xrightarrow{L_2} X$.

The proof of this result is not trivial and requires a background in probability and measure theory. Interested readers are referred to the original paper [39] for details. This result can also be extended to multidimensional polynomial chaos expansions as shown in [65]. As discussed previously, the mean-square convergence of Wiener–Hermite polynomial chaos expansions implies convergence in the mean (L_1), convergence is probability, and convergence in distribution.

3.5 Generalized Polynomial Chaos

The Wiener–Hermite chaos expansion has been effectively used to solve stochastic differential equations with Gaussian inputs based on the Cameron–Martin theorem. However, for more general (non-Gaussian) random inputs, the convergence rate can be quite slow [272]. To handle a more general set of random inputs, *Askey-chaos* has been introduced as a generalization of the Wiener's original Hermite-chaos [272]. The expansion basis is formed by the complete set of polynomials from the Askey-scheme. In Askey–chaos, the underlying random variables are not limited to Gaussian random variables. Instead, the orthogonal polynomial basis of the Askey–chaos is selected based on the PDF of the input random variables.

Similarly to that shown in Section 3.4, a general second-order random process $X(\theta)$ can be represented as

$$\begin{aligned} X(\theta) = c_0 I_0 & \\ &+ \sum_{i_1=1}^{n} c_{i_1} I_1(\zeta_{i_1}(\theta)) \\ &+ \sum_{i_1=1}^{n} \sum_{i_2=1}^{i_1} c_{i_1,i_2} I_2(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta)) \\ &+ \sum_{i_1=1}^{n} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} c_{i_1,i_2,i_3} I_3(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta), \zeta_{i_3}(\theta)) \\ &+ \cdots, \end{aligned} \tag{3.43}$$

where $I_j(\zeta_{i_1}, \cdots, \zeta_{i_j})$ represent the Askey–chaos polynomials of order j in terms of the multidimensional random variables $(\zeta_{i_1}, \cdots, \zeta_{i_j})$. Again, for notational convenience, (3.43) is rewritten as

$$X(\theta) = \sum_{j=0}^{\infty} \hat{c}_j \Phi_j(\boldsymbol{\zeta}), \tag{3.44}$$

where $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$. Again, there is a one-to-one correspondence between the functions $I_j(\xi_{i_1}, \cdots, \xi_{i_j})$ and $\Phi_j(\boldsymbol{\zeta})$ as discussed in Section 3.4. The key difference between (3.33) and (3.43) is that H_n are Hermite polynomials while I_n are not restricted to be Hermite polynomials, but could be any type of orthogonal polynomials from the Askey scheme. The orthogonality condition of the Askey–chaos polynomial expansion is written as

$$\langle \Phi_i, \Phi_j \rangle = \langle \Phi_i^2 \rangle \delta_{ij}. \tag{3.45}$$

Here, the inner product $\langle \cdot, \cdot \rangle$ is defined in the Hilbert space of random variables $\boldsymbol{\zeta}$

$$\langle f(\boldsymbol{\zeta}), g(\boldsymbol{\zeta}) \rangle = \int f(\boldsymbol{\zeta}) g(\boldsymbol{\zeta}) W(\boldsymbol{\zeta}) d\boldsymbol{\zeta}, \tag{3.46}$$

where $W(\boldsymbol{\zeta})$ is the weighting function corresponding to the Askey-chaos basis $\{\Phi_i\}$. The key idea is to select a polynomial basis that is orthogonal with respect to a weight $W(\boldsymbol{\zeta})$ that equals the PDF of certain types of random variables. Table 3.1 shows the Wiener–Askey polynomial basis for particular distributions of $\boldsymbol{\zeta}$.

The basic steps of how to compute a generalized polynomial chaos expansion (PCE) and a systematic method for determining the number of terms to keep in the expansion is described in Figure 3-1.

The subsequent chapters of this thesis introduce the model predictive control framework and attempt to utilize generalized PCE within model predictive control to handle probabilistic parameter uncertainty. This is an important problem because, as shown later, parameter uncertainty can destabilize model-based "optimal" controllers. This methodology allows for operators to systematically tradeoff between system

Table 3.1: Correspondence of Wiener–Askey polynomial chaos basis to the distribution of the random input

Random Input	Wiener–Askey polynomial chaos	Support
Gaussian	Hermite–chaos	$(-\infty, \infty)$
Beta	Jacobi–chaos	(a, b)
Gamma	Laguerre–chaos	$[0, \infty)$
Uniform	Legendre–chaos	$[a, b]$

performance and system robustness with minimal controller tuning parameters.

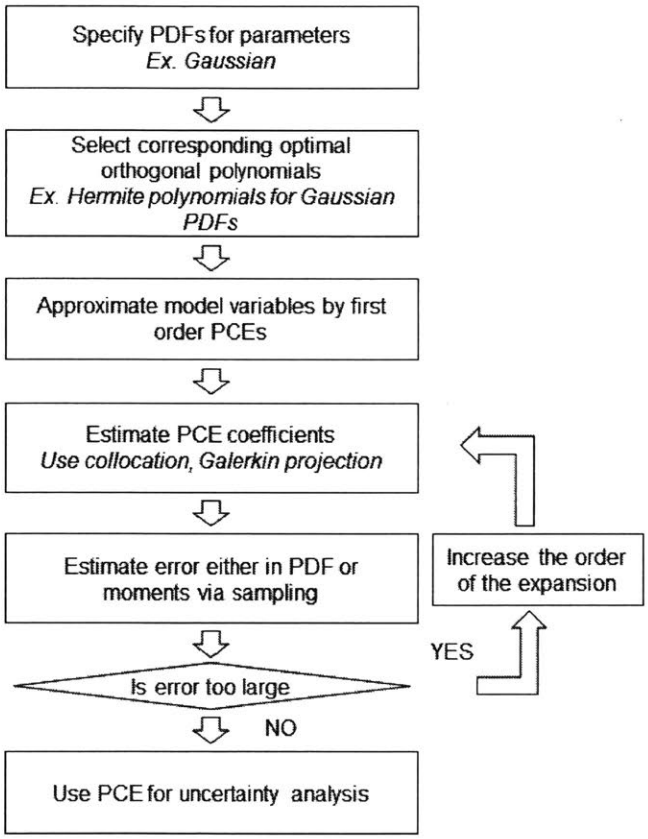


Figure 3-1: Basic description of the polynomial chaos framework and how to systematically select the order to achieve a desired level of accuracy.

Part III

Constrained Predictive Control of Large-scale Systems

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Optimization Methods for Fast Model Predictive Control

4.1 Introduction

In classical model predictive control (MPC), the control action that minimizes some performance objective subject to constraints is computed as the solution to an online optimization problem at each time step. This optimization problem will be a convex quadratic program (QP) whenever the model is linear, the constraints are polyhedral, and the cost is quadratic. Solving this QP online using general purpose methods can be quite slow and has traditionally limited MPC to applications with relatively slow dynamics. However, a significant reduction in the computational cost can be achieved by exploiting the structure and sparsity of the QP [268, 215, 260].

In this chapter, we look to review a collection of methods that can be used to greatly reduce the online cost of the computation of the control action in MPC. We explore two main ideas: 1) the choice of optimization method used to solve the QP and 2) the formulation of the optimization problem to provide "optimal" structure. Interior point, active set, and fast gradient projection methods have all been successfully applied to MPC problems. Significant decreases in computational cost have been demonstrated in these methods when taking into account the formulation and structure of the problem when compared to a naive approach. We also briefly discuss

currently available toolboxes that implement these methods for MPC problems.

Not only must we choose the particular optimization algorithm to solve the problem, we must also choose a formulation of the optimization (i.e., which decision variables to include). Traditionally, the MPC optimization problem is formulated either in a form where the states are also kept as decision variables or a form where only the control actions are decision variables. The latter is often referred to as *condensing.* If the states are kept, then the system of equations solved at each iteration is sparse, whereas if the condensed formulation is used then the problem becomes dense. For example, using an interior point method on a problem with n_x states, n_u inputs, and horizon of length N will require $O((n_x + n_u)^3 N)$ operations per iteration if the sparsity is exploited in the sparse formulation [215] while it requires $O(n_u^3 N^3)$ operations per iteration for the condensed version of the problem [260]. Some recent work suggests that it is possible to consider not just these two distinct choices of formulations, but instead consider a family of formulations each with a different level of sparsity and number of variables to obtain even better performance [6]. As such, we would expect the fastest formulation to depend strongly on the problem data and size and, in particular, the relative size of n_x, n_u, and N.

Recently, it has been shown that the MPC QP can be solved explicitly offline as a function of the initial state [15], such that the control action can be determined online using a simple lookup table. Although this enables MPC to be applied to very fast sampled systems (sampling times < 1 μs) [116], the table size can grow exponentially with the horizon, state, and input dimensions. As a result, so-called explicit MPC is only tractable for small problems. We review the details of explicit MPC as well as discuss recent approaches that attempt to make the offline program more tractable.

The quadratic dynamic matrix control (QDMC) algorithm is a particularly prevalent MPC method in industry that solves a QP online [76, 211]. This has been shown to be a special case of the more general state-space MPC in [142]. We speculate on why this method is heavily used in the chemical process industry as well as highlight the circumstances in which it will be most effective. We also demonstrate its performance on an example system to illustrate that the choice of optimization algorithm

and formulation has a great impact on the QP solution speed.

Organization The standard nominal MPC problem is defined. An overview of recent theoretical advances for MPC is then overviewed. Special cases of the MPC problem that provide additional structure that can exploited during solution of the QP are also discussed. Then, the interior point, active set, and fast gradient projection methods for solving convex QPs are discussed. Different formulations of the QP in the MPC setting are then explored, with a focus on how the structure of the problem can be exploited in each of these methods to greatly reduce the cost of solving the MPC problem. It is shown that QDMC is a special case of state-space MPC with a particularly advantageous structure amenable to fast compuations (illustrated on a distributed parameter system). Lastly, some conculding remarks are made.

Notation Let $\mathbb{I}_{\geq 0} = \{0, 1, \ldots\}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}^{n\times m}$, $\mathbb{S}^n_+$, and $\mathbb{S}^n_{++}$ denote the sets of non-negative integers, non-negative real numbers, real matrices with n rows and m columns, and symmetric positive semidefinite and positive definite n by n matrices, respectively, I_N is the identity matrix of size N by N, $\mathbf{1}_N$ is a column vector of ones of size N, diag$(\cdot)$ puts the vector argument along the diagonal of a matrix, blkdiag$(\cdot)$ is a block diagonal matrix, and $\otimes$ is the Kronecker product. A polyhedron is the intersection of a finite number of half-spaces, a polytope is a closed and bounded polyhedron, and $U^N = U \times \cdots \times U$ for any set U. Lastly, let $O(\cdot)$ denote order of magnitude.

4.2 Problem Formulation

System Dynamics

Throughout this chapter, we consider the regulation of the discrete-time linear system

$$x(k+1) = Ax(k) + Bu(k), \tag{4.1}$$

$$y(k) = Cx(k), \tag{4.2}$$

to the origin while fulfilling general mixed constraints

$$(x(k), u(k)) \in \mathbb{Z}, \tag{4.3}$$

for all $k \in \mathbb{I}_{\geq 0}$, where $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the manipulated input, and $y(k) \in \mathbb{R}^{n_y}$ is the output. Here, we focus on the nominal problem, which neglects model uncertainty and disturbances, to highlight fast computational procedures for optimization problems prevalent in model-based control. The mixed state and input constraint set is assumed to be a collection of n_c linear inequalities

$$\mathbb{Z} = \{(x, u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mid F_x x + F_u u \leq f\}, \tag{4.4}$$

where $F_x \in \mathbb{R}^{n_c \times n_x}$, $F_u \in \mathbb{R}^{n_c \times n_u}$, and $f \in \mathbb{R}^{n_c}$. To simplify notation in the sequel, we will commonly drop the time index k and express (4.1)–(4.2) as

$$\begin{aligned} x^+ &= Ax + Bu, \\ y &= Cx, \end{aligned}$$

where the superscript $^+$ denotes the successor state at the next sample time.

Constrained Linear-Quadratic Regulation

Constrained linear-quadratic (LQ) control can be considered to be the core problem in MPC formulations and is defined as follows [161]. Assume full state information x

is available at the current time step. Then, solve the optimization problem

$$\min_{\mathbf{u},\mathbf{x}} V_N(x,\mathbf{u}) = \sum_{j=0}^{N-1} l(x_j,u_j) + V_f(x_N), \tag{4.5a}$$

$$\text{s.t. } x_{j+1} = Ax_j + Bu_j, \quad j = 0,\ldots,N-1, \tag{4.5b}$$

$$x_0 = x, \tag{4.5c}$$

$$(x_j,u_j) \in \mathbb{Z}, \quad j = 0,1,\ldots,N-1, \tag{4.5d}$$

$$x_N \in \mathbb{X}_f, \tag{4.5e}$$

where x_j denotes the state predicted j steps into the future from initial state x under control sequence $u_0,\ldots,u_{j-1}$ by (4.1), N is the control horizon (assumed equal to the prediction horizon for simplicity), and $\mathbb{X}_f$ is the terminal set assumed to be a polytope of the form

$$\mathbb{X}_f = \{x \mid F_N x \leq f_N\}, \tag{4.6}$$

with $F_N \in \mathbb{R}^{n_N \times n_x}$ and $f_N \in \mathbb{R}^{n_N}$. The stage cost is assumed to be of the form

$$l(x,u) = \frac{1}{2}\begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + q^\top x + r^\top u, \tag{4.7}$$

and the terminal cost is assumed to be of the form

$$V_f(x) = \frac{1}{2}x^\top Q_N x + q_N^\top x, \tag{4.8}$$

where Q, R, M, and Q_N satisfy

$$R \in \mathbb{S}_{++}^{n_u}, \quad \begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix} \in \mathbb{S}_{+}^{n_x+n_u}, \quad Q_N \in \mathbb{S}_{+}^{n_x},$$

such that the optimization problem (4.5) is a convex quadratic program (QP) and has a unique solution. Denote the input sequence associated with (4.5) as

$$\mathbf{u} = (u_0, \ldots, u_{N-1}), \tag{4.9}$$

and the corresponding state trajectory as

$$\mathbf{x}(x, \mathbf{u}) = (x_0, \ldots, x_N(x, \mathbf{u})). \tag{4.10}$$

The optimization problem (4.5) implicitly defines a set of feasible control sequences

$$\mathscr{U}_N(x) = \{\mathbf{u} \mid (\mathbf{x}(x, \mathbf{u}), \mathbf{u}) \in \mathbb{Z}^N \times \mathbb{X}_f\}, \tag{4.11}$$

and set of feasible initial states

$$\mathscr{X}_N = \{x \mid \mathscr{U}_N(x) \neq \emptyset\}, \tag{4.12}$$

commonly referred to as the domain or region of attraction in the MPC literature. By definition, problem (4.5) is infeasible for any $x \notin \mathscr{X}_N$. For any $x \in \mathscr{X}_N$, the solution to (4.5) yields an optimal control sequence

$$\mathbf{u}^\star(x) = \arg\min_{\mathbf{u}} \{V_N(x, \mathbf{u}) \mid \mathbf{u} \in \mathscr{U}_N(x)\}, \tag{4.13}$$

and the implicit MPC control law

$$\kappa_N(x) = [I \ 0 \ \ldots \ 0]\mathbf{u}^\star(x), \tag{4.14}$$

is computed based on what is known as a receding horizon implementation, in which, the first element of $\mathbf{u}^\star(x)$ is supplied to the nominal system (4.1). The closed-loop system then evolves as

$$x^+ = Ax + B\kappa_N(x), \tag{4.15}$$

starting from some initial condition $x = x(0)$.

Many methods reviewed in this work for solving (4.5) can be extended to problems with costs and constraints on the input rates $u_{j+1} - u_j$, time-varying costs, constraints, and/or models, reference tracking, affine system descriptions, input parametrizations $u_k = Kx_k + v_k$, and different control and prediction horizons. Additionally, sequences of LQ problems similar to (4.5) often arise in control of nonlinear systems [267, 215, 10]. For example, when applying the sequential quadratic programming method to nonlinear MPC problems, we obtain a search direction at each iteration by solving a problem like (4.5) [11].

Problem (4.5) is a multi-stage optimization parametrized by initial condition x and can theoretically be solved to optimality using dynamic programming (DP) methods [21, 216]. The DP solution yields an optimal sequence of control laws, which is a list of N functions that will often vary with time over the horizon. MPC, on the other hand, implements the time-invariant control law κ_N at every time step. Thus the evolution of (4.15) will differ from the closed-loop system evolution generated by the optimal DP control laws. As such, MPC is not truly optimal for (4.5) on a finite horizon. However, DP is computationally intractable for constrained and/or nonlinear control problems unless the problem size is very small while MPC is a practical way to evaluate the maps $V_N^\star(x)$ and $\kappa_N(x)$ at a point x. A detailed discussion of these issues, including worked out example problems, is provided in [216].

4.3 Cases of Model Predictive Control

In this section, we describe special cases of (4.5) that commonly arise in MPC problems. We also describe how selection of the terminal cost V_f and constraint set $\mathbb{X}_f$ can be used to guarantee feasibility of (4.5), ensure stability of the resulting closed-loop system, and exactly compute the infinite-horizon cost.

Feasibility and Stability

When $N < \infty$, there is no explicit guarantee that problem (4.5) is always feasible as the system might enter regions of the state-space where no solution exists; however, setting $N = \infty$ yields an infinite number of constraints that are not possible to handle directly in the optimization [15]. Recursive feasibility of (4.5) can be guaranteed whenever the following assumption is satisfied for some terminal controller Kx

$$(A + BK)\mathbb{X}_f \subseteq \mathbb{X}_f,\ \mathbb{X}_f \times K\mathbb{X}_f \subseteq \mathbb{Z}, \tag{4.16}$$

such that $\mathbb{X}_f$ is a *positively invariant (PI) set* for the system (4.1) and constraint set $\mathbb{Z}$ under control law Kx, i.e., ensures once the state enters $\mathbb{X}_f$, there exists at least one controller that keeps the state inside of $\mathbb{X}_f$ for all time while satisfying constraints [161].

A number of methods have been developed to compute an $\mathbb{X}_f$ that satisfies (4.16). The maximal PI (MPI) set is a polytope that can be finitely determined for stable $A+BK$ [82], but is expensive to compute for higher dimensions. Most approaches derive $\mathbb{X}_f$ as a level set of a quadratic Lyapunov function. This ellipsoidal approximation to the MPI can be computed from a simple linear program (LP) that is solvable for high dimensions. Although this ellipsoidal approximation can be conservative, it is often the only choice when the state dimension exceeds 6–7 [277]. Here, we focus on the case in which $\mathbb{X}_f$ is a polytope (4.6) for simplicity such that problem (4.5) is a QP. When $\mathbb{X}_f$ is an ellipsoid, (4.5) is a quadratically constrained QP (QCQP). Many of the methods reviewed below for quickly and efficiently solving the MPC problem will directly extend to this case as well since the resulting QCQP is convex and has a limited number of quadratic constraints [277].

If (4.16) holds, $\kappa_N(x)$ asymptotically stabilizes the closed-loop system (4.15) whenever assumption

$$V_f((A + BK)x) - V_f(x) \leq -l(x, Kx),\ \forall x \in \mathbb{X}_f, \tag{4.17}$$

is satisfied, since this condition guarantees that the optimal value function $V_N^\star(x) = V_N(x, \mathbf{u}^\star(x))$ is a Lyapunov function for (4.15) with region of attraction $\mathscr{X}_N$. Interested readers are referred to [161] for explicit proofs of these statements.

Infinite-Horizon Cost

The constrained infinite-horizon LQ regulator minimizes the cost $\sum_{j=0}^{\infty} l(x_j, u_j)$ subject to constraints (4.5b)–(4.5e) with $q = r = 0$ in (4.7). This problem is not solvable directly since it is posed as an infinite-dimensional optimization [215]. We can reduce this to a finite-horizon problem of the form (4.5) by using a linear control law to determine u_k after a certain time horizon, that is,

$$u_k = Kx_k, \quad \text{for all } k \geq N. \tag{4.18}$$

With these added constraints, the states x_k, $k > N$ and inputs u_k, $k \geq N$ are completely determined by the state at the end of the time horizon x_N. Then, the tail of the infinite-horizon cost is exactly written as

$$\frac{1}{2}\sum_{j=N}^{\infty}(x_j^\top Q x_j + x_j^\top K^\top R K x_j + 2x_j^\top M K x_j), \tag{4.19}$$

where this infinite summation can be replaced with a single quadratic term $(1/2)x_N^\top P x_N$ whenever $A + BK$ is stable. The gain K is normally found from the solution to the classical unconstrained LQ regulator, which minimizes (4.19) in the absence of constraints. In this case, P is found as the solution to the following discrete-time algebraic Riccati equation

$$P = A^\top P A - (A^\top P B + M)(R + B^\top P B)^{-1}(B^\top P A + M^\top) + Q, \tag{4.20}$$

and the feedback gain K is computed from P by

$$K = -(R + B^\top P B)^{-1}(B^\top P A + M^\top), \tag{4.21}$$

which corresponds to the optimal unconstrained linear control law (see, e.g., [215]). The terminal cost term in problem (4.5) will then exactly represent the infinite tail of the cost whenever $Q_N = P$ and $q_N = 0$.

Problem-Dependent Structure

As mentioned previously, (4.5) can be straightforwardly extended to include a number of of common variants of MPC and it represents a very general form of MPC for linear systems. However, as one might expect, there are a number of ways to restrict the objective and constraints such that (4.5) has more structure. One example often encountered is when the objective and constraints are separable in the state and inputs. This means $M = 0$ and the constraints (4.5d) can be written separately as

$$x_j \in \mathbb{X}, \quad u_j \in \mathbb{U}.$$

A further specialization is when Q and R are diagonal matrices. Also, the state and input constraints can be simple box constraints

$$x_{\min} \leq x_j \leq x_{\max}, \quad u_{\min} \leq u_j \leq u_{\max}$$

that consist only of lower and upper bounds. Only output constraints $\mathbb{Y}$ can be considered in place of state constraints $\mathbb{X}$ such that $y_j \in \mathbb{Y}$ (or $y_{\min} \leq y_j \leq y_{\max}$ in the case of box constraints) where $y_j = Cx_j$ for all $j \in \mathbb{I}_{\geq 0}$.

Obviously, the exact formulation of problem (4.5) depends on the specific problem one is trying to solve. As discussed in the sequel, the structure and sparsity of the optimization problem can greatly influence the speed at which one can solve (4.5). We focus on the more general case in this work, as it is not possible to enumerate all possibilities, but point out that any and all structuring/simplifications should be taken into account when solving problem (4.5) to maximize the reduction in complexity when solving the optimization.

Output feedback

The MPC problem (4.5) assumes full (perfect) state information at every time step. In most applications, the full state cannot be measured exactly and must be reconstructed from (typically noisy) measurements instead. This problem is known as state estimation. We let the output $y = Cx$ denote the measured variables for notational simplicity. The outputs are typically a small subset of the states, such that $n_y < n_x$.

A number of state estimation methods have been developed and explored in the literature. Some of the most common ones are the Kalman filter and moving horizon estimation (MHE). For linear systems subject to normally distributed process and measurement noise (with known covariances), the Kalman filter is the optimal estimator and has a relatively simple analytic solution [120]. When the model is nonlinear or constraints on the estimates need to be considered, a closed-form solution usually cannot be obtained. Linear MHE, on the other hand, solves a finite horizon least-squares optimization to obtain state estimates [214].

In the sequel, we denote the estimate of the state as $\hat{x}$. The majority of output feedback MPC methods for linear systems look to apply a certainty equivalence argument, wherein the state estimator is independently designed from the controller, such that x can merely be replaced with its estimate $\hat{x}$ in problem (4.5). In the control literature, this is referred to as the separation principle and will retain optimality under certain assumptions, for example, in LQ Gaussian control [5].

Another common state estimator for deterministic systems is the Luenberger observer

$$\hat{x}^+ = A\hat{x} + Bu + L(y - \hat{y}), \tag{4.22}$$

$$\hat{y} = C\hat{x}, \tag{4.23}$$

where $\hat{x}$ and $\hat{y}$ are the estimator state and output, respectively, and L is some observer gain. If L can vary with time, (4.22)–(4.23) can be used to represent the more general case of a Kalman filter. The observer error $e = x - \hat{x}$ for the nominal system then

satisfies

$$e^+ = (A - LC)e, \tag{4.24}$$

and will converge to zero when $A-LC$ is stable. Stability of the combined Luenberger observer/MPC controller system $x^+ = Ax + B\kappa_N(\hat{x})$, for the case in which only the initial condition is unknown, was explored in [162] using results for perturbed systems. When e is exponentially stable and $\kappa_N(\cdot)$ is Lipschitz continuous, asymptotic stability can be guaranteed for some initial states $x(0) \in \mathscr{C} \subset \mathscr{X}_N$ and initial observer error $e(0) \in \mathscr{E}$ where $\mathscr{E}$ is a small enough bounded set.

Reference Tracking

Setpoint tracking, in which one wants to design a controller such that the output tracks a constant reference $y_s \in \mathbb{R}^{n_y}$, is a common variant of MPC. We can easily extend problem (4.5) to setpoint tracking by computing the state and input target condition from [216]

$$\begin{bmatrix} I_{n_x} - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s \end{bmatrix}, \tag{4.25}$$

which must have $(x_s, u_s) \in \mathbb{Z}$ for y_s to be reachable. If this set of linear equations has a solution, we can convert the tracking problem to a regulation problem by shifting the coordinates of (4.1) in terms of deviation variables $\tilde{x} = x - x_s$ and $\tilde{u} = u - u_s$. Interested readers are referred to [178] for further details.

This method, however, assumes that the model is accurate and the system is not affected by any unknown disturbances (which is rarely the case in practice). It is often a desired property of the MPC controller to force the outputs to their setpoints without offset, referred to as zero-offset MPC. We can obtain integral action in MPC (making it zero-offset) by redefining the system model (e.g., rewriting (4.1) in terms of the change in the input $u_j - u_{j-1}$ or in "velocity" form [177]) or by modeling and estimating the disturbance [216]. The latter is a standard method and works as

follows. Augment the state-space system (4.1) with a step disturbance $d \in \mathbb{R}^{n_d}$

$$\begin{bmatrix} x \\ d \end{bmatrix}^+ = \begin{bmatrix} A & B_d \\ 0 & I_{n_d} \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u, \tag{4.26}$$

$$y = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix}, \tag{4.27}$$

where matrices B_d and C_d describe how the disturbance effects the state and output, respectively, and are selected by the user. Next, design an observer of the form (4.22)–(4.23) so that the state and disturbance can be estimated at every time step. Then, compute targets x_s and u_s at every time step using a modified version of (4.25) to account for the estimated disturbance, and solve a MPC regulation problem (4.5) replaced with deviation variables. Under certain assumptions, this strategy will provide a zero-offset MPC controller. Please refer to [216, §1.5] for further details.

4.4 Optimization Methods for Quadratic Programming

Problem (4.5) is a convex QP under the aforementioned assumptions of a linear model, polyhedron constraints, and a quadratic cost. Many methods and solvers have been proposed for solving the QP that arises in MPC for linear systems. In this section, we consider three approaches: interior point methods, active set methods, and fast gradient projection methods. We give an overview of these algorithms for a general convex QP of the form

$$\min_z \ \tfrac{1}{2} z^\top H z + c^\top z \ \text{ s.t. } \ Dz = d, \ Gz \leq g, \tag{4.28}$$

where $z \in \mathbb{R}^n$ are the decision variables, $H \in \mathbb{S}^n_+$, $c \in \mathbb{R}^n$, $D \in \mathbb{R}^{p \times n}$, $d \in \mathbb{R}^p$, $G \in \mathbb{R}^{m \times n}$, and $g \in \mathbb{R}^m$. Obviously, the LQ problem (4.5) is a special case of (4.28). Thus, the complexity of solving (4.5) will strongly depend on the formulation of the

optimization and the problem size, which are discussed in detail in Section 4.5.

Interior Point Methods

Interior point (aka barrier) methods come in a variety of flavors. The class of primal-dual path-following interior point methods (IPMs) is widely considered to be the most successful, wherein, the predictor-corrector scheme by Mehrotra is the basis for most implementations. This approach has been applied to MPC in, e.g., [215, 60] and has proven to be efficient in practice. An overview of the details are provided in the following.

The Karush-Kuhn-Tucker (KKT) conditions for (4.28) are

$$\mathscr{F}(z,\nu,\lambda,s) = \begin{bmatrix} Hz + c + D^\top \nu + G^\top \lambda \\ Dz - d \\ Gz - g + s \\ S\Lambda \mathbf{1}_m \end{bmatrix} = 0, \tag{4.29a}$$

$$(\lambda, s) \geq 0, \tag{4.29b}$$

where $s \in \mathbb{R}^m$ are slack variables, $\nu \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}^m$ are Lagrange multipliers, $S = \text{diag}(s_1, s_2, \ldots, s_m)$, and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$.

Primal-dual IPMs generate iterates $(z^i, \nu^i, \lambda^i, s^i)$, $i > 0$ for which $(\lambda^i, s^i) > 0$ and approach satisfaction of (4.29) as $i \to \infty$. Like most iterative optimization algorithms, they are composed of two basic ingredients: a procedure for computing a step and a measure of optimality of each point. The step is composed of a search direction $(\Delta z^i, \Delta \nu^i, \Delta \lambda^i, \Delta s^i)$ and a step-length α_i that determines the distance to be taken along the search direction. The search directions are Newton-like directions for the nonlinear set of equations in (4.29a) determined from the following system of linear

equations

$$\begin{bmatrix} H & D^\top & G^\top & \\ D & & & \\ G & & & I_m \\ & & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \nu \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} r_H \\ r_D \\ r_G \\ r_S \end{bmatrix} \tag{4.30}$$

where we have dropped the superscript to denote the current iterate and (r_H, r_D, r_G, r_S) are the residuals. When the residual vector is set equal to $\mathscr{F}(z, \nu, \lambda, s)$, this is a pure Newton step. The duality gap is defined as

$$\mu = \lambda^\top s / m \tag{4.31}$$

and is commonly used as the measure of optimality of the current iterate so that the algorithm will terminate when μ is below some tolerance.

Primal-dual IPMs differ mainly by their choice in right hand sides (r_H, r_D, r_G, r_S) [269]. The Mehrotra predictor-corrector algorithm is one of the most widely used as it has been very successful in practice [163]. Here, we will elaborate on the most expensive step of these methods, which is the computation of the search direction (4.30).

Block elimination can always be applied to (4.30) to derive a reduced system of equations with a convenient structure. First, we can eliminate Δs

$$\begin{bmatrix} H & D^\top & G^\top \\ D & & \\ G & & -\Lambda^{-1} S \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \nu \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_H \\ r_D \\ r_G - \Lambda^{-1} r_S \end{bmatrix} \tag{4.32}$$

where $\Delta s = -\Lambda^{-1} r_S - \Lambda^{-1} S \Delta \lambda$. Since $\Lambda^{-1} S$ is a diagonal matrix with all positive elements, we can also eliminate $\Delta \lambda$ to obtain the so-called augmented system

$$\begin{bmatrix} \Phi & D^\top \\ D & \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} r_\Phi \\ r_D \end{bmatrix} \tag{4.33}$$

where $\Delta\lambda$ is recovered from the relationship $\Delta\lambda = S^{-1}\Lambda(G\Delta z + r_G) - S^{-1}r_S$ while the terms in (4.33) are given by $\Phi = H + G^\top S^{-1}\Lambda G$ and $r_\Phi = r_H + G^\top S^{-1}(\Lambda r_G - r_S)$.

A very similar set of equations to (4.33) is obtained when using an infeasible start primal barrier IPM to solve the QP (4.28). The main difference is that Φ and r_Φ are replaced with new values Φ_κ and r_κ, respectively, which are functions of a barrier parameter κ and the inequality constraint violation. See [261] for more details including the definitions of Φ_κ and r_κ.

The KKT system, either (4.30), (4.32), or (4.33), is typically solved with some modified form of Gaussian elimination that scales cubically with respect to the number of unknowns. Since the coefficient matrix is a function of the current iterate in IPMs, the system must be solved from scratch at every iteration. Therefore, it is cheapest to solve (4.33) instead of (4.30) or (4.32) since it has the fewest number of unknowns.

Active Set Methods

Active set methods (ASMs) are the main alternative to IPMs for handling inequality constraints in convex optimization problems. ASMs solve (4.28) by identifying which constraints are *active* (that is, hold with equality) at its solution. The collection of active inequalities at the solution is called the *active set.* These methods start with an initial guess of the active set, termed the *working set.* At every iteration, the working set is modified by either adding a constraint to it or deleting a constraint from it. In this way, the guess of the active set is iteratively refined until the exact active set is determined [268].

ASMs are classified as either primal or dual [129]. All primal ASMs start with a feasible initial iterate and ensure that all subsequent iterates remain feasible. This is useful for MPC problems as the algorithm can be stopped at any moment while still providing an input sequence that satisfies constraints. However, a feasible point must be found before the procedure can actually start, which can be expensive to find. Dual ASMs systematically identify constraints that are not active at the solution. As such, they only satisfy constraints at the final iterate, but do not require a feasible initial iterate. Furthermore, the dual QP only involves box inequality constraints

that can simplify computations. See [129, §5.1] and the references therein for more on primal versus dual ASMs.

The details of ASMs are more complicated than those in IPMs. We focus on describing one iteration of the primal approach (see [74] for a more complete description) to give some insight into the algorithm and make a high-level comparison to IPMs. Let $\bar{G}$ denote the subset of the rows of G that make up the working set of constraints. A step is computed from the current point z by minimizing the objective in (4.28) while maintaining activity of the working set constraints and ensuring $Dz = d$ holds

$$\begin{aligned} \min_{\Delta z} \ & \tfrac{1}{2}(z+\Delta z)^\top H(z+\Delta z) + c^\top(z+\Delta z), \\ \text{s.t.} \ & D\Delta z = 0, \quad \bar{G}\Delta z = 0. \end{aligned}$$

This equality-constrained QP is equivalent to [268]

$$\min_{\Delta z} \ \tfrac{1}{2}\Delta z^\top H \Delta z + \bar{c}^\top \Delta z, \ \text{s.t.} \ D\Delta z = 0, \quad \bar{G}\Delta z = 0, \tag{4.34}$$

where $\bar{c} = c + Hz$. To obtain the step-length, a standard line search can be performed along this direction that terminates when a new constraint is encountered or when the minimum of the objective function along this direction is reached. The KKT conditions for (4.34) are

$$\begin{bmatrix} H & D^\top & \bar{G}^\top \\ D & & \\ \bar{G} & & \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \nu \\ \Delta \bar{\lambda} \end{bmatrix} = \begin{bmatrix} -\bar{c} \\ 0 \\ 0 \end{bmatrix} \tag{4.35}$$

where $\Delta\bar{\lambda}$ are the Lagrange multipliers associated with the working set.

Similarly to IPMs, ASMs require the solution of a KKT system (4.35) that is almost identical to (4.32) except for the diagonal matrix $-\Lambda^{-1}S$ in the lower left corner and some rows deleted from G. Notice that we cannot derive an augmented system of the form (4.33) in this case. We can, however, recognize that the systems (4.35) solved at each iteration are closely related in that only a column is added to

and/or deleted from the working set matrix $\bar{G}$. Therefore, it is inefficient to solve (4.35) from scratch at every iteration. For ASMs, we can significantly decrease the cost of solving this KKT system (from cubic to quadratic in the unknowns) by simply modifying the matrix factorization computed at a previous iteration. Readers are referred to [268] for a method for updating these factorizations specifically in MPC problems.

The search direction is relatively inexpensive to compute at each iteration in ASMs when the matrix factorizations are updated efficiently, but they may require a large number of iterations if the active set changes a lot. In fact, the computational complexity is exponential in the worst-case and is a strong function of the number of active constraints at the optimum [11]. In comparison, IPMs usually involve a relatively small number of iterations (that is largely insensitive to the number of inequality constraints), but each step is relatively expensive to compute. One pays a "fixed price" to solve the KKT system at each iteration that may overwhelm the expense of the active set solver for problems with few active constraints [11]. Therefore, as a general rule-of-thumb, ASMs are suited for medium-sized problems (as the number of inequality constraints is relatively small) while IPMs are suited for large/sparse problems.

Fast Gradient Projection Methods

Fast gradient projection (GP) methods [185, 186] can be used to iteratively solve constrained optimization problems, such as (4.28). The basic idea stems from the traditional gradient method (aka steepest descent [186]) for solving unconstrained optimizations wherein the new iterate z^{i+1} is computed from the previous iterate z^i based on the gradient at the current iterate, i.e., $z^{i+1} = z^i - \alpha_i \nabla f(z^i)$ where α_i is the step size and $\nabla f(z^i)$ is the gradient of the objective function $f(\cdot)$. For constrained optimization problems, one can simply replace the gradient $\nabla f(\cdot)$ with a projection of the gradient onto the feasible region defined by the collection of equality and inequality constraints. Fast GP methods construct the "optimal" projection of the gradient in the sense that no better convergence ratio can be attained when relying

solely on gradient information for minimizing a convex objective subject to a closed, convex constraint set (see [186, §2.1] for details).

There are a few advantages to fast GP methods that make them a promising alternative to active set and interior point methods in a variety of MPC applications. First, it is possible to compute a practical/tight upper bound on the number of iterations needed to obtain a solution of pre-specified accuracy with fast GP methods [217]. On the other hand, convergence of ASMs can only be guaranteed after a finite number of steps in general (that scales exponentially in the worst-case as every possible combination of active constraints may need to be checked) and bounds on the number of iterations in IPMs are too conservative to be of use in practice. Furthermore, fast GP methods do not involve the solution of a linear system of equations at every iteration, which is often a limiting factor for simple control hardware (e.g., microcontrollers or field-programmable gate arrays).

One major disadvantage of most fast GP methods is that the gradient projection operation (required at every iteration) is difficult to compute in general. Therefore, these methods are limited to cases in which the projection operation is simple, mainly box inequality constraints and no equality constraints. This limits the applicability of certain fast GP methods in MPC to special cases such as box input-constrained MPC problems [217].

4.5 Sparse and Condensed Formulations of MPC

In this section, we present different ways of formulating the MPC problem (4.5) as a QP (4.28) of varying dimension and levels of sparsity. Two formulations are dominant in the MPC community. The first formulation keeps the state and input as decision variables resulting in a relatively large, but sparse QP. In the second formulation, the state variables are eliminated from the optimization problem by substituting the dynamic equations (4.5b) into the objective and constraints. This is commonly referred to as the condensed approach and produces a smaller, but dense QP. We also describe how the different iterative solution methods in Section 4.4 can be applied

to each particular formulation, including currently available software packages that implement these algorithms that are customized for MPC problems.

Sparse

The sparse (aka non-condensed) formulation of (4.5) keeps the future states as decision variables and enforces the system dynamics with equality constraints [215]. In this case, the decision variable vector is

$$z \leftarrow [u_0^\top, x_1^\top, \ldots, u_{N-1}^\top, x_N^\top]^\top,$$

and the QP (4.28) has the following problem data

$$H \leftarrow \begin{bmatrix} R & & & & & & \\ & Q & M & & & & \\ & M^\top & R & & & & \\ & & & \ddots & & & \\ & & & & Q & M & \\ & & & & M^\top & R & \\ & & & & & & Q_N \end{bmatrix}, \; c \leftarrow \begin{bmatrix} r + 2M^\top x \\ q \\ r \\ \vdots \\ q \\ r \\ q_N \end{bmatrix},$$

$$D \leftarrow \begin{bmatrix} B & -I_{n_x} & & & & & \\ & A & B & -I_{n_x} & & & \\ & & & & \ddots & & \\ & & & & & A & B & -I_{n_x} \end{bmatrix}, \; d \leftarrow \begin{bmatrix} -Ax \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$G \leftarrow \begin{bmatrix} F_u & & & & & & \\ & F_x & F_u & & & & \\ & & & \ddots & & & \\ & & & & F_x & F_u & \\ & & & & & & F_N \end{bmatrix}, \; g \leftarrow \begin{bmatrix} f - F_x x \\ f \\ \vdots \\ f \\ f_N \end{bmatrix},$$

in which the matrices are sparse and block banded. The relevant dimensions of the QP (4.28) are then

$$n \leftarrow N(n_x + n_u), \quad p \leftarrow Nn_x, \quad m \leftarrow n_c N + n_N.$$

If the KKT system was solved with a naive dense $LDL^\top$ factorization, it would cost $O(N^3(n_x + n_u)^3)$ flops. As described in [268, 215], exploiting the structure of the coefficient matrix for interior point or active set methods reduces this cost to be $O(N(n_x + n_u)^3)$ flops, which is linear instead of cubic in the horizon. This can be improved upon in ASMs by updating the factorization as discussed in Section 4.4, but the cost is also a function of the number of active constraints.

Due to the receding-horizon implementation of the controller, a sequence of MPC problems (4.5) need to be solved online in which the problem data and/or initial condition vary only slightly from one problem to the next. Solvers can benefit substantially by using this information in the algorithm, e.g., by selecting good initial guesses for the variables. The process of using previous information to solve a perturbed problem is called *warm starting* in the optimization literature [268].

An example of warm starting in MPC is based on the fact that we plan the control policy for the next N time steps. As such, we can use the previously computed trajectories as a good starting point for the current problem [261]

$$\begin{aligned}(x_0, x_1, \ldots, x_N) &= (x, x_2^-, \ldots, x_N^-, x_N^{\text{new}}),\\ (u_0, u_1, \ldots, u_{N-1}) &= (u_1^-, u_2^-, \ldots, u_{N-1}^-, u_{N-1}^{\text{new}}),\end{aligned}$$

where x is the current initial state, u_j^-, x_j^- are the optimal input and state solutions at the previous step, and u_{N-1}^{new}, x_N^{new} are estimates for the final stage.

Warm starting strategies such as this typically save significantly more computation time in ASMs than in IPMs [268]. This is likely due to the fact that IPMs try to follow the central path, which can be quite sensitive to data perturbations near the solution. It may then require many iterations to get back to the central path. ASMs, on the

other hand, look to identify the active set of constraints, which will typically not change significantly from problem to problem. Therefore, they can locate the active set for the new problem starting from the previous active set in a few iterations.

Several attempts have been made to develop efficient warm starting methods for IPMs. A warm starting technique for the primal barrier IPM is presented in [261] wherein the previous solution is used as the initial guess for the first Newton step of the KKT system at the reduced value of the barrier parameter κ. Furthermore, [261] discusses ways to decrease complexity of the algorithm by sacrificing optimality by solving the KKT system at a single fixed value of κ, which produces reasonable closed-loop performance in their case studies.

For IPMs, we can further speed up computations by more efficiently solving (4.33) in *normal equation form* since the matrices are block banded [60]. This compact formulation is obtained from the Schur complement of the coefficient matrix [269]

$$Y\Delta\nu = b \tag{4.36}$$

where $Y = D\Phi^{-1}D^\top \in \mathbb{S}_+^{Nn_x}$, $b = r_D - D\Phi^{-1}r_\Phi$, and $\Delta z = \Phi^{-1}(-r_\Phi - D^\top\Delta\nu)$. The system (4.36) is efficiently solved by Cholesky factorization due to the symmetric block tri-diagonal structure of Y.

There is software available for automatically generating high-speed custom solvers that take into account the structure of the sparse (non-condensed) MPC formulation in the ways outlined above. Two state-of-the-art examples are CVXGEN [158] and FORCES [60], which generate custom primal-dual interior point solvers that can yield computation times that are orders of magnitude faster than generic solvers. Readers are referred to these works for further details.

As discussed previously, the complexity of solvers based on the sparse formulation of the MPC optimization problem (4.5) is roughly $O(N(n_x + n_u)^3)$ since the number of iterations depends weakly on the horizon [215]. Therefore, this formulation is best suited for problems with long horizons as the complexity scales linearly in N. However, systems with a large number of states and/or inputs can still present major

computational challenges for these algorithms as they scale cubically with respect to these variables. As discussed in the next section, one way to mitigate this effect is through condensing of the state variables wherein we produce a smaller, but less structured QP.

Condensed

In the condensed formulation of (4.5), we eliminate the state variables by expressing them as an explicit function of the current state and the input variables

$$\mathbf{x} = \mathbf{A}x + \mathbf{B}\mathbf{u}, \tag{4.37}$$

where

$$\mathbf{A} = \begin{bmatrix} I_{n_x} \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$

In this case, the decision variable vector becomes

$$z \leftarrow [u_0^\top, u_1^\top, \ldots, u_{N-1}^\top]^\top,$$

while the problem data for the inequality-constrained QP (4.28) is defined by

$$H \leftarrow \mathbf{B}^\top \mathbf{Q}\mathbf{B} + \mathbf{R} + 2\mathbf{B}^\top \mathbf{M},$$

$$c \leftarrow (\mathbf{B}^\top \mathbf{Q}^\top \mathbf{A} + \mathbf{M}^\top \mathbf{A})x + \mathbf{B}^\top \mathbf{q} + \mathbf{r},$$

$$D \leftarrow 0,$$

$$d \leftarrow 0,$$

$$G \leftarrow \mathbf{F_x B} + \mathbf{F_u},$$

$$g \leftarrow \mathbf{f} - \mathbf{F_x A} x$$

where

$$\mathbf{Q} = \begin{bmatrix} I_N \otimes Q & 0 \\ 0 & Q_N \end{bmatrix}, \ \mathbf{M} \begin{bmatrix} I_N \otimes M \\ 0 \end{bmatrix}, \ \mathbf{q} = \begin{bmatrix} \mathbf{1}_N \otimes q \\ q_N \end{bmatrix},$$

$$\mathbf{R} = I_N \otimes R, \quad \mathbf{r} = \mathbf{1}_N \otimes r,$$

$$\mathbf{F_x} = \begin{bmatrix} I_N \otimes F_x & 0 \\ 0 & F_N \end{bmatrix}, \ \mathbf{F_u} = \begin{bmatrix} I_N \otimes F_u \\ 0 \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} \mathbf{1}_N \otimes f \\ f_N \end{bmatrix}.$$

The relevant dimensions of the QP (4.28) are

$$n \leftarrow N n_u, \quad p \leftarrow 0, \quad m \leftarrow n_c N + n_N.$$

In this formulation, G is a Topelitz lower block triangular matrix while H is a symmetric positive definite dense matrix, hence the KKT system can be solved using an unstructured Cholesky factorization in $O(N^3 n_u^3)$ flops [114]. Notice that the computational requirements are independent of the number of states, making this approach preferable for problems with a large state dimension. Recently, a method for exploiting the structure of the dense Hessian matrix H in condensed unconstrained LQ problems was described in [75] costing roughly $2N^2 n_x^2 n_u + 3N n_x n_u^2 + (1/3) N n_u^3$ flops. This approach may be faster when the number of states and horizon length are of a

moderate dimension.

Standard IPMs are usually applied in the condensed approach. On the other hand, an active set strategy for the condensed approach has been developed in [68] that exploits the parametric nature of the QP (see Section 4.6 for more details). This strategy builds upon traditional warm starting techniques by fully exploiting the knowledge of the solution of the previous QP under the assumption that the active set does not change much from one QP to the next. The open-source software package qpOASES [69] implements this parametric ASM.

In [217], fast GP methods were applied to the condensed version of the MPC problem. The state variables needed to be eliminated from the primal formulation in order to make the gradient projection operation simple enough to calculate analytically by removing the dynamic equality constraint. The Matlab toolbox FiOrdOs [251] implements this fast GP method, but only works for simple constraints. As discussed in [113], the fast GP approaches are especially well-suited for inexpensive embedded platforms as they only require simple operations instead of the solution to a linear system of equations.

A dual fast GP algorithm was proposed for MPC problems in [196] that can handle general polyhedral state and input constraints. It does this by solving the dual QP that involves only box constraints on the Lagrange multipliers. This formulation, however, uses the non-condensed approach such that is scales cubically with respect to the state and input dimensions.

Quasi-Sparse

The sparse and condensed approaches represent two extremes in which one keeps all or none of the state variables, respectively. The authors of [6] explored the possibility of doing better by combining both approaches. The basic idea is to *partially condense* the state by partitioning the horizon N into $\tilde{N}$ blocks of length M_{block} and then eliminating the states within each block.

As shown in [6], the resulting optimization problem can be interpreted as an MPC problem of the form (4.5) with a virtual horizon $\tilde{N} = N/M_{\text{block}}$, virtual state

dimension $\tilde{n}_x = n_x$, and virtual input dimension $\tilde{n}_u = M_{\text{block}} n_u$. The number of flops required to solve the KKT system can then easily be computed as a function of M_{block}. The theoretically optimal choice for M_{block} is derived in [6] when solving (4.33) with a sparse Ricatti factorization. For a small number of states, the classical sparse approach $M_{\text{block}} = 1$ is optimal whenever $n_x \approx n_u$.

Although the methods discussed in this section can speed-up the solution of the MPC QP substantially, they all require it to be solved online. In the next section, we review a method for pre-computing the solution to this QP offline for all possible values of the initial state. Even though this is only possible for problems with reasonably small dimensions, the online cost of the controller merely requires the evaluation of a lookup table of linear control gains. This method opens the door to applications with very fast sampling times and has been applied to systems with sampling times on the order of 1 μs (see, e.g., [116]).

4.6 Explicit MPC

The main disadvantages to the solution approaches reviewed in Section 4.5 is that they require the solution to a QP online. In this section, we review a popular alternative approach to these methods that is valid for low-dimensional problems and was introduced to the MPC community by [15]. The main idea is to construct an explicit representation of the QP solution offline (i.e., before the runtime of the process). See, e.g., [2] for a recent review on explicit MPC.

Piecewise Linear Solution

The MPC problem (4.5) can be written in terms of a number of equivalent QP representations (see Section 4.5). All possible QPs differ only in the gradient $c(x)$ in the objective function and the constraint vector $g(x)$, which are affine functions of the current state x [68]. This means that the MPC problem is defined by a *parametric* QP with a particularly favorable structure, such that the control law $\kappa_N(x)$ is completely parametrized by the current value of the state x.

When (4.5) is of a small enough dimension (usually with the state, input, and constraint dimensions all lower than ~ 10), the QP can be solved offline to determine $\kappa_N(x)$ as an explicit function of x. This is done by noting that the domain of attraction $\mathscr{X}_N$ can be subdivided into N_c polyhedral critical regions Π_i on each of which the set of active constraints is constant. The critical regions are mutually exclusive and collectively exhaustive with respect to the domain of attraction. The result is a piece-wise affine (PWA) control law of the form

$$\kappa_N(x) = \begin{cases} F_1 x + g_1, & \text{if } x \in \Pi_1 \\ F_2 x + g_2, & \text{if } x \in \Pi_2 \\ \vdots & \\ F_{N_c} x + g_{N_c}, & \text{if } x \in \Pi_{N_c} \end{cases}$$

Explicit MPC can easily be implemented using built-in functions available in the Matlab toolbox MPT3 [100]. After the control law $\kappa_N(x)$ is calculated explicitly offline, the only online computations will be a set membership calculation, a matrix-vector multiplication, and a vector addition. Therefore, the major advantage of explicit MPC is the very small online computational cost.

The largest drawback of explicit MPC is the large offline computational cost, which generally scales poorly with the number of states and constraints, and the space requirements of storing Π_i, F_i, and g_i. The online set membership test can also be prohibitively computationally expensive for large problems. These drawbacks have limited the application of explicit methods to small problems. Another drawback is that explicit MPC is best applied when $\kappa_N(x)$ is time-invariant, making applications difficult for time-varying systems, trajectory tracking, and online tuning [68].

Methods for Complexity Reduction

As mentioned above, when there are many complicated critical regions Π_i, storing these sets and evaluating membership can be difficult for large problems. This has

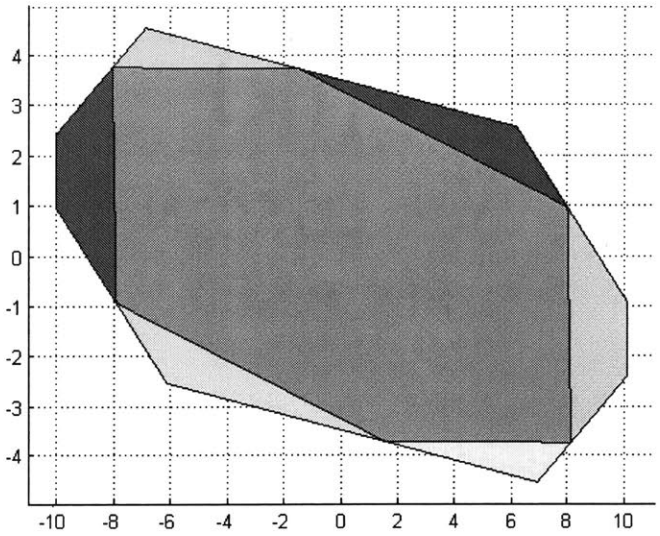


Figure 4-1: Critical regions Π_i for a randomly generated example.

motivated the development of various methods for replacing the complicated explicit control law $\kappa_N(x)$ by a simpler control law $\tilde{\kappa}_N(x)$. These methods can be divided into two different classes: 1) approximate methods, where $\tilde{\kappa}_N(x)$ is chosen to approximate $\kappa_N(x)$, and 2) exact methods, where $\tilde{\kappa}_N(x) = \kappa_N(x)$ for all x in $\mathscr{X}_N$.

The challenge with approximate explicit MPC methods is to approximate $\kappa_N(x)$ by a less complicated function $\tilde{\kappa}_N(x)$ while preserving properties of the original controller $\kappa_N(x)$, e.g. recursive feasibility, stability, and constraint satisfaction of the closed-loop system. For example, [135] approximates $\kappa_N(x)$ by a single polynomial of pre-specified degree by solving a single linear program. [244] proposes approximating $\kappa_N(x)$ by solving a simpler MPC optimization problem (for example, one with a shorter prediction horizon) and finding a PWA function $\tilde{\kappa}_N(x)$ defined over the critical regions of the simpler problem which minimizes the error between $\kappa_N(x)$ and $\tilde{\kappa}_N(x)$. Another method involves performing explicit MPC on a reduced-order model, although it may be difficult to use bounds on the approximation error to preserve guarantees [108].

One method for simplifying the control law $\kappa_N(x)$ involves deriving an equivalent

PWA function $\tilde{\kappa}_N(x)$ which is minimal in the number of polyhedra [80]. This method, called optimal region merging (ORM), can in some cases reduce the complexity of $\kappa_N(x)$ by an order of magnitude, with the downside being that ORM is NP-hard. Another approach is to replace regions in which $\kappa_N(x)$ obtains a saturated value by extending regions in which the value of $\kappa_N(x)$ is unsaturated. This procedure defines a resulting control law $\hat{\kappa}_N(x)$ and a clipping function $\phi(\cdot)$ so that $\tilde{\kappa}_N(x) = \phi(\hat{\kappa}_N(x))$ is equivalent to $\kappa_N(x)$ for all x in $\mathscr{X}_N$. An extension of this technique uses a function which strictly separates saturated regions to eliminate these regions from the definition of the control law [134].

4.7 Example: Quadratic Dynamic Matrix Control

In the early 1970s, engineers at Shell Oil developed an MPC technology for constrained multivariable systems, which they named dynamic matrix control (DMC). The DMC algorithm uses a linear step response model to predict the future behavior of the plant while the optimal inputs are computed as the solution to a least-squares problem that tries to drive the output to follow the stepoint as closely as possible [54]. To explicitly handle input and output constraints, Shell engineers posed the DMC algorithm as a QP. This algorithm, termed quadratic DMC (QDMC), was published in a comprehensive paper in 1986 [76]. One of the key features in QDMC is the constant disturbance update rule used to provide feedback within the control algorithm. The simplicity and power of this methodology has had an enormous impact on industrial MPC, especially within chemical process industries. See, e.g., [211] for a review of industrial MPC technology.

Step Response Modeling

In QDMC, a finite step response (FSR) model is used to predict the dynamic response of the output to changes in the input from some steady state. For the single-input

single-output (SISO) case, the FSR model is of the form

$$y_{k+j} = \sum_{i=1}^{n-1} s_i \Delta u_{k+j-i} + s_n u_{k+j-n} \tag{4.38}$$

where s_i are the step response coefficients, which are defined as the integral of the impulse response, and is equivalent to the output value at time step i when a unit step in the input is applied at time zero. The summation is truncated whenever the input no longer effects the future output; truncated at n in (4.38), which is commonly referred to as the *model length*. This model will only be accurate for stable linear systems in which $s_n \approx s_{n+1} \approx \ldots \approx s_\infty$. Note that the step response coefficients can be obtained from a model of the plant or from data.

Multiple inputs and multiple outputs are handled by superposition in which many SISO models can be stacked into one model of the form (4.38) where s_i are matrices (instead of scalars) of the appropriate dimensions [142]. We can also readily feedforward measured disturbances by modeling the effect of the measured disturbance on the output using an FSR model (4.38).

State-Space Interpretation

The FSR model (4.38) can be put into the standard state-space form (4.1)–(4.2), as was demonstrated in [142, 139], where the states, inputs, and outputs are chosen to be

$$x(k) \leftarrow [y_0(k)^\top, y_1(k)^\top, \ldots, y_{n-1}(k)^\top]^\top, \quad u(k) \leftarrow \Delta u(k), \quad y(k) \leftarrow y_0(k),$$

and the problem data is of the form

$$A \leftarrow \begin{bmatrix} 0 & I_{n_y} & 0 & \cdots & 0 \\ & 0 & I_{n_y} & \ddots & \vdots \\ \vdots & & 0 & \ddots & 0 \\ & & & \ddots & I_{n_y} \\ 0 & & \cdots & & I_{n_y} \end{bmatrix}, \quad B \leftarrow \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix},$$

$$C \leftarrow \begin{bmatrix} I_{n_y} & 0 & \cdots & 0 \end{bmatrix}.$$

The vector of states in this case has dimension $n_x \leftarrow n_y n$ and represent the so-called *dynamic states* or *free response* of the system. In this section, we denote the dynamic states at time step k as $Y(k)$. In fact, each element $y_j(k)$ of $Y(k)$ can be interpreted as the output at time $k+j$ assuming constant inputs into the future. Thus, $Y(k+1)$ is merely the previous "states" $Y(k)$ shifted up by n_y elements plus the contributions from the most recent input. Note that the A matrix in this case is very sparse and the dynamic states are only a function of the model length and the output dimension (independent of the "true" state dimension n_x).

Output Feedback via Disturbance Update Rule

Using the FSR (4.38), one can write the predicted future outputs as a linear combination of future input moves. However, the measured plant outputs will be different than the predicted values due to measurement noise and unmeasured disturbances. Therefore, we must make some assumption about these before we can proceed. QDMC uses an additive disturbance assumption in which the future unmeasured disturbances remain constant over time and can be estimated as the difference between the current measurement and the current predicted output. The QDMC predictor is then de-

scribed by the following equations [149]

$$\mathscr{Y}(k+1|k) = M_p\bar{Y}(k) + S_p^m\Delta\mathscr{U}(k) + \mathscr{I}[y(k) - \bar{y}(k)], \tag{4.39}$$

where $\mathscr{Y}(k+1|k)$ is the predicted outputs over a horizon of p future time steps given current information at time step k, M_p is the first pn_y rows of the A matrix from Section 4.7, $\bar{Y}(k)$ and $\bar{y}(k)$ are the dynamic states and model output from the state-space recursion form of the FSR model described above, respectively, $y(k)$ is the measured output from the plant, $m \leq p$ is the control horizon (after which the input is assumed to remain constant to reduce complexity), $\Delta\mathscr{U}$ is a vector of m future input moves, $\mathscr{I} = [I_{n_y}, \ldots, I_{n_y}]^\top$, and S_p^m, given by,

$$S_p^m = \begin{bmatrix} s_1 & 0 & & 0 \\ s_2 & s_1 & 0 & 0 \\ & & & \\ s_m & s_{m-1} & & s_1 \\ & & & \\ s_p & s_{p-1} & & s_{p-m+1} \end{bmatrix},$$

is the so-called *dynamic matrix* that relates the predicted future outputs to the future input moves. Note that this procedure is very related to the processes of condensing discussed in Section 4.5. As such, the optimization cost in QDMC is completely independent of the state dimension n_x and of the dynamic state dimension $n_y n$. Since the inputs are fixed after the control horizon m, the cost is also independent of the prediction horizon p. This technique for reducing the number of decision variables is commonly referred to as *move blocking*. Thus, the complexity of solving the QDMC optimization problem using a standard QP solver is roughly $O(m^3 n_u^3)$.

The objective function is chosen as a quadratic function similar to that in (4.5). In fact, we can exactly represent the QDMC algorithm as a special case of the general MPC algorithm (4.5) by selecting our state-space system to be of the form (4.26)–(4.27) with $B_d = 0$ and $C_d = I_{n_y}$. This is done by initializing the state with the

observer (4.22)–(4.23) using a filter gain $L = [0, I_{n_y}]^\top$ on the augmented system. It has been shown in [139] that this filter gain is optimal for stochastic integrated white noise disturbances (random steps) at each output and noise-free measurements. Interested readers are referred to [177, 139, 149] and the references therein for further details.

When to use QDMC: Pros and Cons

The recent academic literature on MPC has favored the state-space formulation due to its generality. Often, input-output formulations such as QDMC are viewed as limited. This way of thinking is well-illustrated from a paper in the mid-1990s titled "Limitations of Dynamic Matrix Control" [149]. The three limitations described in this paper are 1) good performance may require a large number of step response coefficients, 2) poor performance may be observed for disturbances affecting the plant inputs or ramp-like disturbances on the output, and 3) poor robust performance for multivariable plants with strong interactions [149]. We focus on the first and second limitations below.

Although these can be viewed as limitations of DMC and QDMC in many situations, they may have advantages in certain cases that are often overlooked. For example, although one may need a relatively large model length n for systems with slow time constants, the dynamic state dimension can still be significantly lower than the original state dimension of the plant model. In this case, the FSR model is a simple/accurate reduced-order model. The FSR model also introduces significant sparsity as discussed previously. As such, the QDMC algorithm is a very efficient alternative to state-space MPC for large-scale systems with a high-dimensional state vector.

In terms of the second limitation, although the disturbance update rule is not optimal for non-additive disturbances, it completely avoids the need for state estimation. This is important because state estimation can be quite expensive depending on the methodology used and may not even be possible when the system is not observable, which is commonly the case for systems with high state dimension. Therefore,

it is extremely cheap from a computation point-of-view. It is also a generic rule that works regardless of the true disturbance affecting the plant. Although it may not be optimal, designing a better rule requires additional knowledge about how the disturbances enter the system. Since these disturbances are unknown by definition, it is often difficult to choose a better assumption beforehand. Additionally, optimal rules for other types of disturbances can be sensitive and may lead to performance and robustness issues.

It is also important to note that since QDMC is merely a special case of the more general MPC problem (4.5), many of the ideas touched on in Section 4.3 can also be incorporated into the QDMC algorithm. For example, we can set $p = \infty$ by adding a terminal constraint based on the solution to a Ricatti equation. In essence, the key concepts of input-output modeling and the disturbance update rule (for output feedback) can be used to reduce the cost of the MPC optimization problem and overall algorithm in a simple manner as well as can be nicely blended with more recent advances in the field of MPC. Furthermore, QDMC formulates the optimization problem in a condensed form (see Section 4.5), which is the best choice for systems with many states.

Case Study: A Hyperbolic Distributed Parameter System

Hyperbolic partial differential equations (PDEs) are commonly used to model an important class of problems that arise in applications exhibiting wave-like behavior. Such PDEs are found in a variety of fields including quantum mechanics, elastic and plasma physics, acoustics, and fluid dynamics. The second-order hyperbolic PDE of interest is given by

$$\frac{\partial^2 E}{\partial t^2} + \gamma \frac{\partial E}{\partial t} + \lambda E(z,t) = c^2 \frac{\partial^2 E}{\partial z^2} + u(t), \tag{4.40}$$

for $z \in [0, L]$ and $t \in \mathbb{R}_{\geq 0}$ with initial conditions

$$E(z,0) = 0, \quad \left.\frac{\partial E}{\partial t}\right|_{t=0} = 0, \tag{4.41}$$

and boundary conditions

$$E(0,t) = E(L,t) = 0, \tag{4.42}$$

where $E(z,t)$ is the dimensionless distributed state, z is the spatial coordinate from zero to length L, t is continuous time, c is a constant related to the propagation speed of the wave, γ is a damping coefficient, and λ is a constant external restoration factor. The control input $u(t)$ appears directly in the PDE as a time-dependent forcing function. The output of interest is the state at the center of the domain, i.e., $y(t) = E(L/2,t)$.

The open-loop transfer function $G(s)$ can be analytically derived to be

$$G(s) = \frac{1}{c^2}\left[\frac{e^{\sqrt{\alpha}L} - 2e^{\sqrt{\alpha}\frac{L}{2}} + 2e^{-\sqrt{\alpha}\frac{L}{2}} - e^{-\sqrt{\alpha}L}}{\alpha(e^{\sqrt{\alpha}L} - e^{-\sqrt{\alpha}L})}\right] \tag{4.43}$$

where $\alpha = (1/c^2)(s^2 + \gamma s + \lambda)$. The poles, found by solving for the roots of the denominator of $G(s)$, are

$$-\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{\gamma^2 - 4\left(\gamma + \frac{\pi^2 n^2 c^2}{L^2}\right)},$$

for all $n \in \mathbb{I}_{\geq 0}$. There are an infinite number of poles as this is a distributed parameter system. For realistic parameter values of $c^2, \gamma, \lambda, L > 0$, all poles will have negative real parts, thus, the system is open-loop stable.

We look to apply QDMC to this system to illustrate how fast the algorithm can be for high state dimension systems while still providing very good performance. We selected the parameters to be $c = 1\ \frac{\text{m}}{\text{s}}$, $\gamma = 1\ \text{s}^{-1}$, $\lambda = 1\ \text{s}^{-2}$, and $L = 1\ \text{m}$. The control objective was to track a sinusoidal reference trajectory subject to input constraints $u \in [-10, 10]$, actuator constraints $\Delta u \in [-10, 10]$, and output constraints $y \in [-10, 10]$.

Since the system (4.40) is open-loop stable, we can generate a step response model (4.38) by performing a unit step test as shown in Figure 4-2. The plant output is

shown in Figure 4-3 while the corresponding input profile, computed using QDMC, supplied to the plant is shown in Figure 4-4. In the simulation, the prediction and control horizon were both chosen to be 30 seconds. Furthermore, an additive measured load disturbance occurs at time 6.5 seconds, an additive unmeasured output disturbance occurs at time 10.5 seconds, and the output measurements are continually corrupted with white noise.

Notice that the QDMC controller almost perfectly rejects the load disturbance through its feedforward action, even with the relatively fast-changing reference trajectory. Additionally, the unmeasured disturbance is also quickly rejected due to the simple disturbance update rule that provides feedback. This controller can very easily be implemented in an online fashion, as each QP takes only approximately 0.01 seconds to solve. This is in direct contrast to a standard state-space implementation of the MPC controller using a discretized version of (4.40), which could take on the order of minutes (or longer) per QP depending on the fineness of the discretization.

To get a feel for the impact on computation time, in Figure 4-5, we compare a naive implementation of the sparse MPC formulation to QDMC for this example. Both optimization problems were solved using the Matlab® built-in `quadprog` function. We can see that the time required to solve the QDMC optimization problem is independent of the number of states in the original model as only the FSR model is used for optimization. On the other hand, the naive state-space MPC optimization problem (where the states are kept as decision variables) is a strong (cubic) function of the number of states used to discretize (4.40). Although this can be improved by taking into account sparsity and applying better heuristics, the poor scaling with respect to state dimension is inherent in the formulation of the problem.

4.8 Concluding Remarks

In this chapter, we reviewed and discussed the many methods that have been developed to efficiently solve the convex optimization problem arising in MPC problems for linear systems. We focused on three different optimization methods for solving

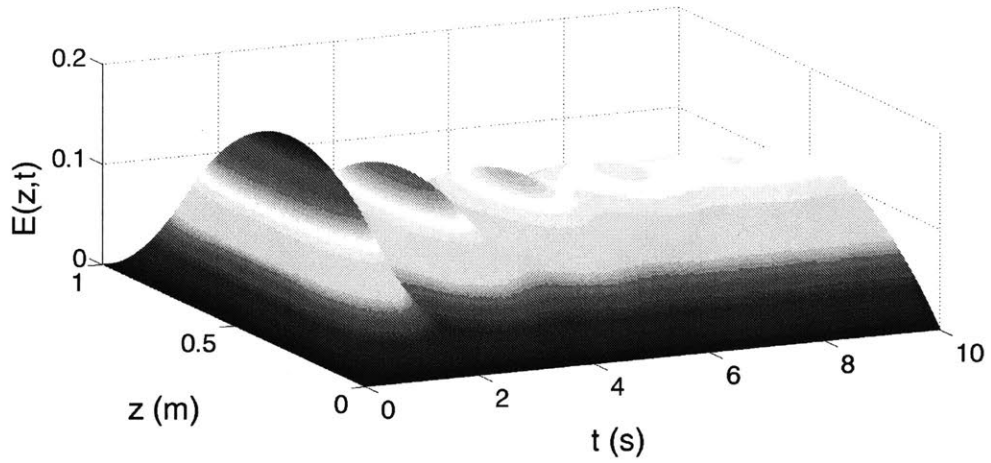


Figure 4-2: Solution to (4.40) for a unit step in the input. The output values, and thus step response coefficients, are directly computable from the $E(1/2, t)$ line on this plot.

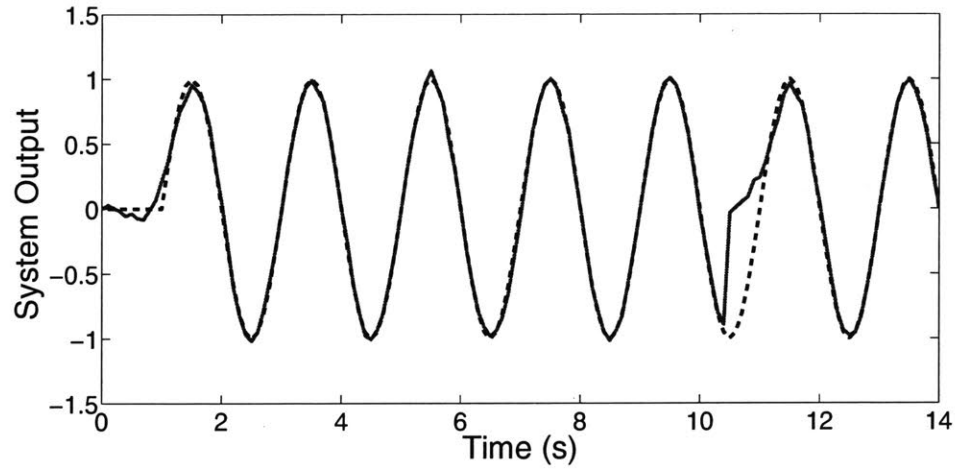


Figure 4-3: Closed-loop response of the hyperbolic PDE system output. See Section 4.7 for the details of the simulation.

this problem including interior point methods, active set methods, and fast gradient projection methods.

We discuss how, not only the optimization method used, but also how the MPC optimization problem is formulated can greatly impact the solution speed. In fact, we demonstrate that the fastest method and formulation for a particular application will be a strong function of the state and input dimensions as well as the number of

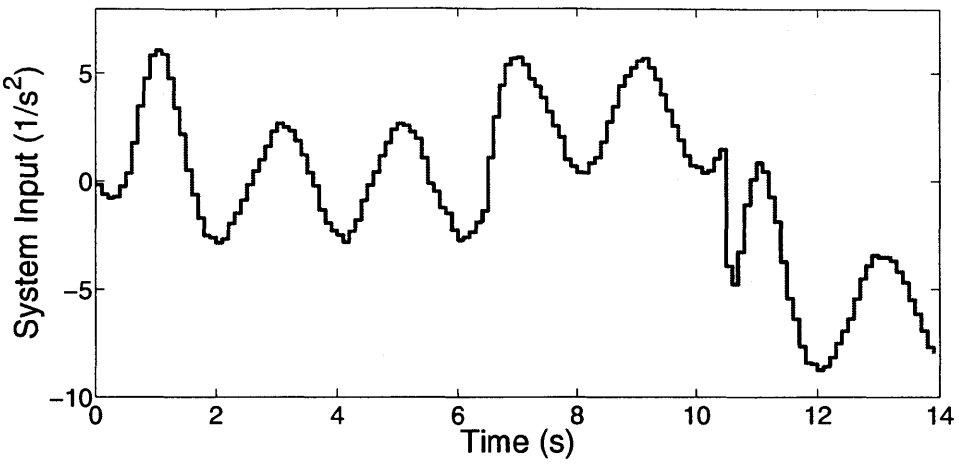


Figure 4-4: Input profile, computed using the QDMC algorithm, supplied to the hyperbolic PDE system and corresponds to the output profile shown in Figure 4-3. See Section 4.7 for the details of the simulation.

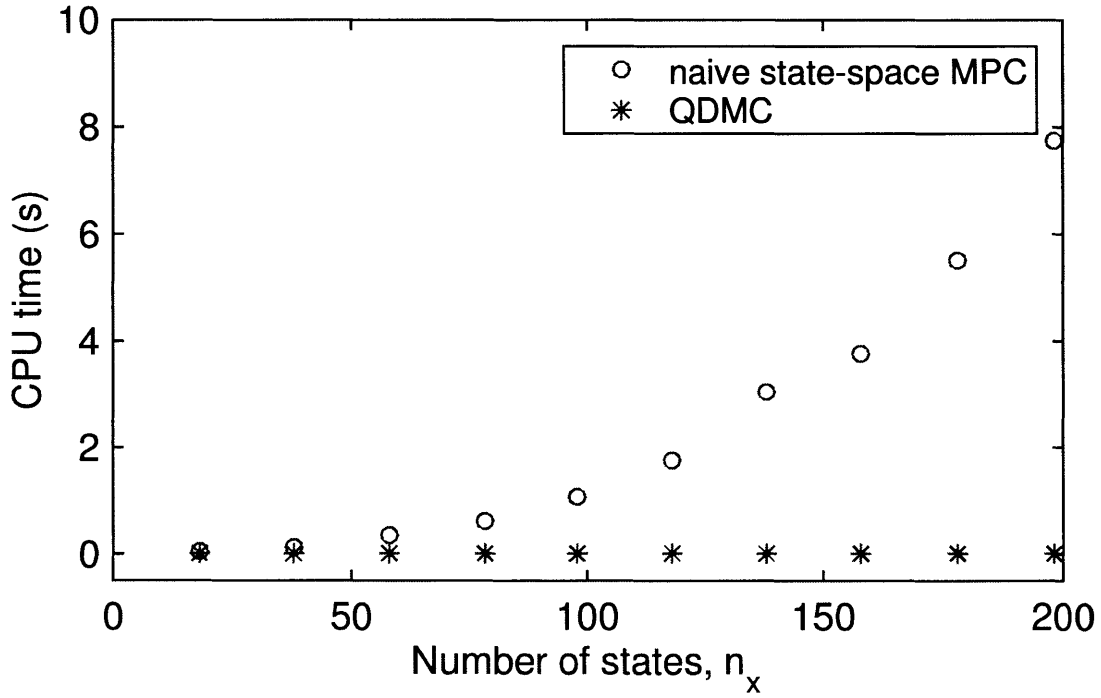


Figure 4-5: Comparison of CPU times (of a single optimization problem averged over 20 runs) versus number of states in the true system for a naive implementation of sparse state-space MPC and QDMC. The algorithms were coded in Matlab® and run on a laptop PC (Intel i7, 2.7 GHz, 8 GB RAM).

constraints included in the optimization. For small enough problems, an explicit MPC method, which completely solves the parametric optimization offline for all possible initial conditions, will be the fastest method as the only online cost will be the cost of traversing a lookup table. However, the number or regions that one must compute grows exponentially with the size of the problem and quickly becomes intractable for state, input, and constraints dimensions larger than around five.

One particularly popular MPC approach used in industry is QDMC. The main advantages of QDMC are its input-output modeling framework and simple, yet effective, feedback strategy. We discuss how QDMC is a particular case of the more general state-space MPC as well as demonstrate its performance and speed on a distributed parameter system whose dynamics are governed by a PDE. We also note that the disturbance update rule used in QDMC is an optimal state estimator for integrated white-noise disturbances on the output and noise-free measurements.

Although only touched on briefly, accounting for uncertainty directly in the synthesis of the MPC controller is important for guaranteeing safe operation and good performance, and is still a very active area of research. Many forms of uncertainty may be present in practical problems including time-varying disturbances, sensor noise, bias, and drift, parametric uncertainty, and model structure errors. Many different strategies have been developed to account for these types of uncertainties. The term robust MPC broadly refers to methods in which the uncertainty is assumed to lie in a deterministic and bounded set (see, e.g., [14]) while stochastic MPC assumes that the uncertainty is described by an underlying probability distribution (see, e.g., [164]).

Many of the results discussed in this chapter can be applied to robust and stochastic MPC algorithms, especially as they share the same multistage structure as nominal MPC. One interesting route for handling additive disturbances and measurement noise is so-called *tube-based MPC* [160]. Since the tube-based MPC problem reduces to almost exactly a nominal MPC problem of the form (4.5), its structure can be exploited in exactly the same way. As an example of this, real-time implementable tube-based MPC has been explored in [277].

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Plant-wide Control for Continuous Pharmaceutical Manufacturing

5.1 Introduction

In recent years, there has been a growing interest in the pharmaceutical industry to adopt novel synthesis and manufacturing approaches [140, 242]. The paradigm shift from conventional pharmaceutical manufacturing (i.e., batch-wise processing) is mainly driven by the demands for enhanced sustainability, reliability, and cost-effectiveness of pharmaceutical processes, as well as the need for novel synthesis pathways [203]. *Integrated continuous manufacturing* (ICM) has received increasing attention to realize efficient and cost-effective pharmaceutical production. Recent studies have demonstrated that ICM can substantially reduce environmental footprint, manufacturing times, and costs compared to the existing batch pharmaceutical processes [218, 17, 223]. In addition, the increased use of online monitoring (i.e., process analytical technology [270]) in ICM leads to improved real-time understanding of process dynamics. This understanding facilitates *online control* to achieve consistently high-quality product in the presence of process uncertainties and disturbances by taking corrective actions before product goes off-spec.

Online control is an alternative to so-called design space-based control strategies in pharmaceutical manufacturing [145]. A *design space* is defined as the multi-

dimensional space of *critical process parameters* (CPPs)–input variables and process parameters–that has been demonstrated to result in acceptable *critical quality attributes* (CQAs) of the product. The design space-based control strategies are typically devised to monitor (and possibly control) the CPPs within the design space. Such control approaches enable robust process operation in that process variations within the design space (due to uncertainties and disturbances) can be readily accounted for without any active control (i.e., quality by design [276]). However, establishing a design space involves a costly and exhaustive practice of identifying a high-dimensional space of CPPs a priori, which is often of limited applicability during process scale-up [145]. In particular, the establishment of a design space is likely to be more challenging for ICM processes. This is because the interactions between several process units connected through a network of mass and energy streams can result in an excessively large design space, and any unexplored attractive region of operation (i.e., set of CPPs leading to adequate product quality) can drastically reduce flexibility in process operation. On the other hand, when the product CQAs are monitored in real-time, feedback control can be applied to retain the CQAs within their admissible limits by actively manipulating CPPs to counteract process variations. Hence, active control is likely to enable a more robust and flexible process operation compared to design space-based control strategies.

The key challenge in active control of a continuous pharmaceutical manufacturing process arises from complex plant-wide dynamics of the integrated process units. Advanced control of isolated process units (e.g., crystallizers, thin-film processing units, granulation units, compaction units, etc.) in pharmaceutical processes has been extensively investigated (e.g., see [259, 109, 213, 182, 167, 165] and the references therein). However, the shift from batch-wise processing to ICM requires harnessing the multivariable dynamics of a plant composed of several interconnected units with recycle, bypass, and heat streams, which can significantly increase the complexity of the plant-wide dynamics. The plant-wide interactions in an ICM process often result in poor performance of decentralized control systems at the process unit level. Hence, a plant-wide control strategy should be designed for the integrated process units to

realize the stringent regulatory requirements on the CQAs of the end product.

This chapter investigates plant-wide model predictive control of an end-to-end integrated continuous pharmaceutical manufacturing pilot plant. Model predictive control (MPC) is the most commonly applied approach for advanced control of complex dynamical systems due to its ability to systematically deal with multivariable dynamics, system constraints, and competing sets of objectives [174]. The ICM process manufactures a pharmaceutical product from start (synthesis of intermediate compounds) to finish (molded tablets in final dosage form) in a fully continuous mode [157]. This includes chemical synthesis, purification, formulation, and tableting. A nonlinear plant-wide dynamic model of the pilot plant is used to simulate the dynamics of the real process[1] [18]. The plant-wide dynamics are described by a set of nonlinear differential algebraic equations (DAEs) with nearly 8,000 state variables. The *quadratic dynamic matrix control* (QDMC) algorithm [76] is used to devise an input-output control framework for plant-wide MPC of the integrated continuous manufacturing pilot plant. The input-output framework of QDMC is independent of the state dimension. Therefore, QDMC alleviates the prohibitive costs of plant-wide control of ICM processes with a large state dimension, as such processes typically have a relatively small number of inputs (CPPs) and outputs (CQAs). In addition, QDMC enables incorporating output constraints into the control problem. This is particularly important for online control of pharmaceutical processes, as quality-by-design considerations (i.e., design spaces) can be explicitly accounted for in the plant-wide MPC framework.

To design a plant-wide QDMC system, two modeling approaches are investigated to obtain a linear time-invariant (LTI) approximation of the nonlinear plant dynamics. In the first approach, subspace identification [148] is applied to identify a low-dimensional state-space description of the plant dynamics using input/output data generated by the plant simulator around some desired steady state operating condition. Alternatively, the set of nonlinear DAEs is readily linearized around the steady

[1]Certain features of the plant simulator (e.g., recycle streams) have not been implemented in the real ICM pilot plant, which was built at the Novartis-MIT Center for Continuous Manufacturing.

state operating condition to arrive at a high-dimensional state-space model based on first principles. The developed LTI descriptions of the plant-wide dynamics are used to characterize the finite step response (FSR) dynamics[2] of the plant and, subsequently, design two plant-wide QDMC systems.

The performance of the plant-wide QDMC systems is assessed in closed-loop operation with the nonlinear plant simulator under various scenarios for process uncertainties and disturbances, as well as setpoint changes. To demonstrate the potential benefits of plant-wide model predictive control for the ICM process, the performance of the QDMC systems is compared against that of a plant-wide regulatory control system presented in [136].

Organization The integrated continuous pharmaceutical manufacturing pilot plant is presented in Section 5.2. Section 5.3 describes the data-driven and first-principles approaches adopted to develop a linear time-invariant state-space representation for the nonlinear plant-wide dynamics using the existing plant simulator. The formulation of the plant-wide MPC problem for the ICM process considered in this chapter is presented in Section 5.4 along with simulation results comparing MPC and a regulatory control system under various uncertainties and disturbances.

5.2 Process Description

A schematic representation of the integrated continuous pharmaceutical manufacturing pilot plant is depicted in Figure 5-1 (see [18, 157] for a detailed description of the process). The target active pharmaceutical ingredient (API) is aliskiren hemifumarate (compound **6** in Figure 5-2), which is synthesized from aliskiren (intermediate compound **5**). The process consists of several units for synthesis and purification of intermediates and the API, followed by a series of downstream units in which excipients are added to the API and tablets are formed. The process starts with mixing the inter-

[2]FSR models are commonly generated in industry by perturbing the inputs of the plant during operation around a steady-state point and measuring the resulting outputs. This alternative method, of developing FSR models from the first principles model equations, does not require any plant data such that material need not be wasted when generating the model.

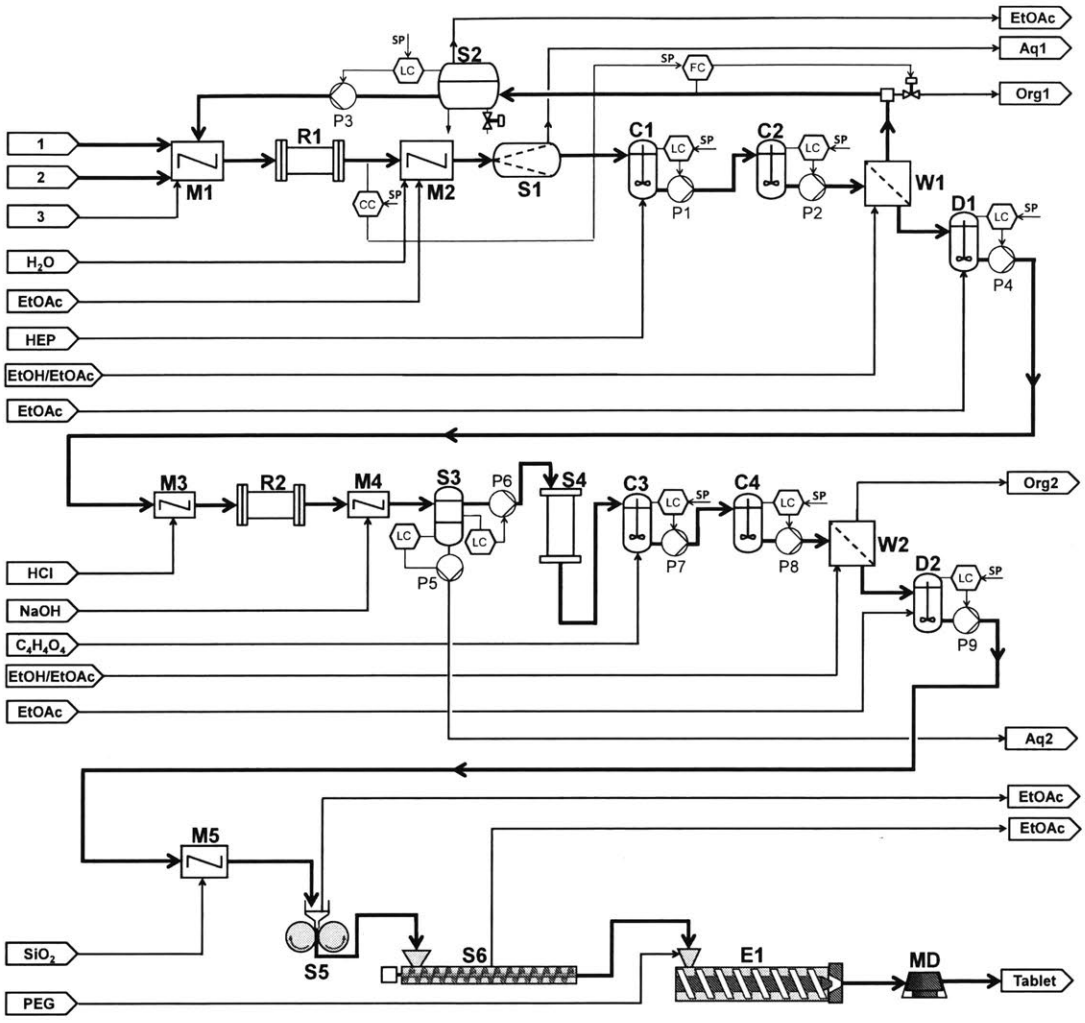


Figure 5-1: Integrated continuous pharmaceutical manufacturing pilot plant equipped with a stabilizing control layer. R reactor, S separator, C crystallizer, M mixer, W washing/filtering unit, D dilution tank, E extruder, MD molding unit, P pump, CC concentration control, FC flow control, LC level control, SP setpoint.

mediate compound **1** with amine **2** and acid catalyst **3** (see Figure 5-2), which is then fed to a tubular reactor (**R1**) to produce the intermediate compound **4**. Water and ethyl acetate (EtOAc) are added to the reactor outlet stream to solubilize the reagents before cooling in a static micromixer (**M2**). The two-phase stream is separated in a membrane-based liquid-liquid separator (**S1**), from which an aqueous-phase stream (Aq1) containing **2** and **3** is purged. The outlet stream of **S1** (containing **1** and **4**) is fed into a two-stage, mixed suspension, mixed product removal (MSMPR) crystal-

lization unit (**C1** and **C2**) to crystallize the intermediate compound **4** by antisolvent heptane (HEP) addition. The crystallization slurry in then fed into a continuous filter (**W1**) to wash and filter crystals with ethanol (EtOH) and EtOAc to remove mother liquor. The permeate stream of the filter unit **W1** contains a substantial amount of reactant **1** and, therefore, is recycled back to reactor **R1**. A flash evaporator (**S2**) is used to remove EtOAc from the recycled stream. A fraction of the material in the recycle loop is purged (Org1) to avoid excessive buildup of impurities (i.e., reaction byproducts) in the process.

The purified crystals of the intermediate compound **4** in the outlet stream of **W1** are diluted with EtOAc in a dilution tank (**D1**) to adjust the concentration of **4** for the second reaction (i.e., **4** to **5** in Figure 5-2). The slurry of compound **4** in EtOAc is mixed with aqueous hydrogen chloride, and fed into a tubular reactor (**R2**) to perform acid-catalyzed removal of the Boc protecting group (Boc=tert-butoxycarbonyl) from **4**. The reactor outlet stream, which contains the second intermediate compound **5**, is quenched with sodium hydroxide (NaOH) to neutralize the acid catalyst. The two-phase mixture is then separated in a decanter (**S3**), from which the aqueous phase (Aq2) is purged. The organic-phase stream containing compound **5** is passed through an adsorption column (**S4**) to remove the traces of water, as EtOAc is the main solvent that can be used in the subsequent units. The API **6** is formed in a reactive crystallization step (**C3**), in which fumaric acid reacts with the second intermediate compound **5**. The API is initially synthesized in the first MSMPR vessel (**C3**), and the yield is further increased in a second MSMPR vessel (**C4**). The API crystals are purified in a combined washing and filtration unit (**W2**), similar to **W1**, and then fed into a dilution tank (**D2**) to adjust the concentration of the API wet cake by adding EtOAc.

Prior to tablet formulation, the first excipient (SiO_2) is added to the crystal slurry to improve the flowabilty of the needle-shaped API crystals. This is followed by two drying steps. The bulk of EtOAc is evaporated in a double drum dryer (**S5**) and, subsequently, the traces of the solvent are removed in a screw dryer (**S6**). The dried powder is then mixed with polyethylene glycol (PEG) to improve the stability of the

final tablets. The powder mixture is conveyed to an extruder (**E1**), which is coupled to a molding unit (**MD**) that forms tablets with a defined geometry. At the end of the process, the solvent content, total impurity content, and API dosage of the final tablets are measured using a near infrared instrument. In addition, the production rate of the tablets is measured. In this study, the potential CPPs consist of flow rates of all the inlet reactant, solvent, and excipient streams to the process. The API dosage and total impurity content constitute the primary CQAs of the manufactured tablets, whereas the production rate is considered as the secondary CQA.

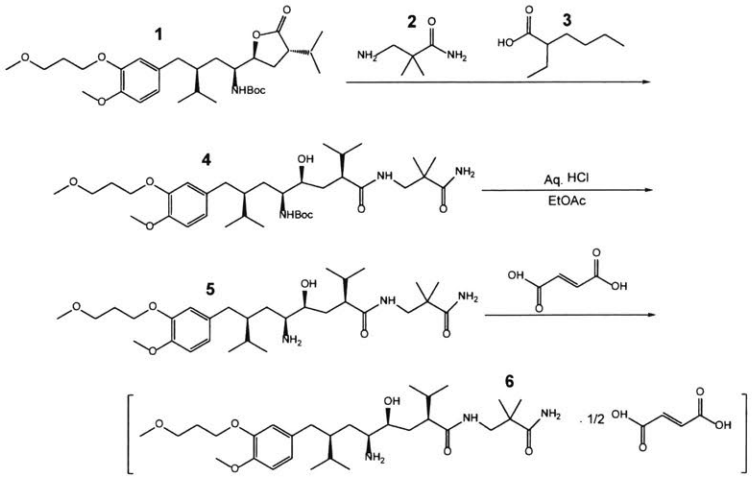


Figure 5-2: Synthetic reactions from intermediate **1** to active pharmaceutical ingredient **6**.

The ICM pilot plant depicted in Figure 5-1 is equipped with a stabilizing control layer to maintain sufficient holdup in each vessel. The stabilizing control layer consists of proportional-only level controllers (LC), which regulate the outlet flow rate from the vessel using a pump (P). In addition, a control loop is established to reject disturbances in the recycle stream to reactor **R1**. An in-line measurement instrument is used in the outlet stream of the reactor to measure the concentration of the reaction effluents to control the concentration of the byproducts recycled back to **R1**. This is done by cascade control of the flow rate of the purge stream (Org1), where the manipulated variable of a concentration controller (CC) serves as the setpoint of a flow controller (FC) implemented on the purge stream (see [136] for the design of the

stabilizing control layer).

5.3 Control-relevant Process Modeling

This section presents two modeling approaches to obtain a linear time-invariant description of the plant-wide dynamics in terms of a state-space model that can be used for designing the plant-wide MPC. A plant simulator is utilized to simulate the nonlinear dynamics of the end-to-end integrated continuous pharmaceutical manufacturing pilot plant [18]. The plant simulator is developed in the JACOBIAN simulation platform (RES Group, Inc.) based on first principles (i.e., mass, energy, and moment conservation laws) and empirical equations that describe physicochemical phenomena such as reaction kinetics, crystallization kinetics, and washing/filtration characteristics. The stabilizing control layer depicted in Figure 5-1 is incorporated into the plant simulator.

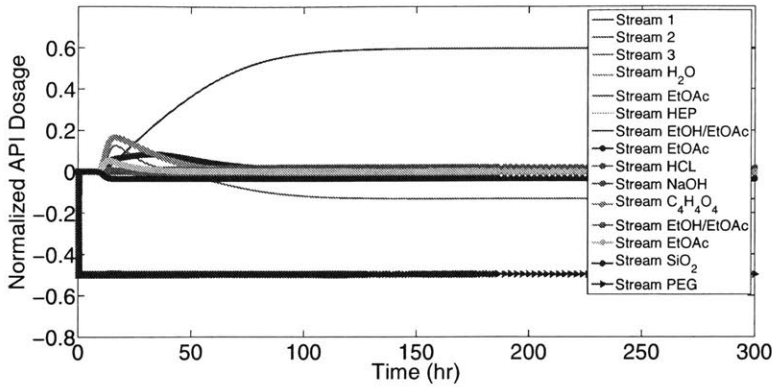
Identification of a Low-dimensional Model

System identification is a common engineering practice to build models of dynamical systems from measured input/output data using statistical methods [148]. System identification is an alternative to first-principles modeling of complex systems, when the latter approach is too involved or the complexity of a first-principles model makes its use prohibitively expensive (e.g., for real-time control). An identified model is typically developed specifically for a certain application to tradeoff model complexity versus accuracy given the application requirements.

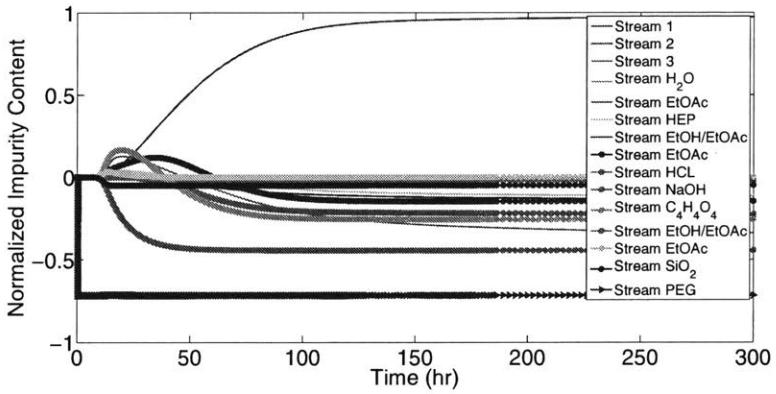
In this study, the subspace identification approach (e.g., see [210]) is used to obtain a low-dimensional description of the plant-wide dynamics in the form of a LTI state-space model

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{5.1a}$$

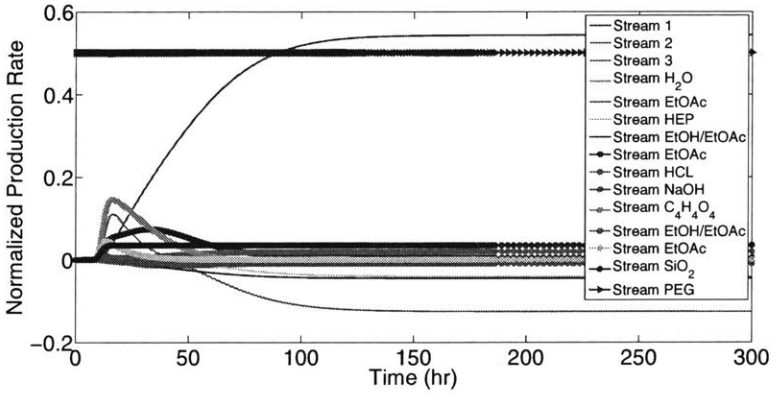
$$y(t) = Cx(t) + Du(t) + n(t) \tag{5.1b}$$



(a) API dosage



(b) Total impurity content



(c) Production rate

Figure 5-3: Dynamic sensitivity analysis of the critical quality attributes with respect to the potential critical process parameters (the streams are ordered as in Figure 5-1).

where A, B, C, and D denote the system state-space matrices, t denotes time, $x \in \mathbb{R}^{n_x}$ denotes the state variables, $\dot{x}$ is the derivative of x with respect to time, $u \in \mathbb{R}^{n_u}$

denotes the system inputs (i.e., CPPs), $y \in \mathbb{R}^{n_y}$ denotes the system outputs (i.e., CQAs), $w \in \mathbb{R}^{n_x}$ denotes the process noise, and $n \in \mathbb{R}^{n_y}$ denotes the measurement noise. In general, subspace identification involves two steps: (i) the model order n_x and a state sequence $\hat{x}$ are determined by projecting row spaces of the input/output data block Hankel matrices, and applying a singular value decomposition; and (ii) a least-squares problem is solved to obtain the state-space matrices [253].

Prior to system identification, a dynamic sensitivity analysis is preformed using the plant simulator to determine the CPPs to which the CQAs are most sensitive. This enables identifying the CPPs with the largest influence on the plant-wide dynamics relevant to control of the ICM pilot plant. Hence, merely the chosen CPPs are used to excite the pilot plant (i.e., plant simulator) to generate sufficiently informative input-output data for identification of a state-space model (5.1). This is to avoid excessively long experimentation times and unnecessary process perturbations during data collection. Figure 5-3 shows the results of the dynamic sensitivity analysis for the different CQAs. The simulation results indicate that the flow rates of the Streams 2, 3, H_2O, and NaOH in the pilot plant (see Figure 5-1) have a negligible dynamical effect on the CQAs and, therefore, are not useful as effective CPPs. On the other hand, even though the flow rates of the Streams HEP and $C_4H_4O_4$ influence the CQA profiles somewhat largely, these CPPs cannot be utilized for plant-wide control due to practical considerations pertaining to the process operation (see [136]).

Table 5.1: The effective CPPs used for plant-wide identification and control (the streams are ordered as in Figure 5-1).

Flow rate of Stream 1 (compound 1 in Figure 5-2)
Flow rate of Stream EtOAc
Flow rate of Stream EtOH/EtOAc
Flow rate of Stream EtOAc
Flow rate of Stream HCl
Flow rate of Stream EtOH/EtOAc
Flow rate of Stream EtOAc
Flow rate of Stream SiO_2
Flow rate of Stream PEG

The CPPs that can be used to effectively regulate the CQAs are listed in Table 5.1. These CPPs, along with purity of Stream 1 that is considered as a measured disturbance, are excited in a multistep fashion to generate input/output data for identification of the plant-wide dynamics. The canonical variate analysis (CVA) subspace identification method [138] is used to identify a LTI state-space model of order 12 (i.e., $n_x = 12$ in (5.1)). The subspace identification is performed using the MATLAB function `n4sid`. The predictions of the identified model validated against an independent data set are depicted in Figure 5-4. The model validation results suggest that the identified low-dimensional model provides an adequate description of the steady-state process behavior.

Linearization of the Plant-wide Model

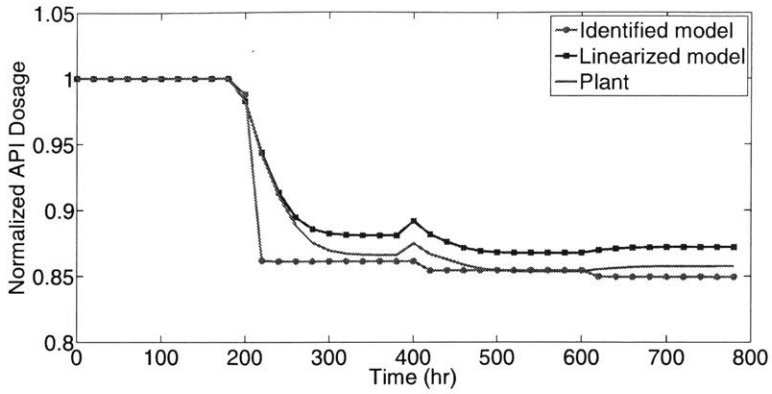
The plant simulator describes the plant-wide dynamics by a set of nonlinear, continuous-time differential algebraic equations

$$0 = F(\dot{z}(t), z(t), v(t), u(t), \theta), \quad z(0) = z_0 \tag{5.2}$$

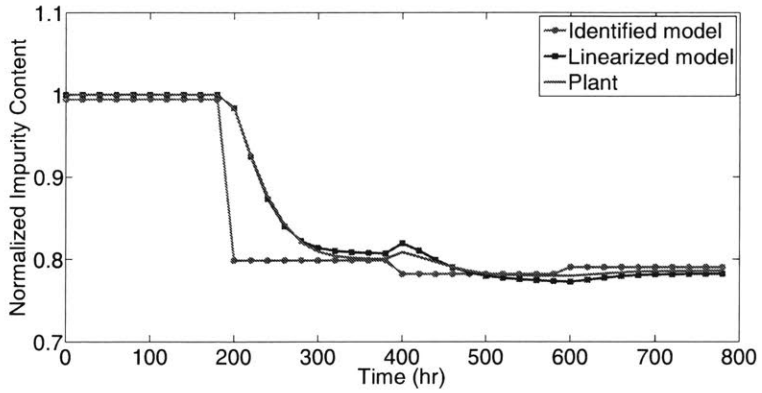
where $z \in \mathbb{R}^{n_z}$ denotes the differential state variables, $\dot{z}$ denotes the derivative of z with respect to time, $z(0) = z_0$ denotes *consistent* initial conditions, $v \in \mathbb{R}^{n_v}$ denotes the algebraic state variables, $\theta \in \mathbb{R}^{n_\theta}$ denotes the system parameters, and $F : \mathbb{R}^{2n_z+n_v} \rightarrow \mathbb{R}^{n_z+n_v}$ denotes the $n_z + n_v$ equations describing the nonlinear system dynamics. The system state vector is comprised of the differential and algebraic state variables denoted by $x = \left[z^T \; v^T\right]^T \in \mathbb{R}^{n_x}$ with $n_x = n_z + n_v$. The system outputs (i.e., $y \in \mathbb{R}^{n_y}$) are algebraically related to x, u, and θ. Hence, y can be included in the definition of x in (5.2) as algebraic state variables for notational simplicity.

Taylor series expansion can be used to obtain a LTI state-space approximation of (5.2) around a steady-state operating point (e.g., see [19])

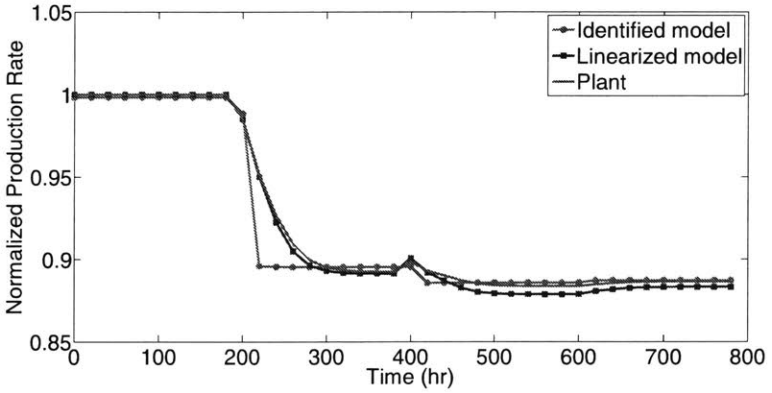
$$M\dot{x}(t) = A\big(x(t) - x_{ss}\big) + B\big(u(t) - u_{ss}\big) \tag{5.3}$$



(a) API dosage



(b) Total impurity content



(c) Production rate

Figure 5-4: Validation of the identified and linearized models with respect to an independent data set generated using the plant simulator.

with the subscript "ss" being the steady-state solution (i.e., $x(\infty) = x_{ss}$ that satisfies

(5.2) given $u(\infty) = u_{ss}$, $\dot{z}(\infty) = 0$, and θ). The system matrices are defined by

$$M = \left[\left. \frac{\partial F}{\partial \dot{z}} \right|_{ss} \quad \mathbf{0}_{n_x \times n_v} \right] \tag{5.4a}$$

$$A = -\left[\left. \frac{\partial F}{\partial z} \right|_{ss} \quad \left. \frac{\partial F}{\partial v} \right|_{ss} \right] \tag{5.4b}$$

$$B = -\left. \frac{\partial F}{\partial u} \right|_{ss} \tag{5.4c}$$

where $\left. \frac{\partial F}{\partial \alpha} \right|_{ss}$ denotes the Jacobian of the function F with respect to the variable α evaluated at the steady state such that the ij^{th} element is the partial derivative of F_i with respect to α_j, and $\mathbf{0}_{n_x \times n_v}$ denotes a zero matrix of size n_x by n_v. The state-space model (5.3) is high dimensional with 7,613 state variables ($n_z = 6,087$ and $n_v = 1,526$). The system inputs consists of the CPPs listed in Table 5.1 ($n_u = 9$).

The system matrices (5.4) are derived efficiently using the automatic differentiation feature of DAEPACK [249], which takes a Fortran-based system model as input and generates the Jacobian matrices of the model. A Fortran version of (5.2) was generated using a code generation patch to the JACOBIAN simulation platform, in which the plant simulator is implemented. The consistent initial conditions for the high-dimensional state-space model are obtained by solving the sparse linear set of equations in (5.3) at $t = 0$

$$\underbrace{\left[\left. \frac{\partial F}{\partial v} \right|_{ss} \left. \frac{\partial F}{\partial \dot{z}} \right|_{ss} \right]}_{A_{initial}} \begin{bmatrix} \widetilde{v}_0 \\ \dot{z}_0 \end{bmatrix} = \underbrace{B\widetilde{u}_0 - \left. \frac{\partial F}{\partial z} \right|_{ss} \widetilde{z}_0}_{b_{initial}} \tag{5.5}$$

where $\widetilde{x}$ denotes the deviation variable form of x (i.e., $\widetilde{x} := x - x_{ss}$) and x_0 denotes the initial condition of x (i.e., $x_0 := x(t = 0)$). For the given initial inputs $\widetilde{u}_0$ and set of differential state values $\widetilde{z}_0$, (5.5) is solved to determine the remaining unknowns $\widetilde{v}_0$ and $\dot{z}_0$. In this study, the stiff, ordinary differential equation solver `ode15s` in MATLAB is used to solve (5.3) with the consistent initial conditions given by (5.5).

The predictions of the state-space model (5.3) are shown in Figure 5-4. The high-dimensional model can adequately describe the predictions of the nonlinear plant

simulator. The simulation results indicate that the high-dimensional model outperforms the identified low-dimensional model in describing the transient dynamics from one steady-state operating point to another. This arises from the first-principles nature of the linearized model, which leads to more accurate representation of process dynamics at the expense of computational complexity due to the significantly larger state dimension.

5.4 Application to Integrated Continuous Pharmaceutical Pilot Plant

In this section, we explore how well QDMC can reject realistic process uncertainties and disturbances within the integrated continuous pharmaceutical manufacturing pilot plant. Two plant-wide QDMC systems are designed using the low-dimensional identified model (5.1) and the high-dimensional linearized model (5.3) to obtain finite step response models (labeled SS-MPC and LM-MPC below, respectively). Closed-loop simulation results are obtained by applying the optimal control inputs to the nonlinear plant simulator. The CQAs are assumed to be measured and sampled every 5 minutes, which are fed back to the control systems to update the model predictions. The performance of these plant-wide QDMC systems is compared to that of a plant-wide regulatory control system that consists of multi-loop proportional-integral controllers (see [136]). These plant-wide control systems are mounted on top of a stabilizing control layer to ensure stable operation. These results are also compared to an open-loop case, in which only the stabilizing control layer is applied to the plant. In what follows, all CQA plots are normalized with respect to desired steady-states. Note that a small amount of random i.i.d measurement noise was included in all simulations using LM-MPC whereas it was not considered in simulations with SS-MPC.

Control Problem Formulation

As listed in Table 5.1, the process has nine effective CPPs that can be used as manipulated variables. The control objective is to regulate the API dosage and production rate of the manufactured tablets to follow desired setpoint trajectories in the presence of process uncertainties and disturbances, while ensuring that the total impurity content of the tablets remains below an admissible threshold.

The QDMC algorithm presented in Section 4.7 is adopted in this work. The outputs y are defined as

$$y \leftarrow [\text{API},\ \text{PR},\ \text{IMP}]^\top$$

where API is the concentration of the active pharmaceutical ingredient in the produced tablets, PR is the production rate of tablets, and IMP is the impurity content of the tablets. Box constraints are considered on each of the input values such that the inputs can vary within 20% of their steady-state values, i.e., $0.8u_{ss} \leq u(k) \leq 1.2u_{ss}$. One-sided box constraints are also considered for the impurity content of the tablets, which cannot increase more than 8% of its steady-state value, i.e., $\text{IMP}(k) \leq 1.08\text{IMP}_{ss}$. These output constraints are enforced over the entire prediction horizon of the QDMC problem (can be thought of as state constraints as shown in Section 4.7).

The output weight matrix Q is selected to be a diagonal matrix with a large value for (1,1) element since API dosage is the primary CQA of the manufactured tablets. The impurity content in the tablets must always meet the constraint limit or many days worth of produced tablets must be discarded due to current regulations. Note that the QDMC framework is very flexible as it also allows for QbD constraints to be directly included in the optimization to ensure regulatory compliant operation.

The plant-wide QDMC system is implemented in a receding-horizon mode, as illustrated in Figure 5-5. This requires online solution of a optimal control problem of the form (4.5) over the horizon at every time instant. The online measurements of CQAs are used to continuously update the prediction model at each sampling time in-

stant. The receding-horizon implementation of the control system partly circumvents performance degradation of the optimal control inputs due to model imperfections and process disturbances.

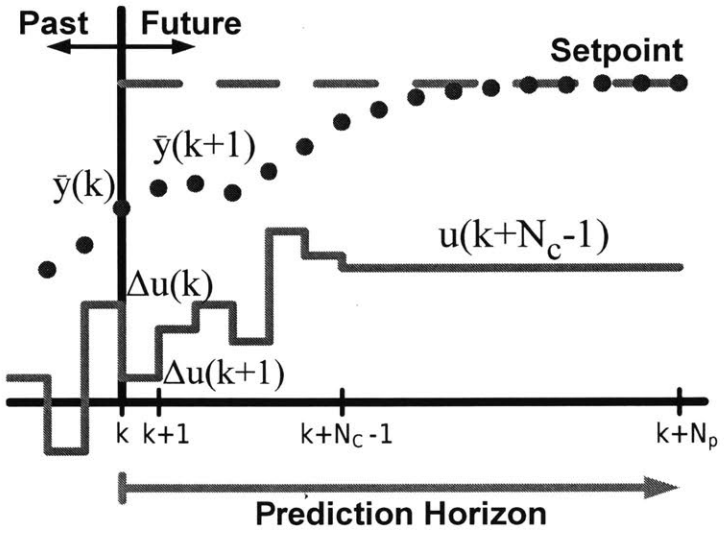


Figure 5-5: Illustration of the receding-horizon implementation of the plant-wide QDMC system.

Parametric Uncertainties in Reaction Kinetics

To investigate the effect of parametric uncertainties on plant-wide control of the ICM pilot plant, a gradual change in synthesis of the intermediate and API compounds in reactors **R1** and **R2** (see Figure 5-1) is induced by defining the reaction kinetics as

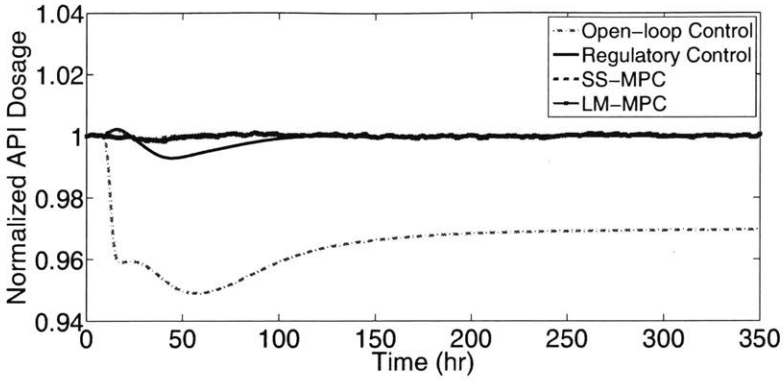
$$\frac{k_{R1,1}(t)}{k^0_{R1,1}} = \begin{cases} 1 - 0.005t & \text{if } t < 20 \\ 0.90 & \text{if } t \geq 20 \end{cases} \quad \Bigg| \quad \frac{k_{R2,2}(t)}{k^0_{R2,2}} = \begin{cases} 1 + 0.01t & \text{if } t < 100 \\ 2.0 & \text{if } t \geq 100 \end{cases}$$

where $k_{R1,1}$ and $k_{R2,2}$ denote the rate constants for the intermediate and API compound synthesis reactions, respectively (see Tables 1 and 2 in [18]). $k^0_{R1,1}$ and $k^0_{R2,2}$ denote the nominal values of the rate constants.

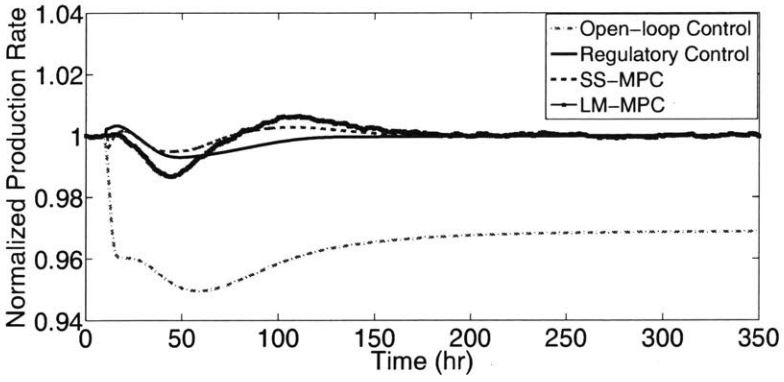
Figure 5-6 depicts the closed-loop simulation results for the plant-wide QDMC and regulatory control systems, as well the open-loop case where only the stabilizing

control layer is used for level control. As can be seen, in the open-loop case with no active control the ICM pilot plant cannot be retained at the desired steady-state operating condition in the presence of process uncertainties (i.e., the normalized API dosage and production rate profiles do not remain at 1.0 in Figures 5-6a and 5-6b, respectively). In addition, the impurity level violates its maximum admissible level in the open-loop case, leading to production of off-spec tablets. This calls for active control of the ICM pilot plant to be able to fulfill the stringent regulatory requirements for the CQAs of the manufactured tablets.

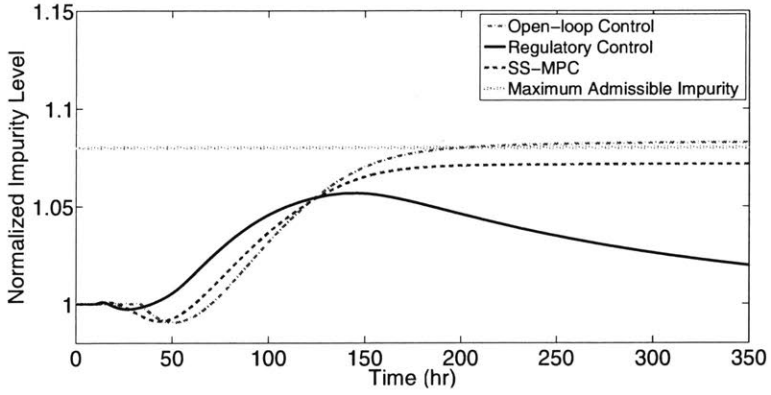
The simulation results indicate the ability of plant-wide MPC in effectively dealing with process uncertainties, as the API dosage and production rate profiles can follow the setpoint trajectories closely. Figure 5-6a shows that both plant-wide QDMC systems with the underlying low-dimensional identified model (SS-MPC) and the high-dimensional linearized model (LM-MPC) outperform the regulatory control system in terms of maintaining the desired API dosage specification. This is due to the ability of MPC to deal with multivariable dynamics of the ICM pilot plant, while multiloop PID controllers cannot accomplish this systematically. Even though the improvement gained in the API dosage setpoint tracking is approximately 1% using the plant-wide QDMC systems as compared to the plant-wide regulatory control system, this is paramount in pharamaceutical manufacuring since the CQAs of the tablets cannot be compromised. As shown in Figure 5-6b, SS-MPC outperforms the LM-MPC in terms of the production rate setpoint tracking, which can be attributed to controller tuning. Figure 5-6c clearly shows the ability of plant-wide MPC in constarint handling (i.e., circumventing violation of the maximum admissible impurity level). Note that the plant-wide regulatary control system maintains the impurity level at some pre-specified value using a PI controller (see [136]) and, therefore, the impurity level profile remains at 1.0. However, by explicitly incorporating the CQA constarints into the control framework, plant-wide MPC systems are able to systematicaly exploit the extra degrees of freedom of the process to more effectively achieve the regulatory requirements for the other CQAs (e.g., the API dosage in this scenario).



(a) API dosage



(b) Production rate



(c) Total impurity content

Figure 5-6: Closed-loop control of the ICM pilot plant with the plant-wide MPC and regulatory control systems in the presence of parametric uncertainties in the intermediate and API synthesis reaction kinetics (reactors **R1** and **R2** in Figure 5-1).

Persistent Disturbance in Filtration Units

The effect of persistent disturbances on plant-wide control of the ICM pilot plant is investigated by persistently decreasing the washing efficiency in the filtration units

W1 and **W2** (see Figure 5-1) as

$$\frac{K_{W,1}(t)}{K^0_{W,1}}, \frac{K_{W,2}(t)}{K^0_{W,2}} = \begin{cases} \exp(-0.002t) & \text{if } t < 200 \\ 0.70 & \text{if } t \geq 200 \end{cases}$$

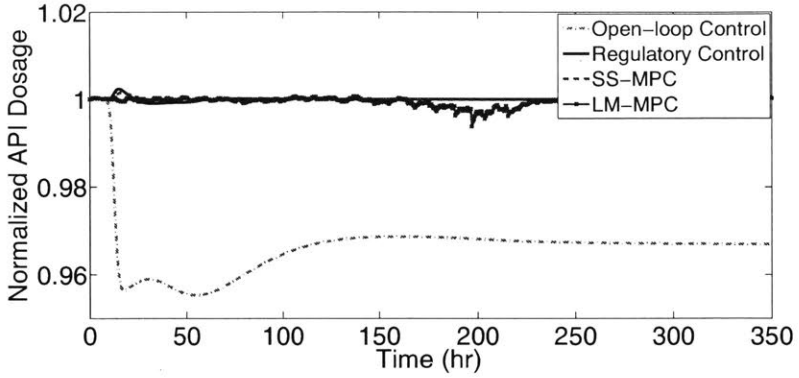
where $K^0_{W,1}$ and $K^0_{W,2}$ are the nominal values for the wash factor in the filtration units **W1** and **W2**, respectively (see [18]). The closed-loop simulation results are depicted in Figure 5-7. It is evident that the plant-wide QDMC systems (especially SS-MPC) lead to effective tracking of the setpoint trajectories and, therefore, enable operating the ICM pilot plant around the desired steady-state operating point effectively in the presence of persistent disturbances. The plant-wide QDMC system with the low-dimensional identified model (SS-MPC) slightly outperforms the plant-wide regulatory control system in terms of achieving smoother setpoint tracking. Yet again, open-loop control with the stabilizing control layer results in a substantial degradation of the CQAs of tablets due to significant and persistent process disturbances.

Temporary Disturbance in Purity Level of Intermediate Compound

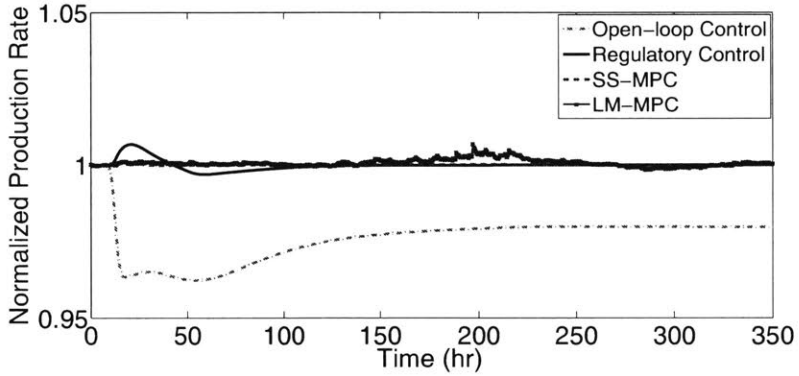
The purity level of the intermediate compound **1** (stream **1** in Figure 5-1), denoted by $x_1(t)$, is changed, i.e.,

$$x_1(t) = \begin{cases} 0.975 & \text{if } 50 \leq t \leq 200 \\ 0.99 & \text{otherwise,} \end{cases}$$

to investigate the ability of the plant-wide control systems to deal with temporary disturbances. The closed-loop simulation results suggest that the plant-wide QDMC systems outperform the plant-wide regulatory control system, in particular for the API dosage setpoint tracking (see Figure 5-8). Both SS-MPC and LM-MPC enable maintaining the API dosage tightly at the desired setpoint in the presence of feed impurities. This is crucial for industrial-scale pharmaceutical manufacturing to have



(a) API dosage



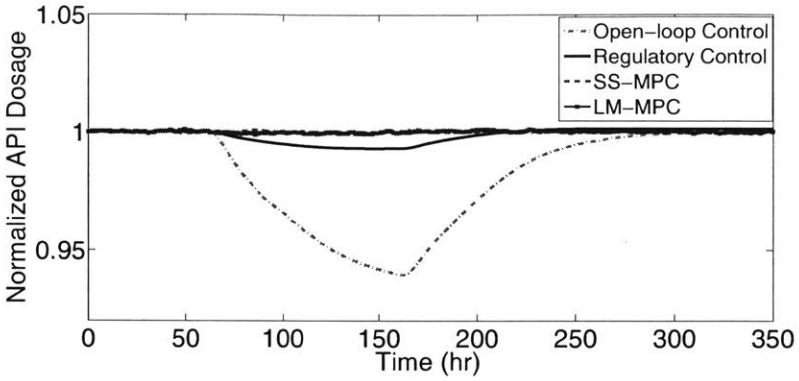
(b) Production rate

Figure 5-7: Closed-loop control of the ICM pilot plant using the plant-wide MPC and regulatory control systems in the presence of persistence disturbance in the filtration units (filter units **W1** and **W2** in Figure 5-1).

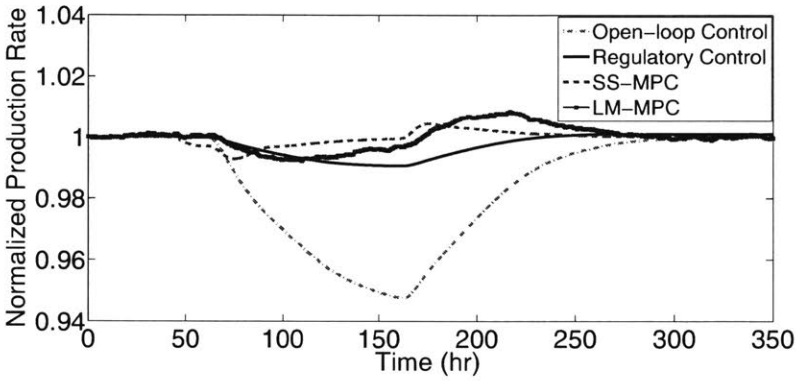
robust fulfillment of the regulatory requirements for CQAs in terms of coping with process changes and disturbances.

Step Increase in Production Rate

The ability of the plant-wide control systems in dealing with setpoint changes is evaluated by changing the production rate setpoint for 5% (see Figure 5-9b). As can be seen in Figure 5-9a, the plant-wide QDMC systems enable tracking the API doage setpoint trajectory more closely than the plant-wide regulatory control system. In addition, the plant-wide QDMC systems lead to faster transition dynamics to the new setpoint, as shown in Figure 5-9b. Hence, plant-wide MPC results in production of



(a) API dosage



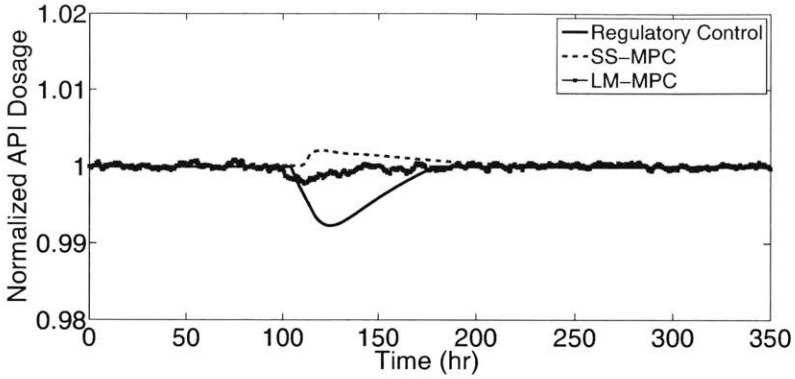
(b) Production rate

Figure 5-8: Closed-loop control of the ICM pilot plant using the plant-wide MPC and regulatory control systems in the presence of temporary disturbance in the purity level of the intermediate compound **1** (stream **1** in Figure 5-1).

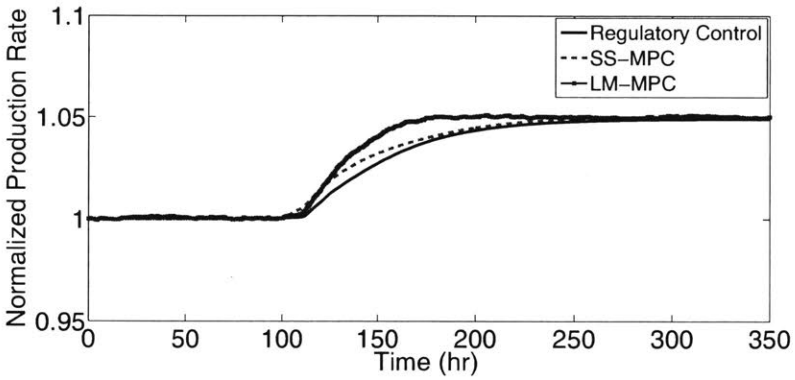
less off-spec tablets during the production rate setpoint change and, therefore, offers more flexibility in operation of the ICM pilot plant.

5.5 Conclusions

This chapter presents two plant-wide MPC systems for an end-to-end continuous pharmaceutical manufacturing pilot plant built at the Novartis-MIT Center for Continuous Manufacturing. Two modeling approaches (subspace identification and linearization of the nonlinear DAE plant-wide model) are investigated to obtain a linear low-dimensional and high-dimensional representation of the nonlinear plant dynam-



(a) API dosage



(b) Production rate

Figure 5-9: Dynamic response of the ICM pilot plant to a step increase in the production rate. The pilot plant is in closed-loop operation with the plant-wide MPC and regulatory control systems.

ics, respectively. The quadratic dynamic matrix control algorithm is applied to design an input-output control framework for plant-wide MPC of the integrated continuous manufacturing pilot plant, independent of the plant state dimension.

The closed-loop performance of the two plant-wide MPC systems designed using the low-dimensional and high-dimensional plant models is evaluated for various scenarios pertaining to process uncertainties, disturbances, and setpoint changes. The simulation results indicate that plant-wide MPC facilitates effective regulation of CQAs and flexible process operation in the presence of process uncertainties and disturbances of different nature. In addition, plant-wide MPC enables incorporating the QbD considerations into the control problem through input and output constraints

to ensure regulatory compliant process operation. This is paramount for meeting the stringent regulatory requirements in the pharmaceutical industry.

Part IV

Stochastic Model Predictive Control

Chapter 6

Fast Model Predictive Control of High-dimensional Systems with Probabilistic Uncertainty

6.1 Introduction

Model predictive control (MPC) is the most widely used approach for the advanced control of complex dynamical systems due to its ability to systematically deal with multivariable dynamics, system constraints, and competing sets of objectives [174]. Chapter 4 provides a detailed discussion on the current state-of-the-art methods in deterministic MPC. While many MPC formulations have been developed to account for exogenous disturbances and measurement noise, few formulations are able to systematically handle probabilistic uncertain parameters, in spite of their ubiquity in complex systems. Probabilistic uncertainties can lead to severe closed-loop performance degradation and, as a result, impair high-performance operation of complex systems using classical MPC approaches.

Robust MPC is a popular approach for dealing with model uncertainties (e.g., see [14]). Assuming that uncertainties are bounded, most robust MPC approaches compute the optimal control law to minimize the performance for the worst-case

model uncertainties. While this approach can guarantee constraint satisfaction under all disturbances and uncertainties, the resulting control law is conservative in most control applications, as the worst-case perturbations usually have a very low probability of occurrence. Recently, novel methods for robust MPC have been investigated using the so-called *scenario approach* (e.g., see [38, 278], and the references therein). The scenario approach provides a sampling-based technique to solve convex chance-constrained optimization problems and, therefore, enables a paradigm shift from deterministic algorithms to randomized robust MPC approaches that exploit the statistical description of uncertainties. However, typically the number of samples (i.e., scenarios) results in an on-line computational cost that is too high for implementation in high-dimensional systems.

Stochastic MPC (SMPC) offers an alternative approach for robust MPC in a probabilistic uncertainty setting. SMPC approaches enable shaping the predicted probability distribution functions (PDFs) of system states and outputs in an optimal manner over a finite prediction horizon (e.g., see [27, 41, 125, 110, 169], and the references therein). In SMPC, chance constraints can be considered in a probabilistic sense to circumvent the inherent conservatism of deterministic worst-case robust MPC approaches. However, a key challenge in SMPC is the propagation of probabilistic uncertainties through the system model. The commonly used approaches for probabilistic uncertainty analysis (e.g., Monte Carlo methods [255]) are prohibitively expensive for real-time control for high-dimensional systems.

This work presents a fast stochastic MPC algorithm applicable to high-dimensional stable systems with time-invariant probabilistic uncertainties in initial conditions and system parameters. The quadratic dynamic matrix control (QDMC) algorithm [76], which is the MPC algorithm most widely applied to large-scale industrial systems, is adopted to formulate an input-output framework for SMPC with output constraints. Such a probabilistic input-output framework has an online computational cost that is independent of the state dimension, which enables its application to uncertain systems with high state dimension[1].

[1]Such systems are referred to as *large-scale* or *high-dimensional systems* in this chapter.

Generalized polynomial chaos theory [266, 272] is used for propagation of probabilistic uncertainties through the high-dimensional system model. In polynomial chaos theory, the implicit mappings between uncertain variables/parameters and the system states are replaced with expansions of orthogonal polynomials that are functions of the random variables (see Chapter 3 for further details). In contrast to sampling-based uncertainty analysis approaches, the orthogonality property of polynomial chaos expansions (PCEs) enables efficient computation of the statistical properties of the state PDFs. Galerkin projection is ued to determine the coefficients of the PCEs for a general class of linear differential algebraic equations (DAEs). As such, the proposed fast SMPC approach can be applied to a large class of complex systems, whose dynamics are described by a set of high-dimensional DAEs.

Organization The next section illustrates that parameter uncertainty can lead to instability in even the unconstrained version of the standard MPC problem (meaning that they do not provide guaranteed robustness properties). This explicitly motivates the need to account for parameter uncertainty within MPC. Section 6.3 formulates the stochastic MPC problem for high-dimensional systems with probabilistic uncertainties. Section 6.4 presents the Galerkin projection method and briefly extends it to linear dynamical systems described by index-1 DAEs. This is combined with the quadratic dynamic matrix control (QDMC) algorithm in Section 6.5 in order to write the MPC problem in terms of the system inputs and outputs, such that the online computational cost is independent of the large state dimension. Then, the proposed approach for fast MPC of large-scale systems with probabilistic uncertainties is applied to an end-to-end continuous pharmaceutical manufacturing pilot plant in Section 6.6 (see Chapter 5 for results for this process with standard QDMC under various disturbance case studies) and conclusions are drawn in Section 6.7.

The majority of this chapter was published in the *Proceedings of the IEEE Conference on Decision and Control* [199].

6.2 Can Parameter Uncertainty Lead to Instability in MPC?

Before developing a method to account for parametric uncertainty in MPC, it is useful to consider how badly performance can be effected by model uncertainty. To do this, we explored the application of linear-quadratic-Gaussian (LQG) control (which is one of the fundamental optimal control problems that was solved in 1960s) to a simple system. LQG control combines the Kalman filter [120] (optimal estimator for linear systems subject to i.i.d. Gaussian white noise) and an unconstrained linear-quadratic regulator (unconstrained version of (4.5) with $l(x_j, u_j) = x_j^\top Q x_j + u_j^\top R u_j$ and initial state x equal to the estimated state from the Kalman filter). The solution to this problem is the best possible controller when the model is known exactly.

However, a well-known example problem, presented in [61], illustrates that the closed-loop system may have an arbitrarily small gain margin when using the LQG controller, which means that the system can go unstable whenever the system gain is slightly different than the assumed gain by an arbitrarily small amount in either direction. The simple two-state system is stated as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + m \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w, \tag{6.1}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v, \tag{6.2}$$

where u is the control action, y is the output of interest, w is the process noise (with variance σ^2), v is the measurement noise (with variance 1), and m is the scalar system gain matrix (nominally equal to 1). We select the state penalty matrix to be $Q = q \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and the input penalty matrix to be $R = 1$. Letting $\hat{x}_1$ and $\hat{x}_2$ denote the state estimates of x_1 and x_2, respectively, the overall closed-loop system can be

derived to be

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & -mf & -mf \\ d & 0 & 1-d & 1 \\ d & 0 & -d-f & 1-f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & d \\ 0 & d \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix}, \tag{6.3}$$

where $f = 2+\sqrt{4+q}$ and $d = 2+\sqrt{4+\sigma^2}$. As shown in [61], the stability margins can be made arbitrarily small in either direction by appropriate choice of q and σ^2. The system simulated for the nominal gain $m = 1$ observed good performance. However, when the gain m is slightly above or below this nominal value, the system response can go unstable even though an optimal estimator and controller is applied (Figure 6-1).

The main point of this example is that optimal-performance LQG solutions do not provide any guaranteed robustness properties. Being able to explicitly account for system uncertainty in the design stage is important for building in these margins to ensure safe operation and guaranteed performance, which is true even for simple two-state systems and becomes more important as system complexity increases.

6.3 Problem Formulation

Consider an uncertain continuous-time stable linear differential-algebraic system,

$$M(\theta)\dot{x}(t,\theta) = A(\theta)x(t,\theta) + B(\theta)u(t) + r(\theta), \quad x(0,\theta) = x_0(\theta), \tag{6.4a}$$

$$y(t,\theta) = C(\theta)x(t,\theta), \tag{6.4b}$$

where $x \in \mathbb{R}^{n_x}$ denotes the system states, $\dot{x}$ denotes the derivative of x with respect to time t, x_0 denotes the initial conditions, $u \in \mathbb{R}^{n_u}$ denotes the inputs to the system, $y \in \mathbb{R}^{n_y}$ denotes the outputs of the system, and $\theta \in \mathbb{R}^{n_\theta}$ denotes the system random variables (i.e., uncertain parameters and initial conditions). Eq. (6.4a) is a differential algebraic equation (DAE) and is assumed to be in its equivalent index-1 form (e.g.,

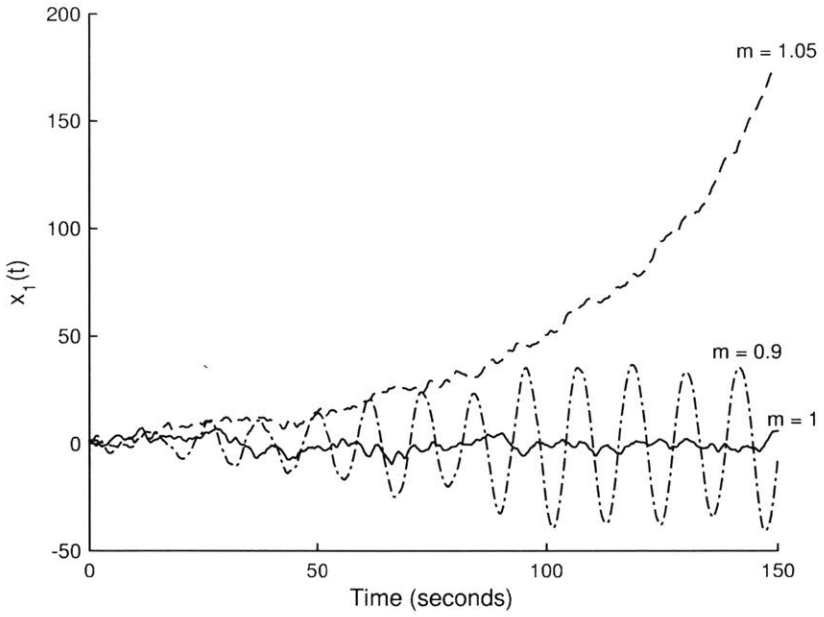


Figure 6-1: Illustration that LQG optimal controller does not provide guaranteed stability margins when there is process gain uncertainty. The output of interest is $y(t) = x_1(t)$.

see [78]). The state vector $x = [x_d^T \; x_a^T]^T$ is composed of differential states $x_d \in \mathbb{R}^{n_d}$ (i.e., states whose derivatives appear in the vector $M(\theta)\dot{x}(t,\theta)$ in (6.4a)) and algebraic states $x_a \in \mathbb{R}^{n_a}$ (i.e., the remaining states) with $n_x = n_d + n_a$. For a consistent initial condition in (6.4a), $n_x + n_d$ variables must be specified ($x(0,\theta)$ and $\dot{x}_d(0,\theta)$), with n_x degrees of freedom (DOF) set by (6.4a) holding at $t = 0$ and n_d additional DOF. The system outputs are assumed to be linearly related to the states as shown in (6.4b). The vector $r \in \mathbb{R}^{n_x}$ is used to represent time-invariant stochastic behavior such as process disturbances and actuator response variations due to, for example, static friction and backlash. Note that r can also be used to represent additional terms or error from linearization of nonlinear dynamics.

The system (6.4a) is often called a *descriptor* or *singular* system in the control literature (e.g., [31, 71] and references cited therein), with systems described by ordinary differential equations being a special case. The extra algebraic terms enable the representation of a much broader classes of systems, including pH neutralization, electrochemical systems, and some classes of electronic and mechanical systems.

In (6.4), θ is composed of independently distributed random variables θ_i with known probability distribution functions (PDFs) f_{θ_i}. A probability triple $(\Omega, \mathcal{F}, \mathcal{P})$ is defined on the basis of sample space Ω, σ-algebra $\mathcal{F}$, and probability measure $\mathcal{P}$ on $(\Omega, \mathcal{F})$. The PDFs are defined such that $\theta_i \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, $\forall i \in \{1, \ldots, n_\theta\}$, where $\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ represents the Hilbert space of all random variables θ_i with finite L_2 norm. The expected value (first-order moment) of a stochastic variable $\psi : \Omega \mapsto \mathbb{R}$ is denoted by $\mathbb{E}[\psi] := \int_\Omega \psi df_\psi$, where f_ψ is the PDF of ψ over its support Ω. The variance (central second-order moment) of ψ is denoted by $\mathbf{Var}[\psi] := \mathbb{E}\big[(\psi - \mathbb{E}[\psi])^2\big]$.

Due to the probabilistic uncertainties θ, the solution trajectories of system (6.4) are probabilistically distributed. In this work, the goal of controller synthesis is to shape the probability distributions of system outputs to have desirable statistics. Assuming that the system states can be estimated at all times, a finite-horizon SMPC problem can be stated as follows.

Problem 6.1 (Finite-horizon Stochastic MPC)

$$
\begin{aligned}
\min_{u(t)} \quad & J(\hat{x}_d(t_k), u(t)), && && (6.5)\\
s.t.: \quad & M(\theta)\dot{\bar{x}}(t,\theta) = A(\theta)\bar{x}(t,\theta) + B(\theta)u(t) + r(\theta), && t_k \le t \le t_p,\\
& \bar{y}(t,\theta) = C(\theta)x(t,\theta), && t_k \le t \le t_p,\\
& A_h \mathbb{E}[\bar{y}(t,\theta)] \le b_h, && t_k \le t \le t_p,\\
& u(t) \in \mathbb{U}, && t_k \le t \le t_m,\\
& \bar{x}_d(0,\theta) = \hat{x}_d(t_k),
\end{aligned}
$$

where $u(t)$, $t \in [t_k, t_m]$, denotes the input profile (control policy), t_m denotes the control horizon, t_p denotes the prediction horizon, $\bar{x}(t,\theta)$ denotes the predicted states from the DAE system model in (6.4a), $\bar{y}(t,\theta)$ denotes the predicted outputs, $\hat{x}_d(k)$ denotes the estimated differential states at time instant t_k computed from measurements, (A_h, b_h) specifies the linear constraints on the expected value of the outputs, and $\mathbb{U} \in \mathbb{R}^{n_u}$ denotes the convex compact set of input constraints.

Problem 6.1 specifies an optimal input profile based on the uncertain DAE model

(6.4a) and the estimated state variables $\hat{x}_d(t_k)$, while satisfying hard constraints on the inputs and general inequality constraints on the outputs. The cost function $J(\hat{x}_d(k), u(t))$ is commonly defined in terms of some statistics or moments of the PDFs of the outputs (see, for example, [169]). Note that a unique consistent initialization of (6.4a) can be computed from $\bar{x}_d(0, \theta)$.

Solving Problem 6.1 for high-dimensional systems is particularly challenging. The difficulties arise from (i) the prohibitive computational costs of model simulation and optimization for on-line control due to a large state dimension, (ii) the need for high-dimensional state estimation to determine $\hat{x}_d(k)$ from limited system measurements (even more difficult since the system is most likely not observable), and (iii) the propagation of probabilistic uncertainties θ through the system model (6.4).

To address challenges (i) and (ii), the QDMC algorithm is used in this work to reformulate the high-dimensional control problem in terms of system inputs and outputs, which are nearly always of much lower dimensions than the system states in real applications. The input-output framework of QDMC not only eliminates the high computational costs associated with on-line control of high-dimensional systems, but also alleviates the need for high-dimensional state estimation since the output measurements can be readily incorporated into the control algorithm to update the system model. To enable efficient uncertainty analysis, polynomial chaos expansions (PCEs) are used to propagate the uncertainties θ through the dynamics in (6.4). PCEs are advantageous as they can be written directly in terms of system inputs and outputs, and enable efficient computation of the statistical moments of outputs.

6.4 Galerkin Projection for DAE Systems

Polynomial Chaos Expansions

As discussed in detail in Chapter 3, polynomial chaos expansions provide a means for approximating a stochastic variable $\psi(\theta) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ with the L_2-convergent

expansion [272]

$$\psi(\theta) = \sum_{k=0}^{\infty} a_k \Phi_k(\theta), \tag{6.6}$$

where a_k denotes the expansion coefficients and $\Phi_k(\theta)$ denotes the polynomial chaos basis functions of degree m with respect to the random variables θ. These basis functions belong to the Askey scheme of polynomials, which encompasses a set of orthogonal basis functions in the Hilbert space defined by the support of the random variables. Hence, the basis functions satisfy $\langle \Phi_i(\theta), \Phi_j(\theta) \rangle = \langle \Phi_i^2(\theta) \rangle \delta_{ij}$ where $\langle h(\theta), g(\theta) \rangle = \int_\Omega h(\theta) g(\theta) f_\theta d\theta = \mathbb{E}[h(\theta) g(\theta)]$ denotes the inner product with respect to the weight f_θ (PDF of θ) and over the domain Ω (support of θ). The choice of orthogonal polynomials is made such that their weight function is the multivariate PDF of θ. Table 3.1 shows the orthogonal polynomials corresponding to particular distributions in θ.

For practical reasons, the PCE (6.6) must be truncated to a finite number of terms. The total number of terms $L + 1 = \frac{(n_\theta + m)!}{n_\theta! m!}$ in the truncated expansion depends on the number of uncertain parameters n_θ and the highest order of the polynomial basis functions m retained in the expansion

$$\hat{\psi}(\theta) := \sum_{k=0}^{L} a_k \Phi_k(\theta) = \mathbf{a}^\top \Lambda(\theta), \tag{6.7}$$

with $\mathbf{a} = [a_0, \ldots, a_L]^\top$ and $\Lambda(\theta) = [\Phi_0(\theta), \ldots, \Phi_L(\theta)]^\top$. The vector of PCE coefficients $\mathbf{a}$ can be computed using probabilistic collocation methods or Galerkin projection, depending on the complexity of system dynamics (e.g., see [272, 168, 248] and references therein). The orthogonality property of the multivariate polynomials can be used to efficiently compute the PDF statistics of the stochastic variable $\hat{\psi}(\theta)$. For instance, the first- and second-order central moments of $\hat{\psi}(\theta)$ are defined by

$$\mathbb{E}[\hat{\psi}(\theta)] = a_0, \tag{6.8}$$

$$\mathbf{Var}[\hat{\psi}(\theta)] = \sum_{k=1}^{L} a_k^2 \langle \Phi_k(\theta)^2 \rangle. \tag{6.9}$$

Galerkin Projection for Index-1 Stochastic DAEs

For linear systems described by a set of ordinary differential equations (ODEs), the Galerkin projection can used to generate a set of deterministic ODEs to determine the PCE coefficients (e.g., [73, 209, 239]). This section outlines how to use Galerkin projection to generate a set of deterministic DAEs for determining the expansion coefficients, which provides an approach to efficiently compute state/output distributions for systems whose dynamics involve uncertain algebraic equations/constraints.

For the uncertain system (6.4), let x_i, y_i, and r_i denote the i^{th} component of x, y, and r, respectively, and M_{ij}, A_{ij}, B_{ij}, and C_{ij} denote the ij^{th} elements (i.e., the i^{th} row and j^{th} column) of matrices M, A, B, and C, respectively. Applying the truncated PCE expression in (6.7) to each element in (6.4a) gives

$$\hat{x}_i(t,\theta) := \sum_{k=0}^{L} x_{i_k}(t)\Phi_k(\theta) = \mathbf{x}_i^\top(t)\Lambda(\theta), \tag{6.10}$$

$$\hat{M}_{ij}(\theta) := \sum_{k=0}^{L} m_{ij_k}\Phi_k(\theta) = \mathbf{m}_{ij}^\top\Lambda(\theta), \tag{6.11}$$

$$\hat{A}_{ij}(\theta) := \sum_{k=0}^{L} a_{ij_k}\Phi_k(\theta) = \mathbf{a}_{ij}^\top\Lambda(\theta), \tag{6.12}$$

$$\hat{B}_{ij}(\theta) := \sum_{k=0}^{L} b_{ij_k}\Phi_k(\theta) = \mathbf{b}_{ij}^\top\Lambda(\theta), \tag{6.13}$$

$$\hat{r}_i(\theta) := \sum_{k=0}^{L} r_{i_k}\Phi_k(\theta) = \mathbf{r}_i^\top\Lambda(\theta), \tag{6.14}$$

$$\hat{C}_{ij}(\theta) := \sum_{k=0}^{L} c_{ij_k}\Phi_k(\theta) = \mathbf{c}_{ij}^\top\Lambda(\theta), \tag{6.15}$$

$$\hat{y}_i(t,\theta) := \sum_{k=0}^{L} y_{i_k}(t)\Phi_k(\theta) = \mathbf{y}_i^\top(t)\Lambda(\theta), \tag{6.16}$$

where $\mathbf{x}_i(t)$, $\mathbf{m}_{ij}$, $\mathbf{a}_{ij}$, $\mathbf{b}_{ij}$, $\mathbf{r}_i$, $\mathbf{c}_{ij}$, $\mathbf{y}_i \in \mathbb{R}^{L+1}$ are defined similarly to $\mathbf{a}$ in (6.7). The elements of M, A, B, r, and C are known *a priori* and, therefore, their PCE

coefficients can be computed from the *normal equations*

$$m_{ij_k} = \frac{\langle M_{ij}, \Phi_k \rangle}{\langle \Phi_k^2 \rangle}, \; a_{ij_k} = \frac{\langle A_{ij}, \Phi_k \rangle}{\langle \Phi_k^2 \rangle}, \; b_{ij_k} = \frac{\langle B_{ij}, \Phi_k \rangle}{\langle \Phi_k^2 \rangle}, \; r_{i_k} = \frac{\langle r_i, \Phi_k \rangle}{\langle \Phi_k^2 \rangle}, \; c_{ij_k} = \frac{\langle C_{ij}, \Phi_k \rangle}{\langle \Phi_k^2 \rangle}.$$

There are a total of $n_x(L+1)$ unknown PCE coefficients for the states $\{x_{i_k}\}_{i=1,\cdots,n_x;k=0,\cdots,L}$ and $n_y(L+1)$ unknown PCE coefficients for the outputs $\{y_{i_k}\}_{i=1,\cdots,n_y;k=0,\cdots,L}$. A *deterministic* DAE for these coefficients can be constructed by

- Creating the polynomial chaos approximation of (6.4) by substituting the PCEs (6.10), (6.11), (6.12), (6.13), and (6.14) into (6.4a) to yield

$$\sum_{j=1}^{n_x}\sum_{k=0}^{L}\sum_{l=0}^{L} m_{ij_k}\dot{x}_{j_l}\Phi_k\Phi_l = \sum_{j=1}^{n_x}\sum_{k=0}^{L}\sum_{l=0}^{L} a_{ij_k}x_{j_l}\Phi_k\Phi_l + \sum_{j=1}^{n_u}\sum_{k=0}^{L} b_{ij_k}u_j\Phi_k + \sum_{k=0}^{L} r_{i_k}\Phi_k, \tag{6.17}$$

  and substituting (6.15) and (6.16) into (6.4b)

$$y_{i_k} = \sum_{j=1}^{d} c_{ij_k}x_{j_l}\Phi_k\Phi_l. \tag{6.18}$$

- Projecting these approximations onto the orthogonal basis functions (i.e., taking the inner product of (6.17) and (6.18) with Φ_m for $i = 1, \cdots, n_x$ and $m = 0, \cdots, L$):

$$\mathbf{M}\dot{\mathbf{X}}(t) = \mathbf{A}\mathbf{X}(t) + \mathbf{B}u(t) + \mathbf{R}, \tag{6.19a}$$

$$\mathbf{Y}(t) = \mathbf{C}\mathbf{X}(t), \tag{6.19b}$$

where $\mathbf{X} = \left[\mathbf{x}_1^\top \; \cdots \; \mathbf{x}_{n_x}^\top\right]^\top$, $\mathbf{Y} = \left[\mathbf{y}_1^\top \; \cdots \; \mathbf{y}_{n_y}^\top\right]^\top$, $\mathbf{R} = \left[(P\mathbf{r}_1)^\top \; \cdots \; (P\mathbf{r}_{n_x})^\top\right]^\top$, and the matrices $\mathbf{M}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are defined by their blocks,

$$\mathbf{M}_{ij} = \sum_{k=0}^{L} m_{ij_k}T_k, \quad \mathbf{A}_{ij} = \sum_{k=0}^{L} a_{ij_k}T_k, \quad \mathbf{B}_{ij} = P\mathbf{b}_{ij}, \quad \mathbf{C}_{ij} = \sum_{k=0}^{L} c_{ij_k}T_k,$$

with the symmetric inner product matrices P and T_k being

$$P = \begin{bmatrix} \langle \Phi_0^2 \rangle & 0 & \cdots & 0 \\ 0 & \langle \Phi_1^2 \rangle & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \langle \Phi_L^2 \rangle \end{bmatrix}, \; T_k = \begin{bmatrix} \langle \Phi_k \Phi_0 \Phi_0 \rangle & \langle \Phi_k \Phi_0 \Phi_1 \rangle & \cdots & \langle \Phi_k \Phi_0 \Phi_L \rangle \\ \langle \Phi_k \Phi_0 \Phi_1 \rangle & \langle \Phi_k \Phi_1 \Phi_1 \rangle & \cdots & \langle \Phi_k \Phi_1 \Phi_L \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \Phi_k \Phi_0 \Phi_L \rangle & \langle \Phi_k \Phi_1 \Phi_L \rangle & \cdots & \langle \Phi_k \Phi_L \Phi_L \rangle \end{bmatrix}.$$

The PCE coefficients for the states (stacked into $\mathbf{X}$) are composed of differential and algebraic states. Denote the PCE coefficients of the differential and algebraic states as $\mathbf{X}_d$ and $\mathbf{X}_a$, respectively. Similar to (6.4a), $\mathbf{X}_a(0)$ and $\dot{\mathbf{X}}_d(0)$ are uniquely determined from $\mathbf{X}_d(0)$ and (6.19a) at $t = 0$ (must be satisfied for consistent initialization). The elements of $\mathbf{X}_d(0)$ can be computed from the known, but possibly uncertain, initial conditions $x_d(0, \theta)$ using

$$x_{i_k}(0) = \frac{\langle x_i(0, \theta), \Phi_k(\theta) \rangle}{\langle \Phi_k(\theta)^2 \rangle}, \quad i = 1, \cdots, n_d; \; k = 0, \cdots, L,$$

where the first n_d elements of x are x_d.

Note that (6.19) is a *deterministic* DAE in terms of the unknown time-varying PCE coefficients of the states and outputs. The PCEs (6.10) and (6.16), along with the solution to (6.19), describes the uncertain states and outputs as an explicit function of time and θ. This description can be used to efficiently compute the evolution of the mean and variance of $\hat{x}(t, \theta)$ and $\hat{y}(t, \theta)$ over time by exploiting the orthogonality property of the multivariate basis functions.

6.5 Fast MPC with Probabilistic Parameter Uncertainty

The QDMC algorithm and its relationship to the more modern state-space MPC was described in detail in Chapter 4 of this thesis. As mentioned there, QDMC has certain major advantages that are particularly prevalent for large-scale systems, mainly that it can significantly reduce the number of states needed to model the

system (while still being exact for linear models of the system) and it avoids the need to design and implement a dedicated state estimation algorithm (which can be quite expensive in addition to the original states of the system possibly not being observable). This section extends the ideas of QDMC to to handle time-invariant probabilistic uncertainty that often occur within first-principles models of processes.

Step Response Modeling of Output PCE Coefficients

The QDMC algorithm utilizes finite step response models, shown in (4.38), to describe the dynamic response of the output due to changes in the input. QDMC uses the following open-loop observer (aka free response)

$$\bar{Y}_k^{\text{PCE}} = M\bar{Y}_{k-1}^{\text{PCE}} + S\Delta u_{k-1}, \tag{6.20}$$

where

$$M = \begin{bmatrix} 0 & I_{n_y(L+1)} & 0 & \cdots & 0 \\ & 0 & I_{n_y(L+1)} & \ddots & \vdots \\ \vdots & & 0 & \ddots & 0 \\ & & & \ddots & I_{n_y(L+1)} \\ 0 & & \cdots & & I_{n_y(L+1)} \end{bmatrix}, \quad S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix}.$$

where $\Delta u_k = u_k - u_{k-1}$ is the change in the input (at discrete time k which corresponds to continuous time $t_k = kT + t_0$ where T is the sampling time and t_0 is the initial time). Here, $\bar{Y}_k^{\text{PCE}}$ represent the so-called *dynamic states* of the system. In this case, we define the dynamic states to correspond to the PCE coefficients of the output defined by (6.19b). Based on the notation introduced by [139] (and that used in Chapter 4), the dynamic states here are defined as

$$\bar{Y}_k^{\text{PCE}} = [\mathbf{Y}_0(t_k)^\top, \mathbf{Y}_1(t_k)^\top, \cdots, \mathbf{Y}_{n-1}(t_k)^\top]^\top, \tag{6.21}$$

where $\mathbf{Y}_j(t_k)$ can be interpreted as the set of output PCE coefficients $\mathbf{Y}$ at time $t_k + Tj$ assuming constant inputs into the future.

Step Response Matrix The matrices $\{S_i\}$ in (6.20) are defined to be

$$S_i = \begin{bmatrix} s_{1,1,i} & s_{1,2,i} & \cdots & s_{1,n_u,i} \\ s_{2,1,i} & s_{2,2,i} & \cdots & s_{2,n_u,i} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n_y(L+1),1,i} & s_{n_y(L+1),2,i} & \cdots & s_{n_y(L+1),n_u,i} \end{bmatrix}, \tag{6.22}$$

for $i = 1, \cdots, n$ where the FSR is truncated whenever the input no longer effects future outputs; truncated at n in (6.20) commonly referred to as the *model length* for which the system should satisfy $S_n \approx S_{n+1} \approx \cdots \approx S_\infty$. The scalar value $s_{i,j,k}$ is the k^{th} step response coefficient of the i^{th} element of $\mathbf{Y}$ resulting from a step in the j^{th} input value. In this case, the step response coefficients can be computed offline by solving (6.19) for steps in each of the input values. Since this Galerkin-projected DAE system is linear, the FSR representation is an extremely accurate reduced-order model. In fact, the FSR representation of (6.19) will be exact for $n = \infty$ (i.e., for any choice of $\epsilon > 0$, there exists a finite $n^\star$ such that the truncation error, defined by some appropriate norm, can be made less than ϵ).

Objective Function

The predicted output values over the prediction horizon p (given current information at discrete time step k) are denoted by

$$\mathscr{Y}_{k+1|k}(\theta) = [\bar{y}(t_k, \theta)^\top, \bar{y}(t_{k+1}, \theta)^\top, \cdots, \bar{y}(t_{k+p}, \theta)^\top]^\top. \tag{6.23}$$

The vector of future control moves over the control horizon $m \leq p$ (after which the input is assumed to remain constant) starting at discrete time step k is denoted by

$$\Delta\mathscr{U}_k = [\Delta u(k)^\top, \Delta u(k+1)^\top, \cdots, \Delta u(k+m-1)^\top]^\top. \tag{6.24}$$

A multi-objective optimization in terms of the moments of the output distribution avoids the conservatism of robust (worst-case) control [180] while shaping the distribution to have a desired PDF [169]. The proposed objective in this work is

$$\begin{aligned} J &= \mathbb{E}\left[(\mathscr{Y}_{k+1|k}(\theta) - \mathscr{R}_{k+1})^\top \mathbf{Q}(\mathscr{Y}_{k+1|k}(\theta) - \mathscr{R}_{k+1})\right] + \Delta\mathscr{U}_k^\top \mathbf{R}\Delta\mathscr{U}_k, \\ &= \mathbb{E}\left[\mathscr{Y}_{k+1|k}(\theta)^\top \mathbf{Q}\mathscr{Y}_{k+1|k}(\theta)\right] - 2\mathscr{R}_{k+1}^\top \mathbf{Q}\mathbb{E}[\mathscr{Y}_{k+1|k}(\theta)] + \mathscr{R}_{k+1}^\top \mathbf{Q}\mathscr{R}_{k+1}, \end{aligned} \tag{6.25}$$

where $\mathscr{R}_{k+1}$ is the output reference trajectory into the future, $\mathbf{Q}$ is a symmetric positive semidefinite output weight matrix, and $\mathbf{R}$ is a positive definite change in input weight matrix. Other choices of objective function can be made, however, this quadratic objective was chosen such that the resulting MPC problem is a convex quadratic program as shown in the following.

Disturbance Update and QP Formulation

We would like to simplify the computation of (6.25) by using PCE to approximate the expectation terms $\mathbb{E}\left[\mathscr{Y}_{k+1|k}(\theta)^\top \mathbf{Q}\mathscr{Y}_{k+1|k}(\theta)\right]$ and $\mathbb{E}[\mathscr{Y}_{k+1|k}(\theta)]$. First, let us write out the prediction equation for the output PCE coefficients (denoted by $\mathscr{Y}^{\text{PCE}}_{k+1|k}$) as a function of the future input moves

$$\mathscr{Y}^{\text{PCE}}_{k+1|k} = M_p \bar{Y}^{\text{PCE}}_k + S_p^m \Delta\mathscr{U}_k + \mathscr{I}^{\text{PCE}}(y(k) - N\bar{Y}^{\text{PCE}}_k), \tag{6.26}$$

where

$$S_p^m = \begin{bmatrix} S_1 & 0 & & 0 \\ S_2 & S_1 & 0 & 0 \\ & & & \\ S_m & S_{m-1} & & S_1 \\ & & & \\ S_p & S_{p-1} & & S_{p-m+1} \end{bmatrix},$$

M_p is the first $pn_y(L+1)$ rows of the matrix M defined above, $\mathscr{I}^{\text{PCE}} = \mathscr{I} \otimes e_{L+1}$, $\mathscr{I} = [I_{n_y}, \ldots, I_{n_y}]^\top$, $y(k)$ is the measured plant output at discrete time step k, $N = I_{n_y} \otimes e_{L+1}^\top$, and $\otimes$ is the Kronecker product, e_{L+1} is an $L+1$-dimensional vector with its first element equal to one and its remaining elements equal to zero.

The term $\mathscr{I}^{\text{PCE}}(y(k) - N\bar{Y}_k^{\text{PCE}})$ is the expression above represents the feedback correction, which is analogous the additive disturbance update in standard QDMC. The main difference here is that the observer output is stochastic. Therefore, we propose to correct over the horizon based on error between the measured value and the observer's mean value at the current time as this is the output's most likely value, which is approximated from the PCE coefficients by $N\bar{Y}_k^{\text{PCE}}$.

From (6.8), we know that the mean of a random variable can be approximated by the first coefficient of its PCE. Using the notation introduced above, we can derive

$$\mathbb{E}[\mathscr{Y}_{k+1|k}(\theta)] = (I_{n_y p} \otimes e_{L+1}^\top)\mathscr{Y}_{k+1|k}^{\text{PCE}} = E\mathscr{Y}_{k+1|k}^{\text{PCE}}. \tag{6.27}$$

The term $\mathbb{E}\left[\mathscr{Y}_{k+1|k}(\theta)^\top \mathbf{Q} \mathscr{Y}_{k+1|k}(\theta)\right]$ is related to the variance/covariance of $\mathscr{Y}_{k+1|k}(\theta)$ so that we can use (6.9) to determine an (approximate) expression of this term in terms of the predicted output PCE coefficients. To simplify notation, the subscript $k+1|k$ is dropped in the following derivation, $\mathscr{Y}_i$ and $\mathscr{Y}_i^{\text{PCE}}$ will be used to denote the i^{th} element of their respective vectors, and q_{ij} will be used to denote the $(i,j)^{\text{th}}$ element of $\mathbf{Q}$.

$$\begin{aligned}
\mathbb{E}\left[\mathscr{Y}(\theta)^\top \mathbf{Q} \mathscr{Y}(\theta)\right] &= \sum_{i=1}^{n_y p}\sum_{j=1}^{n_y p} q_{ij}\mathbb{E}[\mathscr{Y}_i(\theta)\mathscr{Y}_j(\theta)], \qquad (6.28)\\
&\approx \sum_{i=1}^{n_y p}\sum_{j=1}^{n_y p} q_{ij}\sum_{k=0}^{L} \mathscr{Y}_{1+(i-1)(L+1)+k}^{\text{PCE}}\mathscr{Y}_{1+(j-1)(L+1)+k}^{\text{PCE}}\langle\Phi_k^2\rangle,\\
&= \left(\mathscr{Y}^{\text{PCE}}\right)^\top \mathbf{Q}^{\text{PCE}}\mathscr{Y}^{\text{PCE}},
\end{aligned}$$

where $\mathbf{Q}^{\text{PCE}} = \mathbf{Q} \otimes P$.

Since (6.27) and (6.28) are linear and quadratic in $\mathscr{Y}_{k+1|k}^{\text{PCE}}$, respectively, and, as shown in (6.26) $\mathscr{Y}_{k+1|k}^{\text{PCE}}$ is linear in the input variables, the objective function and

constraints are a convex functions with respect to $\Delta\mathscr{U}_k$. This implies that Problem 6.1 can be rewritten as a quadratic program (QP) when using polynomial chaos as the uncertainty propagation tool. This QP can be formulated and solved using the numerous methods detailed in Chapter 4.

6.6 Example: End-to-end Continuous Pharmaceutical Manufacturing

The proposed fast SMPC approach is applied to control an end-to-end continuous process for manufacturing pharmaceutical tablets [18]. This is the same example detailed in Chapter 5, while here the proposed control algorithm is extended to handle parameter uncertainty. The dynamics of this high-dimensional system are described by a set of nonlinear DAEs with nearly 8000 states. The process has nine inputs and three outputs—the active pharmaceutical ingredient (API) dosage of tablets, the impurity content of tablets, and the production rate of the process. The critical quality attributes (CQAs) of the manufactured tablets consist of the API dosage and impurity content of the tablets, which should be effectively regulated in the presence of process uncertainties and disturbances.

In this work, the nonlinear DAE model is linearized around a desired steady-state operating condition and, subsequently, the linearized model is used to develop a fast SMPC controller (from applying FSR modeling and PCEs to solve Problem 6.1 as discussed above) to suppress the adverse effects of uncertainties in kinetic parameters on the CQAs of tablets. The objective is stated as a setpoint tracking problem for the production rate and API with an upper bound on the impurity content of the tablets included as a constraint. The performance of the fast SMPC controller is evaluated for the case of a 5% step increase in the production rate. Figure 6-2 shows the production rate for the fast SMPC and QDMC for 200 closed-loop simulations, where the step change in the setpoint is applied at 5 hr. The fast SMPC controller results in a lower variance in the production rate than that of the QDMC controller (Figure 6-2),

providing a more robust performance in the presence of system uncertainties.

The SMPC approach leads to tighter setpoint tracking of the API dosage profiles as well (see Figure 6-3). Figure 6-4 shows the distributions of the API dosage at different points in the course of the transient system dynamics resulting from the production rate setpoint change. The SMPC produces a lower variance in the API dosage at all times. The variance in the API dosage at time 65 hr is a factor of 25 lower for the fast SMPC controller than that of the QDMC controller (Table 6.1). The optimization consistently took less than one second to solve (on a laptop running Windows 7 with 8 GB of RAM) for a variety of prediction and control horizons, meaning that the control inputs could easily be found and supplied to the plant in real-time. Given the high importance of drug products meeting the specifications in the presence of uncertainties, these results indicate that SMPC is a promising approach for application to pharmaceutical manufacturing. Moreover, the speed at which the proposed fast MPC algorithm is able to compute the solution indicates that algorithm could easily be implemented in real-time on large and complex processes.

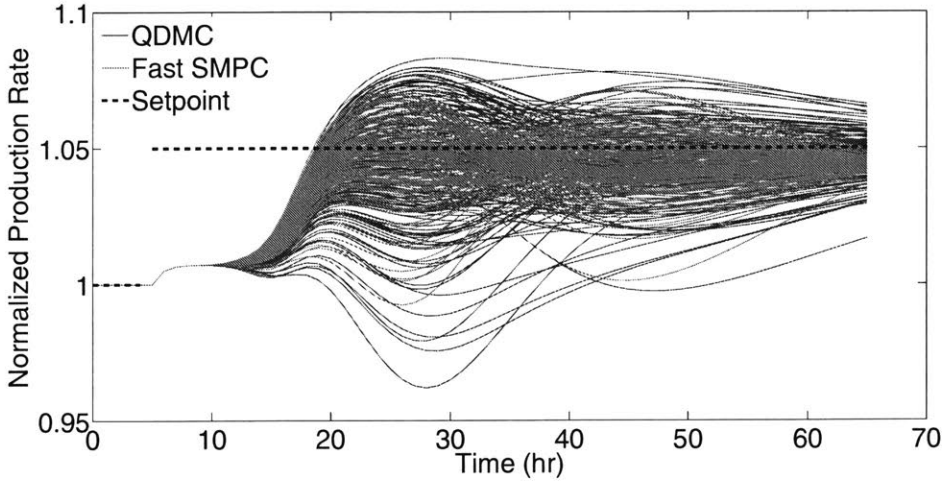


Figure 6-2: Dynamic response of production rate (normalized) for 200 closed-loop simulations of a setpoint change in the production rate.

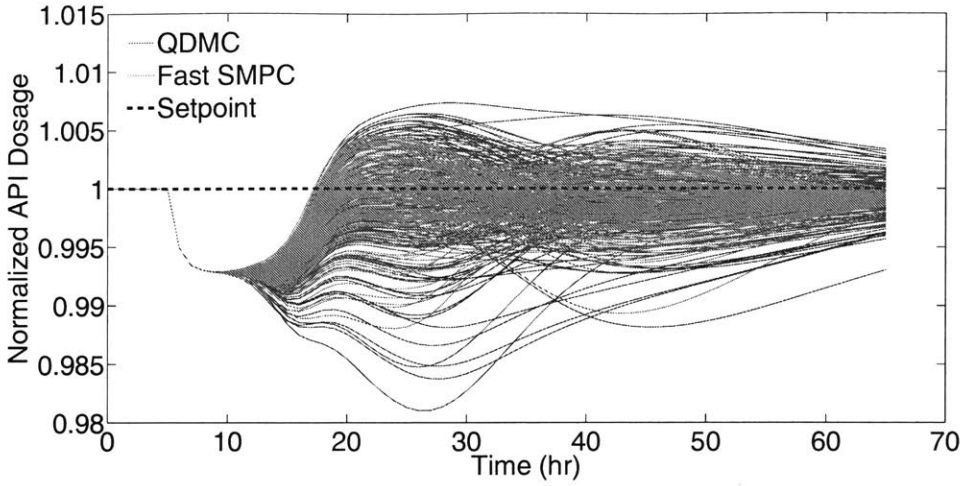


Figure 6-3: Dynamic response of API dosage (normalized) for 200 closed-loop simulations of a setpoint change in the production rate.

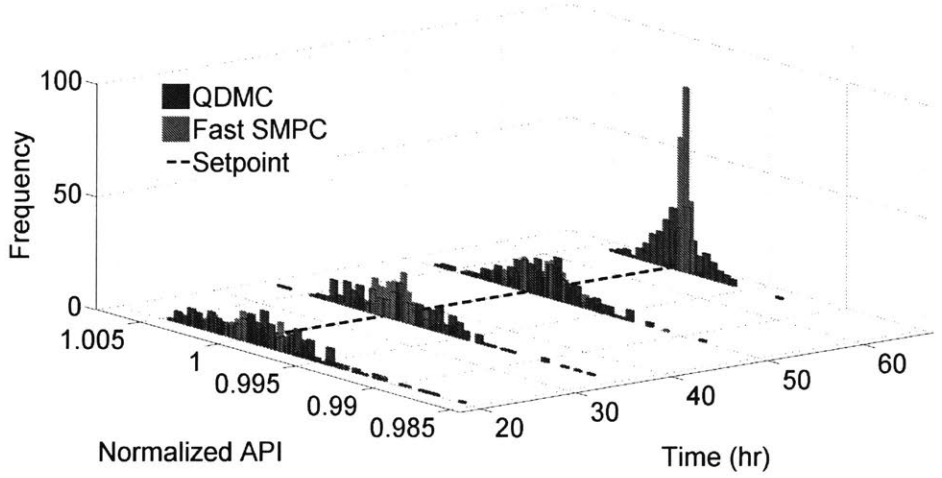


Figure 6-4: Histograms of the API dosage (normalized) at various times based on 200 closed-loop simulations of a setpoint change in the production rate.

Table 6.1: Variance comparisons of API dosage (normalized) at various times based on 200 closed-loop simulations of a setpoint change in the production rate.

Time (hr)	QDMC Variance of API $\times 10^6$	SQMPC Variance of API $\times 10^6$
20	11.57	7.47
35	10.70	2.68
50	6.98	3.99
65	2.42	0.09

6.7 Conclusions

This chapter presents a stochastic MPC approach for high-dimensional systems with time-invariant probabilistic uncertainties. The concepts of quadratic dynamic matrix control and polynomial chaos expansion are used to develop an input-output formulation for fast SMPC whose on-line computational cost is independent of the state dimension. This approach circumvents the prohibitive online computational costs associated with model predictive control of uncertain high-dimensional systems. The Galerkin projection is modified to handle a general class of linear DAEs so that the SMPC approach is applicable to a large class of complex systems that include descriptor/singular systems. The effectiveness of the proposed control approach is demonstrated for control of a continuous pharmaceutical manufacturing process with nearly 8000 states, where the SMPC controller effectively regulates the critical quality attributes of the product in the presence of parametric uncertainties. In addition, the proposed control law was found in less than one second at each time step so that the algorithm can easily be implemented in real-time.

Chapter 7

Optimal Risk Allocation for Disturbance Rejection

7.1 Introduction

Recent years have witnessed significant developments in the area of robust model predictive control (MPC) with the aim to devise optimal control approaches that enable systematic handling of system uncertainties [159]. Generally, robust MPC approaches consider bounded, deterministic descriptions of uncertainties. The deterministic approaches to robust MPC commonly use a min-max optimal control formulation in which the control policy is designed with respect to the worst-case performance and system constraints are satisfied for all possible uncertainty realizations [14]. These approaches can lead to overly conservative or possibly infeasible control designs.

In practice, system uncertainties are often considered to be of stochastic nature. When the stochastic description of uncertainties is available, a natural approach to robust MPC involves explicitly accounting for the probabilistic occurrence of uncertainties in designing the robust control policy. This consideration has led to the emergence of stochastic MPC (SMPC). A core component of SMPC is (state) chance constraints that allow for constraint satisfaction in a probabilistic sense. Chance constraints enable SMPC to trade off robustness to uncertainties (in terms of constraint satisfaction) with control performance in a systematic manner, possibly resulting in

less conservative robust control performance.

A recent review on different SMPC approaches and their applications is given in [164]. *Stochastic tube approaches* to SMPC, e.g., [130], use probabilistic tubes with fixed or variable cross sections to replace chance constraints with linear constraints on the nominal state predictions as well as to construct terminal sets for guaranteeing recursive feasibility. These approaches use a prestabilizing feedback controller to ensure closed-loop stability. However, stochastic tube approaches cannot handle hard input constraints as the prestabilizing state feedback controller is determined offline. SMPC approaches based on an *affine parametrization* of the feedback control law have been extensively investigated, e.g., [105, 192, 202]. Such control law parametrizations allow for obtaining convex SMPC algorithms while solving the stochastic optimal control problem over the feedback gains as well as the open-loop control actions. The notion of affine disturbance (or equivalently state) parametrization of the feedback control laws originates from the fact that disturbance realizations and system states will be known at the future time instants [89]. Therefore, the controller can use this information when determining the future control inputs over the control horizon. A key challenge in using such parametrizations, however, arises from handling hard input constraints in the presence of unbounded stochastic uncertainties (e.g., Gaussian noise), as unbounded uncertainties almost surely lead to excursions of states from any bounded set. To address this challenge, the inclusion of a *saturation function* into the affine feedback control policy has been proposed [105]. Saturation functions render the feedback control policy nonlinear to enable direct handling of hard input constraints without relaxing hard input constraints to input chance constraints. Extensive work has also been reported on SMPC approaches that use the so-called *sample-based approaches* (aka scenario-based approaches). These algorithms characterize the stochastic system dynamics using a finite set of random realizations of uncertainties, which are used to solve the optimal control problem in one shot, e.g., [27, 38]. This class of SMPC approaches typically does not rely on any convexity requirements; however, establishing the recursive feasibility and closed-loop stability of these algorithms is generally challenging, particularly for the case of unbounded

uncertainties.

This chapter considers the MPC problem for stochastic linear systems with arbitrary (possibly unbounded) disturbances. A SMPC approach is presented that handles both joint state chance constraints and hard input constraints under closed-loop prediction in the presence of unbounded additive disturbances. The key contribution of this work lies in using risk allocation [257, 152] in combination with the Cantelli-Chebyshev inequality [155] to obtain computationally tractable surrogates for the joint state chance constraints when only the first two moments of an arbitrary disturbance distribution are known (i.e., the full distribution is unknown). An algorithm is presented for solving the SMPC problem to determine the optimal feedback gain and optimal risk allocation iteratively. The problem setup is similar to that in [105, 67]. In contrast to [105], the proposed SMPC approach accounts for both hard input constraints and state chance constraints. What distinguishes this work from [67] is the direct handling of hard input constraints without relaxation as well as the convexity of the optimization program. The proposed SMPC approach is demonstrated on a continuous acetone-butanol-ethanol (ABE) fermentation process [98], which is used in production of high value-added drop-in biofuels from lignocellulosic biomass. The performance of the proposed approach is evaluated with respect to that of a certainty equivalence MPC algorithm and a MPC algorithm with fixed uniform risk allocation.

Notation In this chapter, $\mathbb{R}$ and $\mathbb{N} = \{1, 2, \ldots\}$ are the sets of real and natural numbers, respectively; $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$. $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ are the sets of positive semidefinite and definite matrices, respectively. I_N denotes the N by N identity matrix and $\mathbf{1}_N$ denotes a column vector of ones of length N. $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. $\|\cdot\|_p$ denotes the standard p-norms. $\otimes$ denotes the Kronecker product. For given random vectors X and Y, $\mathbb{E}[X]$ denotes the expected value, $\sigma[X, Y] \triangleq \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])^\top]$ denotes the cross covariance matrix, and $\Sigma[X] \triangleq \sigma[X, X]$ denotes the covariance matrix. $\mathbb{P}(A)$ denotes the probability of event A. $\mathbb{1}_A(\cdot)$ is the indicator function defined on set A.

7.2 Problem Formulation

Consider a discrete-time stochastic linear system

$$x^+ = Ax + Bu + Gw, \tag{7.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $w \in \mathbb{R}^p$ are the system states, inputs, and disturbances at the current time, respectively; x^+ denotes the system states at the next time; and A, B, and G are the known system matrices. It is assumed that the states x are observed exactly at all times (although this assumption can be relaxed using methods reviewed in Chapter 4), and the disturbances are mutually independent such that system (7.1) is a Markov process. The disturbances w can have an arbitrary (unbounded) distribution that is unknown to the controller; the mean $\mathbb{E}[w]$ and covariance $\Sigma[w] \in \mathbb{S}^p_+$ are, however, assumed to be known.

Let $N \in \mathbb{N}$ be the prediction horizon of the predictive control problem. The states, inputs, and disturbances over the prediction horizon are defined, respectively, by

$$\begin{aligned}
\mathbf{x} &\triangleq (x_0, x_1, \ldots, x_N) \\
\mathbf{u} &\triangleq (u_0, u_1 \ldots, u_{N-1}) \\
\mathbf{w} &\triangleq (w_0, w_1, \ldots, w_{N-1}),
\end{aligned}$$

where $x_k = Ax_{k-1} + Bu_{k-1} + Gw_{k-1}$ is the predicted states k steps ahead from the known current states $x_0 = x$; and u_k and w_k are the inputs and disturbances k time steps into the future, respectively. Using this compact notation, the system model is written as

$$\mathbf{x} = \mathbf{A}x_0 + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{G}\mathbf{w}, \tag{7.2}$$

where the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$, and $\mathbf{G}$ can be straightforwardly derived, e.g., see [104] for details on the explicit construction of these matrices.

The control inputs are assumed to be constrained to a convex feasible region F_U

described by a finite set of N_U linear inequalities

$$F_U \triangleq \{\mathbf{u} \mid \mathbf{H}\mathbf{u} \leq \mathbf{h}\}, \tag{7.3}$$

where $\mathbf{H} \in \mathbb{R}^{N_U \times Nm}$ and $\mathbf{h} \in \mathbb{R}^{N_U}$. The system states are also restricted to lie in a convex region F_X, which is defined by a collection of N_X linear inequality constraints

$$F_X \triangleq \bigcap_{i=1}^{N_X} \{\mathbf{x} \mid \mathbf{a}_i^\top \mathbf{x} \leq b_i\}, \tag{7.4}$$

with $\mathbf{a}_i \in \mathbb{R}^{(N+1)n}$ and $b_i \in \mathbb{R}$. For linear systems, the cost function is typically chosen to be quadratic, i.e.,

$$V_N(x_0, \mathbf{u}, \mathbf{w}) = \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}, \tag{7.5}$$

where $\mathbf{Q} \in \mathbb{S}_+^{(N+1)n}$ and $\mathbf{R} \in \mathbb{S}_{++}^{Nm}$ are specified weight matrices. As the distribution of the disturbances is unknown and could be unbounded, it cannot be guaranteed that there always exists a control action such that hard state constraints $\mathbf{x} \in F_X$ are satisfied. Therefore, $\mathbf{x} \in F_X$ is replaced with a *joint state chance constraint* of the form

$$\mathbb{P}(\mathbf{x} \notin F_X) \leq \delta, \tag{7.6}$$

where $\delta \in (0, 1)$ is the maximum probability of constraint violation. The SMPC problem is now stated as

Problem 7.1 (SMPC)

$$\begin{aligned} \min_{\mathbf{u}} \quad & \mathbb{E}[V_N(x, \mathbf{u}, \mathbf{w})] \\ \text{subject to:} \quad & \mathbf{x} = \mathbf{A}x_0 + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{G}\mathbf{w} \\ & \mathbf{u} \in F_U, \quad \mathbb{P}(\mathbf{x} \notin F_X) \leq \delta, \quad x_0 = x \end{aligned} \tag{7.7}$$

Problem 7.1 is solved online given the most recently observed states x. Let $\mathbf{u}^\star(x)$

be the optimal feedback control policy that solves Problem 7.1 as a function of the initial states. The receding-horizon implementation of Problem 7.1 implies that only the first element of this policy, $u_0^\star$, is applied to the system (7.1). Note that $V_N(x_0, \mathbf{u}, \mathbf{w})$ is a random variable with an unknown distribution (as it is a function of $\mathbf{w}$). Here, the expected value of the value function is optimized to obtain a convex program.

There are two main difficulties that prevent direct solution of Problem 7.1. First, the control input $\mathbf{u}$ should be a causal feedback policy that is some function of the current and past states. In general, solving Problem 7.1 over arbitrary functions of states is impractical using available optimal control approaches. Second, the distribution of the disturbances is unknown and possibly unbounded, which makes handling the hard input constraints and joint state chance constraints challenging.

To address these challenges, a certain class of causal feedback policies is adopted to define the control policy $\mathbf{u}$. The adopted feedback policy allows for building feedback into the prediction to reduce uncertainty in the state predictions as well as directly handling the input constraints in the face of unbounded disturbances. In addition, the joint state chance constraints are approximated for arbitrary disturbance distributions using distributionally-robust bounds that are only a function of the mean and variance of the stochastic disturbance (i.e., bounds do not depend on the full distribution of the disturbance, which is usually not known in practical applications).

7.3 Feedback Parametrization of the Controller

A natural approach to obtaining a computationally tractable surrogate for Problem 7.1 is to adopt an affine state feedback parametrization for the control policy $\mathbf{u}$. Affine state feedback is in fact the solution to the Linear-quadratic-Gaussian (LQG) problem, which minimizes (7.5) in the absence of the input and state constraints. Solving Problem 7.1 over an affine state feedback control policy, however, results in a nonconvex optimization due to the product of the gains over time. An alternative

parametrization is an affine function of the sequence of past disturbances [89]

$$u_i \triangleq \sum_{j=0}^{i-1} M_{i,j} G w_j + v_i, \quad \forall i = 0, \ldots, N-1, \tag{7.8}$$

where $M_{i,j} \in \mathbb{R}^{m \times n}$ and $v_i \in \mathbb{R}^m$. This parametrization yields convex optimizations, and is shown to be equivalent to the class of feedback control policies that are affine in the past states [89, Theorem 9]. Using (7.8), the control policy $\mathbf{u}$ can be written as

$$\mathbf{u} = \mathbf{M}\mathbf{G}\mathbf{w} + \mathbf{v}, \tag{7.9}$$

where the block lower triangular matrix $\mathbf{M} \in \mathbb{R}^{mN \times nN}$ and stacked vector $\mathbf{v} \in \mathbb{R}^{mN}$ are given by

$$\mathbf{M} \triangleq \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ M_{1,0} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ M_{N-1,0} & \cdots & M_{N-1,N-2} & 0 \end{bmatrix}, \tag{7.10}$$

$$\mathbf{v} \triangleq (v_0, v_1, \ldots, v_{N-1}). \tag{7.11}$$

The pair $(\mathbf{M}, \mathbf{v})$ comprises the decision variables in Problem 7.1.

A key challenge in using the feedback control policy (7.9) arises from guaranteeing the hard input constraints (7.3) in the presence of unbounded disturbances. SMPC algorithms commonly overcome this difficulty by relaxing the hard input constraints to expectation-type constraints [208] or probabilistic chance constraints [67]. These approaches, however, suffer from the fact that the computed inputs may not be feasible in practice, which will cause the controller to saturate. In this work, a saturated disturbance affine parametrization of the form [104]

$$\mathbf{u} = \mathbf{M}\mathbf{G}\boldsymbol{\varphi}(\mathbf{w}) + \mathbf{v}, \tag{7.12}$$

is used, where, for any vector $z = (z_1, \ldots, z_n)$, $\boldsymbol{\varphi}(z) \triangleq (\varphi(z_1), \ldots, \varphi(z_n))$ with $\varphi : \mathbb{R} \to \mathbb{R}$ denoting any function with the property $\sup_{a \in \mathbb{R}} |\varphi(a)| \leq \varphi_{\max}$ for some $\varphi_{\max} > 0$. The functions $\varphi(\cdot)$ are known as *saturation functions* [104]. This definition implies that $\|\boldsymbol{\varphi}(\mathbf{w})\|_\infty \leq \varphi_{\max}$, which can be written as a polytope of the form

$$F_W \triangleq \{\mathbf{w} \mid \mathbf{S}\mathbf{w} \leq \mathbf{s}\},$$

with $\mathbf{S} \in \mathbb{R}^{N_W \times N_P}$ and $\mathbf{s} \in \mathbb{R}^{N_W}$, and allows the hard input constraints $\mathbf{u} \in F_U$ to be rewritten as

$$\mathbf{H}\mathbf{v} + \max_{\boldsymbol{\varphi}(\mathbf{w}) \in F_W} (\mathbf{HMG}\boldsymbol{\varphi}(\mathbf{w})) \leq \mathbf{h}, \tag{7.13}$$

where the maximization is row-wise (i.e., maximum of each element in the vector). The below lemma indicates that (7.13) can be defined by a set of linear inequalities.

Lemma 7.1 *The input constraint* (7.13) *is represented exactly by linear inequalities* $\mathbf{H}\mathbf{v} + \mathbf{Z}^\top \mathbf{s} \leq \mathbf{h}$ *and* $\mathbf{Z} \geq 0$ *(element-wise) for any* $\mathbf{Z}$ *satisfying* $\mathbf{Z}^\top \mathbf{S} = \mathbf{HMG}$.

Proof. The proof follows from the concept of the dual norm as shown in, e.g., [24]. Let the i^{th} row of the maximization in (7.13) be the primal linear program. The corresponding dual linear program is

$$\min\ \mathbf{s}^\top \mathbf{z}_i, \quad \text{s.t.:}\ \mathbf{S}^\top \mathbf{z}_i = (\mathbf{HMG})_{(i)}^\top, \quad \mathbf{z}_i \geq 0,$$

where $(\mathbf{HMG})_{(i)}$ denotes the i^{th} row of $\mathbf{HMG}$ and $\mathbf{z}_i \in \mathbb{R}^{N_W}$ denotes the dual variables. By the strong duality theorem, it is known that

$$\max_{\boldsymbol{\varphi}(\mathbf{w}) \in F_W} (\mathbf{HMG})_{(i)} \boldsymbol{\varphi}(\mathbf{w}) \leq \mathbf{s}^\top \mathbf{z}_i$$

holds for any $\mathbf{z}_i$ satisfying the dual linear program constraints. Stacking the dual variables into a matrix $\mathbf{Z} \triangleq [\mathbf{z}_1, \ldots, \mathbf{z}_{N_U}]$ yields the inequality

$$\max_{\boldsymbol{\varphi}(\mathbf{w}) \in F_W} (\mathbf{HMG}\boldsymbol{\varphi}(\mathbf{w})) \leq \mathbf{Z}^\top \mathbf{s}$$

for any $\mathbf{Z} \geq 0$ satisfying $\mathbf{Z}^\top \mathbf{S} = \mathbf{HMG}$. Hence, the assertion of the lemma directly follows. □

7.4 Joint State Chance Constraints

It is generally impractical to ensure that the system states lie in the feasible region $\mathbf{x} \in F_X$ when the disturbances $\mathbf{w}$ are unbounded. Hence, the hard state constraints (7.4) should be replaced with the joint chance constraint (7.6), as in Problem 7.1. Joint chance constraints are, however, intractable and nonconvex.

To obtain a tractable deterministic surrogate for (7.6), this work uses Boole's inequality to bound the probability of violation of the joint chance constraint

$$\begin{aligned} \mathbb{P}(\mathbf{x} \notin F_X) &= \mathbb{P}\left(\mathbf{x} \in \bigcup_{i=1}^{N_X} \left\{\mathbf{x} \mid \mathbf{a}_i^\top \mathbf{x} > b_i\right\}\right) \\ &\leq \sum_{i=1}^{N_X} \mathbb{P}(\mathbf{a}_i^\top \mathbf{x} > b_i). \end{aligned} \tag{7.14}$$

This expression implies that the joint chance constraint (7.6) can be replaced with N_X individual chance constraints of the form

$$\mathbb{P}(\mathbf{a}_i^\top \mathbf{x} > b_i) \leq \epsilon_i, \ i = 1, \ldots, N_X, \tag{7.15}$$

where $\epsilon_i \in [0, \delta]$ denotes the violation probability for the i^{th} individual chance constraint. When the so-called *risk allocation* ϵ_i is chosen such that

$$\textstyle\sum_{i=1}^{N_X} \epsilon_i \leq \delta$$

is satisfied, then the joint chance constraint (7.6) will be satisfied according to (7.14). Two main approaches exist for defining the risk allocation. The first assumes a fixed risk allocation in which the values of ϵ_i are fixed *a priori*, usually using a uniform allocation $\epsilon_i = \delta / N_X$ [184]. Although this approach simplifies the optimization problem, it may lead to significant conservatism in many situations as the prespecified

risk may be better allocated to other constraints. To address this shortcoming, the second approach optimizes the risk allocation by treating ϵ_i as decision variables in the optimization [26].

The knowledge of the cumulative distribution function (cdf) of the disturbances $\mathbf{w}$ is required to exactly evaluate the individual chance constraints (7.15). Once this knowledge is available, the cdf of $\mathbf{x}$ can be straightforwardly determined using the linear relationship (7.2). Then, the probability of violating an individual chance constraint is given by [257]

$$\mathbb{P}(\mathbf{a}_i^\top \mathbf{x} > b_i) = 1 - \mathrm{cdf}_{\mathbf{a}_i^\top \mathbf{x}}(b_i).$$

This expression is, however, difficult to evaluate for general disturbances as their cdfs do not necessarily have a convex form. More importantly, the distribution of disturbances is not known in many practical applications. Hence, the Cantelli-Chebyshev inequality is used in this study to evaluate the individual chance constraints (7.15) for arbitrary distributions of disturbances when only their first two moments are known.

Lemma 7.2 (Cantelli-Chebyshev Inequality [155]) *Let Z be a scalar random variable with finite variance. For every $c \geq 0$, it holds that*

$$\mathbb{P}(Z \geq \mathbb{E}[Z] + c) \leq \frac{\Sigma[Z]}{\Sigma[Z] + c^2}.$$

Proof. Let $p_Z(z)$ denote the PDF of random variable Z. Define a corresponding zero-mean random variable $Y \triangleq Z - \mathbb{E}[Z]$ with PDF $p_Y(y)$. Consider $\mathbb{P}(Z \geq \mathbb{E}[Z] + c) = \mathbb{P}(Y \geq c) = \int_c^\infty p_Y(Y)dz = \mathbb{E}(\mathbb{1}_{[c,\infty)}(Y))$. Define the function $h(y) \triangleq (c+b)^{-2}(y+b)^2$ for any $c \geq 0$ and $b \geq 0$. It is evident that $\mathbb{1}_{[c,\infty)}(y) \leq h(y)$, $\forall y \in \mathbb{R}$ such that $\mathbb{E}[\mathbb{1}_{[c,\infty)}(Y)] \leq \mathbb{E}[h(Y)] = (c+b)^{-2}\mathbb{E}[(Y+b)^2] = (c+b)^{-2}(\Sigma[Y] + b^2)$. The smallest upper bound for $\mathbb{P}(Z \geq \mathbb{E}[Z]+c)$ is obtained by minimizing the right-hand side of the latter inequality with respect to b. The solution to this minimization is $b^\star = \Sigma[Y]/a$. The assertion follows by noticing $\Sigma[Y] = \Sigma[Z]$ and substituting $b^\star$ for b into the upper bound. □

To apply the result of Lemma 7.2 to (7.15), assume some $\Delta b_i \geq 0$ exists such that

$$\mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \Delta b_i \leq b_i. \tag{7.16}$$

The goal is to derive a lower bound on Δb_i. Notice that

$$\begin{aligned}\mathbb{P}(\mathbf{a}_i^\top \mathbf{x} > b_i) &\leq \mathbb{P}(\mathbf{a}_i^\top \mathbf{x} \geq \mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \Delta b_i), \\ &\leq \frac{\mathbf{a}_i^\top \Sigma[\mathbf{x}]\mathbf{a}_i}{\mathbf{a}_i^\top \Sigma[\mathbf{x}]\mathbf{a}_i + \Delta b_i^2}.\end{aligned}$$

When this upper bound is less than or equal to ϵ_i, then the individual chance constraint (7.15) must be satisfied. This inequality implies that

$$\sqrt{\frac{1-\epsilon_i}{\epsilon_i}}\sqrt{\mathbf{a}_i^\top \Sigma[\mathbf{x}]\mathbf{a}_i} \leq \Delta b_i. \tag{7.17}$$

Combining (7.16) with (7.17), the individual chance constraint (7.15) can be (conservatively) approximated by the deterministic constraint

$$\mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \sqrt{\frac{1-\epsilon_i}{\epsilon_i}}\sqrt{\mathbf{a}_i^\top \Sigma[\mathbf{x}]\mathbf{a}_i} \leq b_i, \tag{7.18}$$

which is guaranteed to hold for any distribution of the states $\mathbf{x}$. This result has been derived previously in [37, Theorem 3.1]. The key contribution of this work is to combine this result with optimal risk allocation to substantially reduce the conservatism of (7.18). In the next section, we present an MPC formulation that incorporates these robust constraints while also including feedback into the predictions such that the state variance can be shaped by the controller.

7.5 Optimizing Feedback and Risk Allocation Simultaneously

This section uses the saturated affine disturbance parametrization of the control inputs in conjunction with the risk allocation method for bounding the joint state chance constraint to obtain a tractable formulation for the SMPC Problem 7.1. To this end, explicit expressions are first derived for the mean and covariance of the states $\mathbf{x}$. Using the system model (7.2), the dynamics for $\mathbb{E}[\mathbf{x}]$ and $\Sigma[\mathbf{x}]$ are described by

$$\mathbb{E}[\mathbf{x}] = \mathbf{A}x_0 + \mathbf{B}\mathbb{E}[\mathbf{u}] + \mathbf{DG}\mathbb{E}[\mathbf{w}], \tag{7.19a}$$

$$\Sigma[\mathbf{x}] = \mathbf{B}\Sigma[\mathbf{u}]\mathbf{B}^\top + \mathbf{DG}\Sigma[\mathbf{w}]\mathbf{G}^\top\mathbf{D}^\top + \mathbf{B}\sigma[\mathbf{u},\mathbf{w}]\mathbf{G}^\top\mathbf{D}^\top + \mathbf{DG}\sigma[\mathbf{u},\mathbf{w}]^\top\mathbf{B}^\top, \tag{7.19b}$$

where $\mathbb{E}[\mathbf{w}] = \mathbf{1}_N \otimes \mathbb{E}[w]$ and $\Sigma[\mathbf{w}] = I_N \otimes \Sigma[w]$ are assumed to be known. The statistics of the control inputs $\mathbf{u}$ are derived from (7.12) as

$$\mathbb{E}[\mathbf{u}] = \mathbf{MG}\mathbb{E}[\boldsymbol{\varphi}(\mathbf{w})] + \mathbf{v}, \tag{7.20a}$$

$$\Sigma[\mathbf{u}] = \mathbf{MG}\Sigma[\boldsymbol{\varphi}(\mathbf{w})]\mathbf{G}^\top\mathbf{M}^\top, \tag{7.20b}$$

$$\sigma[\mathbf{u},\mathbf{w}] = \mathbf{MG}\sigma[\boldsymbol{\varphi}(\mathbf{w}),\mathbf{w}]. \tag{7.20c}$$

For any chosen saturation function $\boldsymbol{\varphi}(\cdot)$, the following statistics $\mathbb{E}[\boldsymbol{\varphi}(\mathbf{w})]$, $\Sigma[\boldsymbol{\varphi}(\mathbf{w})]$, and $\sigma[\boldsymbol{\varphi}(\mathbf{w}),\mathbf{w}]$ can be straightforwardly computed by applying the saturation function to the data used to estimate the mean and covariance of the disturbances $\mathbf{w}$. The mean and variance equations (7.19) and (7.20) are used to recast Problem 7.1 to the following program.

Problem 7.2 (Deterministic Surrogate for SMPC Problem)

$$
\begin{aligned}
\min_{\mathbf{M},\mathbf{v},\boldsymbol{\epsilon}} \quad & \mathbb{E}[\mathbf{x}]^\top \mathbf{Q}\mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{u}]^\top \mathbf{R}\mathbb{E}[\mathbf{u}] + \mathrm{tr}(\mathbf{Q}\Sigma[\mathbf{x}]) + \mathrm{tr}(\mathbf{R}\Sigma[\mathbf{u}]) \\
\text{subject to:} \quad & (\mathbb{E}[\mathbf{x}], \Sigma[\mathbf{x}]) && \text{given by (7.19)} && \text{(7.21a)} \\
& (\mathbb{E}[\mathbf{u}], \Sigma[\mathbf{u}], \sigma[\mathbf{u},\mathbf{w}]) && \text{given by (7.20)} && \text{(7.21b)} \\
& \mathbf{M} && \text{satisfies (7.10)} && \text{(7.21c)} \\
& (\mathbf{M}, \mathbf{v}) && \text{satisfy Lemma 1} && \text{(7.21d)} \\
& \beta_i && = \sqrt{(1-\epsilon_i)/\epsilon_i} && \text{(7.21e)} \\
& \nu_i && = \sqrt{\mathbf{a}_i^\top \Sigma[\mathbf{x}]\mathbf{a}_i} && \text{(7.21f)} \\
& \mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \beta_i \nu_i && \leq b_i && \text{(7.21g)} \\
& \epsilon_i && \geq 0 && \text{(7.21h)} \\
& \textstyle\sum_{i=1}^{N_X} \epsilon_i && \leq \delta && \text{(7.21i)} \\
& x_0 && = x && \text{(7.21j)} \\
& \forall i = 1, \ldots, N_X
\end{aligned}
\tag{7.21}
$$

where the risk allocation of the joint state constraint violation is defined in terms of $\boldsymbol{\epsilon} \triangleq (\epsilon_1, \ldots, \epsilon_{N_X})$.

Convexity Analysis

Problem 7.2 is nonconvex due to the multiplication of ν_i and β_i in (7.21g), which makes simultaneous optimization over the feedback gain $\mathbf{M}$ and risk allocation $\boldsymbol{\epsilon}$ a nonconvex problem. An iterative strategy can be devised to solve Problem 7.2 by taking advantage from the fact that the optimization problem is convex when either $\mathbf{M}$ or $\boldsymbol{\epsilon}$ is fixed. To prove this, notice that $\mathbb{E}[\mathbf{u}]$ and $\mathbb{E}[\mathbf{x}]$ are linear functions of the decision variables $\mathbf{v}$ and $\mathbf{M}$, while $\Sigma[\mathbf{u}]$ and $\Sigma[\mathbf{x}]$ are quadratic functions of $\mathbf{M}$. Thus, the objective function is quadratic in $\mathbf{v}$ and $\mathbf{M}$, and is a convex function of the decision variables since $\mathbf{Q}$ and $\mathbf{R}$ are assumed to be positive semidefinite and definite matrices, respectively.

Requiring $\mathbf{M}$ to be lower block triangular can be represented by linear equality constraints so that (7.21c) is convex. The hard input constraints (7.21d) are exactly

represented by a set of linear inequalities that are convex in $\mathbf{v}$ and $\mathbf{M}$ (see Lemma 7.1). Clearly, (7.21h)–(7.21j) are linear inequalities or equalities, which are convex.

Now, let us consider the surrogate expressions for the state chance constraints (7.21g). When the feedback gain $\mathbf{M}$ is fixed, $\Sigma[\mathbf{u}]$ and $\Sigma[\mathbf{x}]$ must be constant matrices from (7.20) and (7.19), respectively. Therefore, ν_i from (7.21f) will be constant for all $i = 1, \ldots, N_X$. Since the risk allocation $\boldsymbol{\epsilon}$ still comprises the decision variables, (7.21g) reduces to $\mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \nu_i \sqrt{(1-\epsilon_i)/\epsilon_i} \leq b_i$. The first term is an affine function of $\mathbf{v}$. The second term is convex for any $\epsilon_i \in [0, 0.75]$, which can be verified by observing that the second derivative of $\sqrt{(1-\epsilon_i)/\epsilon_i}$ is positive on this range. Since the sum of convex functions is a convex function, (7.21g) will be convex for any fixed $\mathbf{M}$ and any choice of $\delta \leq 0.75$.

On the other hand, when the risk allocation $\boldsymbol{\epsilon}$ is fixed, β_i from (7.21e) will be constant for all $i = 1, \ldots, N_X$. In this case, (7.21g) reduces to $\mathbf{a}_i^\top \mathbb{E}[\mathbf{x}] + \beta_i \nu_i \leq b_i$ where the first term is linear in $\mathbf{v}$ and the second term is linear in ν_i. By substituting the expression for $\Sigma[\mathbf{x}]$ in (7.21f), this constraint can be rewritten as a second-order cone constraint

$$\nu_i = \left\| \begin{bmatrix} \Sigma[\boldsymbol{\varphi}(\mathbf{w})] & \sigma[\boldsymbol{\varphi}(\mathbf{w}), \mathbf{w}] \\ \star & \Sigma[\mathbf{w}] \end{bmatrix}^{1/2} \begin{bmatrix} \mathbf{G}^\top \mathbf{M}^\top \mathbf{B}^\top \mathbf{a}_i \\ \mathbf{G}^\top \mathbf{D}^\top \mathbf{a}_i \end{bmatrix} \right\|_2 .$$

This expression can be substituted into (7.21g), which renders Problem 7.2 a convex second-order cone program for fixed $\boldsymbol{\epsilon}$.

Iterative Optimization Strategy

The optimal control problem in Problem 7.2 can be solved by optimizing both the risk allocation $\boldsymbol{\epsilon}$ and the control feedback gain $\mathbf{M}$. An iterative two-stage optimization strategy is presented in [257] to bisect the uniform risk allocation in the upper stage and to optimize the feedback gain with fixed uniform risk allocation in the lower stage. On the other hand, [152] proposed optimizing the risk allocation and feedback gain simultaneously using a tailored interior point method that exploits the sparse

Algorithm 7.1 Coordinate descent for SMPC

Require: Initial feedback gain $\mathbf{M}^{(0)}$ and maximum number of iterations $I_{\max}$.

1: **for** $i = 0$ to $I_{\max} - 1$ **do**
2: Solve convex optimization Problem 7.2 with fixed $\mathbf{M} \leftarrow \mathbf{M}^{(i)}$ for the optimal risk allocation $\boldsymbol{\epsilon}^\star$
3: Set $\boldsymbol{\epsilon}^{(i+1)} \leftarrow \boldsymbol{\epsilon}^\star$
4: Solve convex optimization Problem 7.2 with fixed $\boldsymbol{\epsilon} \leftarrow \boldsymbol{\epsilon}^{(i+1)}$ for the optimal feedback gain $\mathbf{M}^\star$
5: Set $\mathbf{M}^{(i+1)} \leftarrow \mathbf{M}^\star$
6: **end for**

multistage structure of the nonconvex optimization. Although these approaches were developed under different disturbance assumptions and control law parametrizations, they can be applied for solving Problem 7.2 owing to the similar structure of the optimization problems.

In this work, a simple iterative approach is proposed for solving Problem 7.2, as summarized in Algorithm 7.1. The primary notion of Algorithm 7.1 is to solve for the optimal risk allocation given a fixed feedback gain and then solve for the optimal feedback gain given a fixed risk allocation. This approach is similar to the well-known DK iteration used in μ-synthesis problems [8]. This technique is known as a (block-)coordinate descent algorithm, and has been applied more broadly to optimization problems subject to bilinear matrix inequality (BMI) constraints [231]. Although this algorithm is not guaranteed to converge to a local optimum (as each iteration provides a solution that is optimal in the "directions" of one subset of variables, but not in all directions), it is a commonly applied heuristic that performs well in practice.

Two choices have been made in Algorithm 7.1: (i) initializing the algorithm with a fixed feedback gain $\mathbf{M}^{(0)}$ (instead of a fixed risk allocation $\boldsymbol{\epsilon}^{(0)}$ and switching the order of the optimization problems), and (ii) running the algorithm for a fixed number of iterations instead of running until a prespecified tolerance has been met. The initial feedback gain $\mathbf{M}^{(0)}$ can be designed optimally without explicitly considering constraints using any of the numerous existing robust control methods, e.g., [131]. Since there has been a plethora of work on offline feedback control design, initializing the algorithm based on a nearly optimal feedback gain is likely to yield better perfor-

mance than initializing the algorithm using a fixed uniform risk allocation, which will rarely be optimal in practice. In addition, since adequate closed-loop performance can often be obtained with just a few iterations from a near optimal choice of $\mathbf{M}^{(0)}$, it is best to run Algorithm 7.1 for a fixed number of iterations so as to ensure that the control inputs can be computed within a reasonable computation time. This idea has been widely used in the fast MPC literature to significantly reduce the cost of solving MPC problems online as discussed briefly in Chapter 4 and in [261].

Feasibility and Stability Considerations

Due to the inclusion of input and state constraints, the region of attraction $\mathscr{X}_N$ for Problem 7.2 (i.e., the set of initial conditions for which there exists a feasible solution to the optimization problem) will be a subset of $\mathbb{R}^n$. In the robust MPC literature, feasibility is commonly addressed by ensuring that the states remain in $\mathscr{X}_N$ at all times upon entering $\mathscr{X}_N$. When the disturbances lie in a compact set, recursive feasibility (as well as closed-loop stability) of the MPC problem can be guaranteed by defining terminal constraints and/or terminal penalties [161, 89].

The proposed SMPC approach, however, considers arbitrary stochastic disturbances with a (possibly) unbounded support. Hence, it is impractical to ensure that the states remain inside $\mathscr{X}_N$ in the presence of input constraints [43]. One approach for guaranteeing recursive feasibility for SMPC problems with unbounded disturbances is to choose between a closed-loop and open-loop initialization strategy online [67]. The key idea in this approach is to choose the closed-loop strategy when the problem is feasible and to choose the open-loop strategy (whose feasibility is guaranteed through a proper selection of terminal constraints) when the SMPC problem is infeasible for the most recently observed states. Although this approach guarantees recursively feasibility, it completely ignores the most recent state measurements, which will degrade closed-loop performance when the states are not in the region of attraction of the controller.

Alternatively, a backup controller can be applied when the states leave the region of attraction of Problem 7.2. In this case, a natural choice is to soften the state

constraints in Problem 7.2, as this will enable driving the states back into $\mathscr{X}_N$ [192]. To this end, the *exact penalty function* method can be used to ensure that the backup controller yields the same solution as the fully constrained MPC problem when it is feasible [122]. This approach allows for solving a single optimization instead of having to verify feasibility and decide which MPC problem to solve accordingly.

Stability of stochastic linear systems (in a mean-square boundedness sense) in the presence of unbounded disturbances and bounded control inputs has been explored extensively in [44]. If the eigenvalues of the system matrix A lie inside the unit disc, the variance of the states is shown to be bounded as long as the disturbance has bounded variance. When A has eigenvalues on the unit disc (with equal geometric and algebraic multiplicities), the variance of states will be bounded provided that $\|u\|_2 \leq R$ for a large enough R. However, if A has even one unstable eigenvalue and the system is subjected to unbounded stochastic disturbances along the directions of the unstable eigen-subspace of A, the linear system cannot be stabilized by means of bounded control inputs [44]. These results are readily inherited by the proposed SMPC approach (Algorithm 7.1).

7.6 Example: Continuous Bioreactor Process

Bioreactor Model The performance of the proposed SMPC approach is evaluated on a continuous *clostridial* acetone-butanol-ethanol (ABE) fermentation. The model of [98] is linearized around a desired steady-state operating point to obtain the system description (7.1) consisting of 12 states and 2 inputs. All system states are perturbed by arbitrary unbounded disturbances with known mean and variance (see Appendix A for the system description).

The system description, in terms of the notation used throughout this chapter is as follows. The state vector for the continuous ABE fermentation process is defined by

$$x = \left[C_{AC},\ C_A,\ C_{En},\ C_{AaC},\ C_{Aa},\ C_{BC},\ C_B,\ C_{An},\ C_{Bn},\ C_{Ad},\ C_{Cf},\ C_{Ah}\right]^\top,$$

where C denotes concentration (mM) of AC = Acetyl-CoA, A = Acetate, En = Ethanol, AaC = Acetoacetate-CoA, Aa = Acetoacetate, BC = Butyryl-CoA, B = Butyrate, An = Acetone, Bn = Butanol, Ad = adc, Cf = ctfA/B, and Ah = adhE [98]. The input vector is defined as $u = [D\ G_0]^\top$, where D is the dilution rate (hr^{-1}) and G_0 is the inlet glucose concentration (mM). The system matrices are

$$A = 10^{-2}\begin{bmatrix} 51 & 5.3 & 0 & 29 & 0 & 0 & -2.7 & 0 & 0 & 0 & 3.4 & -10 \\ -2.5 & 85 & 0 & -43 & 0 & 0 & 3.4 & 0 & 0 & 0 & -5.1 & -0.014 \\ 37 & 1.8 & 93 & 12 & 0 & 0 & -0.59 & 0 & 0 & 0 & 1.4 & 1.1 \\ 3.3 & -4.5 & 0 & 18 & 0 & 0 & -5.0 & 0 & 0 & 0 & -8.8 & -0.036 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.030 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2.1 & 2.6 & 0 & -35 & 0 & 0 & 85 & 0 & 0 & 0 & -4.1 & 0.012 \\ 4.6 & 4.9 & 0 & 78 & 93 & 0 & 4.9 & 93 & 0 & 0.030 & 9.1 & -0.027 \\ 2.1 & -2.6 & 0 & 35 & 0 & 93 & 8.3 & 0 & 93 & 0 & 4.1 & -0.012 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 93 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 93 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 93 \end{bmatrix},$$

$$B = \begin{bmatrix} -1.7 & 5.7 \times 10^{-5} \\ -13 & -7.9 \times 10^{-7} \\ -7.8 & 1.6 \times 10^{-5} \\ 0.94 & 2.0 \times 10^{-6} \\ 0 & 0 \\ 0 & 0 \\ -10 & -6.8 \times 10^{-7} \\ -45 & 1.5 \times 10^{-6} \\ -51 & 6.8 \times 10^{-7} \\ -1.4 & 0 \\ -14 & 0 \\ -37 & 0 \end{bmatrix},$$

and

$$G = \text{diag}(0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0).$$

Control Problem The control problem is formulated in terms of setpoint tracking for the ABE products while satisfying hard constraints on both inputs as well as a joint chance constraint on acetate and butyrate (i.e., two of the key intermediate species in the metabolic pathway). Algorithm 7.1 is used to iteratively solve the deterministic surrogate for the SMPC Problem 7.2 for determining the optimal feedback control policy and risk allocation (see Table 7.1 for the parameter settings used in this example). The performance of the proposed approach is compared to that of a certainty equivalence MPC algorithm (in which the disturbance is set equal to its expected value for the purposes of prediction) and a MPC algorithm with fixed uniform risk allocation. The fixed gain optimization problem is solved using IPOPT, whereas the CVX package with the Mosek solver is utilized to solve the fixed risk allocation optimization problem [91].

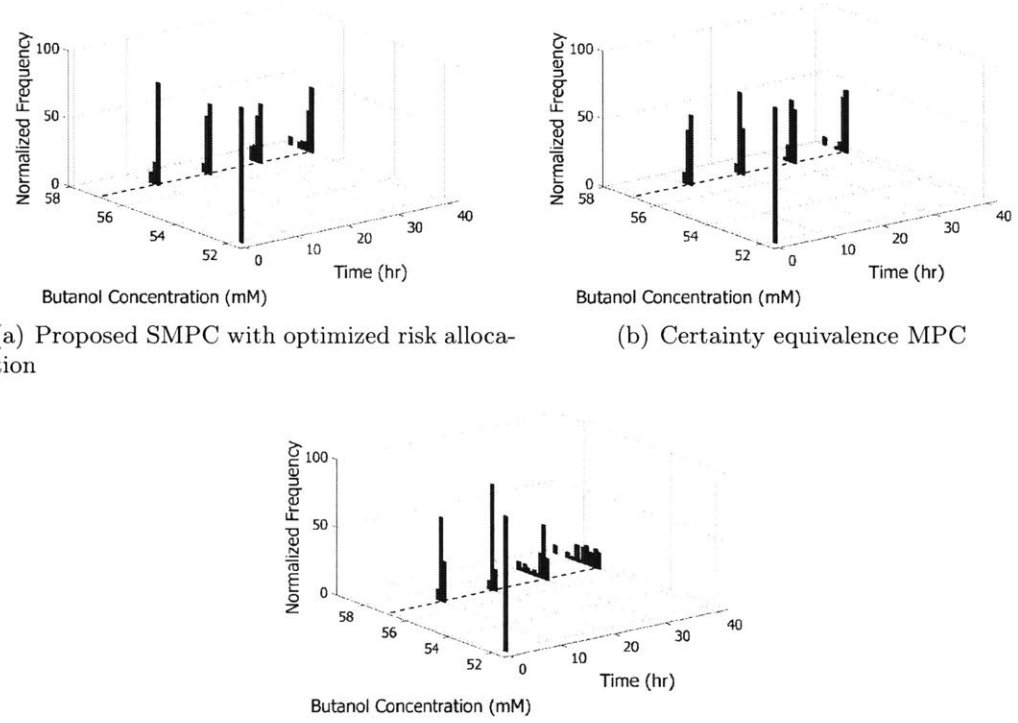
Table 7.1: Settings of the SMPC problem in the continuous bioreactor case study.

Sampling time	1 hr
N	10
Q	diag(0,0,0.01,0,0,0,0,0.01,10,0,0,0)
R	diag(0.1,0.1)
State constraints	13.83 mM $\leq C_A \leq$ 15.68 mM 10.55 mM $\leq C_B \leq$ 12.30 mM
Hard input constraints	0.005 $\text{hr}^{-1} \leq D \leq$ 0.145 hr^{-1} 0 mM $\leq G_0 \leq$ 80 mM

Results Figure 7-1 shows the probability distribution of butanol at times 1, 10, 20, 30, and 40 hr in the case of a butanol setpoint change of 10% applied at the beginning of the process. The histograms in Figure 7-1 are constructed based on 50 Monte Carlo simulation runs under identical disturbance realizations for the three control algorithms. The proposed SMPC approach and the certainty equivalence MPC algorithm show comparable performance in terms of minimizing the variations in butanol concentration around the setpoint. The MPC algorithm with fixed uniform risk allocation resulted in the worst performance in terms of large variance in the butanol concentration. The poor performance of the MPC algorithm with fixed uniform risk allocation can be attributed to its conservative state constraint handling as it attempts to fulfill the individual state chance constraints (decomposed from the joint chance constraint) with equal risk regardless of the likelihood of their violation.

Figure 7-2 shows the *joint* probability distribution of acetate and butyrate at time 1 hr; the process exhibits maximum state constraint violation at this time point. The proposed SMPC approach results in a joint state constraint violation of less than 8%, which is below the prespecified admissible joint constraint violation level of 20%. Figure 7-2 indicates that the proposed approach significantly outperforms the equivalence MPC algorithm in terms of state constraint handling, as the latter algorithm gives rise to 78% constraint violations. The MPC algorithm with fixed uniform risk allocation exhibits a constraint violation level of approximately 14%, implying a less effective constraint handling than the proposed SMPC approach with optimized risk allocation. Overall, the proposed SMPC approach led to the best

control performance in terms of tracking the setpoint while dealing with the joint state chance constraint.



(a) Proposed SMPC with optimized risk allocation

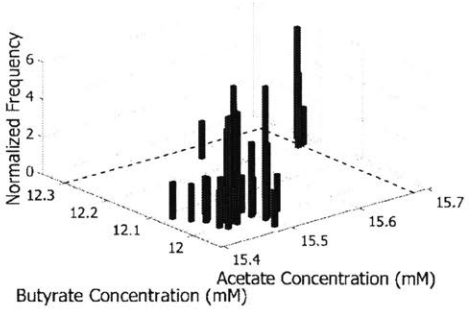
(b) Certainty equivalence MPC

(c) MPC with fixed uniform risk allocation

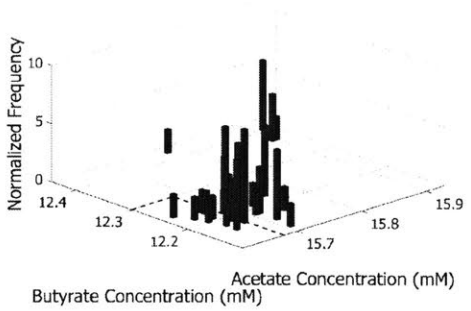
Figure 7-1: Probability distributions of butanol at process times 1, 10, 20, 30, and 40 hr; the black dashed line shows the setpoint. A 10% change in the butanol setpoint is applied at time 0 hr.

7.7 Conclusions

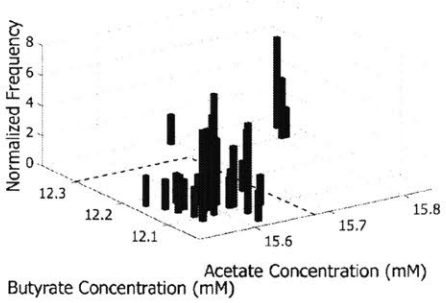
This chapter presents a MPC approach for linear systems subject to arbitrary (possibly unbounded) stochastic disturbances with known mean and variance. The approach enables: (i) accounting for hard input constraints and joint state chance constraints under closed-loop prediction, (ii) efficient handling of joint chance constraints by using the Cantelli-Chebyshev inequality in conjunction with risk allocation, and (iii) determining the optimal feedback gain and risk allocation by iteratively solving convex optimizations. Additionally, feasibility and stability properties of the proposed



(a) Proposed SMPC with optimized risk allocation



(b) Certainty equivalence MPC



(c) MPC with fixed uniform risk allocation

Figure 7-2: Joint probability distribution of acetate and butyrate at process time 1 hr; the black dashed lines show the bounds of the state constraints.

MPC method are briefly summarized. The performance of the method is evaluated using a continuous bioreactor (used for the production of high value-added drop-in biofuels from biomass) case study

Chapter 8

Stability in Stochastic Receding Horizon Control

8.1 Introduction

Robust model predictive control (MPC) approaches have been extensively investigated over the last two decades with the goal to address control of uncertain systems with bounded uncertainties (e.g., see [14] and the references therein). Robust MPC approaches rely on a deterministic setting and set-based uncertainty descriptions to synthesize controllers such that a worst-case objective is minimized or constraints are robustly satisfied [29]. These deterministic approaches may however lead to overly conservative control performance [14] if the worst-case realizations have a small probability of occurrence. An approach that can alleviate the intrinsic limitation of a deterministic robust control setting is to use stochastic descriptions of system uncertainties, which are available in many applications. This notion has led to the emerging field of stochastic MPC (SMPC) (e.g., [225, 103, 52, 42, 20, 27, 193, 43, 38, 169]), in which probabilistic descriptions of uncertainties and chance constraints are used to allow for pre-specified levels of risk in optimal control.

This chapter investigates stability of SMPC. There is extensive literature that deals with tractability and stability of MPC in the deterministic setting (e.g., see [14, 161] and the references therein). However, the technical nature of arguments involved

in stability of stochastic systems is significantly different in the case of unbounded uncertainties, such as Gaussian process noise. In addition, there exist diverse notions of stability in the stochastic setting that are non-existent in the deterministic case [43].

The work on stability of uncertain systems under receding-horizon stochastic optimal control can be broadly categorized into two research directions: first, studies that consider multiplicative process and measurement noise [52, 42, 208] and second, studies that treat process and measurement noise as additive terms in the system model [16, 23, 193, 105]. The latter approaches mainly rely on the notion of affine parameterization of control inputs for finite-horizon linear quadratic problems, which allows converting the stochastic programming problem into a deterministic one. Other approaches to SMPC based on randomized algorithms [20, 27, 38] and SMPC formulations with chance constraints [67] have also been reported.

Organization In this chapter, a SMPC problem is presented for discrete-time linear systems with arbitrarily-shaped probabilistic time-invariant uncertainties and additive Gaussian process noise (Section 8.2). Chance constraints are incorporated into the SMPC formulation to seek tradeoffs between control performance and robustness to uncertainties. To obtain a deterministic surrogate for the posed SMPC problem, the individual chance constraints are converted into deterministic expressions in terms of the mean and variance of the system states and a state feedback paramterization of the control law is applied (Section 8.3). This work uses the generalized polynomial chaos (gPC) framework [266, 81, 272] for probabilistic uncertainty propagation through the system dynamics in order to obtain a computationally tractable formulation for the proposed SMPC problem (Section 8.4). The Galerkin-projection method [81] is used for analytic computation of the coefficients of the series, based on which the state's statistics can be computed in a computationally efficient manner. Inspired by the stability results for Markov processes [170], the closed-loop stability of the stochastic system is established by appropriate selection of the cost function in the unconstrained case. It is proven that the proposed SMPC approach ensures closed-

loop stability *by design* under the corresponding receding-horizon control strategy (Section 8.5). This algorithm is applied to control a continuously stirred-tank reactor that is carrying out the Van de Vusse series of reactions (Section 8.6).

The majority of this chapter was published in the *Proceedings of the American Control Conference* [202].

Notation Hereafter, $\mathbb{N} = \{1, 2, \ldots\}$ is the set of natural numbers; $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$; $\mathbb{R}_{\geq 0}$ is the set of nonnegative real numbers; $\mathbb{Z}_{[a,b]} := \{a, a+1, \ldots, b\}$ is the set of integers from a to b; I_a is the $a \times a$ identity matrix; $\mathbf{1}_a$ is a a-dimensional vector of ones; $\mathbf{I}_a$ is the $a \times a$ all-ones matrix; $\mathcal{I}_A(\cdot)$ denotes the indicator function of the set A; $\mathbb{E}[\cdot]$ or $\bar{(\cdot)}$ is the expected value; $\mathbb{E}[\cdot|x]$ is the conditional expected value given information x; $\mathbf{Var}[\cdot]$ is the covariance matrix; $\mathbb{P}(\cdot)$ denotes probability; $\mathcal{N}(\mu, \Sigma)$ is the Gaussian distribution with mean μ and covariance Σ; $\otimes$ is the Kronecker product; $\circ$ is the Hadamard (entrywise) product; $\mathrm{tr}(\cdot)$ is the trace of a square matrix; $\|x\|_A^2 := x^\top A x$ is the weighted 2-norm; and $\mathrm{vec}(\cdot)$ denotes the column vectorization.

8.2 Problem Formulation

Consider a stochastic, discrete-time linear system

$$x^+ = A(\theta)x + B(\theta)u + Fw, \tag{8.1}$$

where $x \in \mathbb{R}^{n_x}$ is the system state at the current time instant; x^+ is the state at the next time instant; $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ denotes the system inputs, with $\mathbb{U}$ being a nonempty set of input constraints that is assumed to contain the origin; $\theta \in \mathbb{R}^{n_\theta}$ denotes the time-invariant uncertain system parameters with known probability distribution functions (PDFs) $f(\theta)$; and $w \sim \mathcal{N}(0, \Sigma) \in \mathbb{R}^{n_w}$ denotes a normally distributed i.i.d. stochastic disturbance with known covariance $\Sigma \in \mathbb{R}^{n_w \times n_w}$. It is assumed that the pair (A, B) is stabilizable for all uncertainty realizations θ, and that x can be observed exactly at any time.

We include individual state chance constraints, that must be satisfied for all time,

of the form

$$\mathbb{P}(x \in \mathbb{X}_i) \geq \beta_i, \quad i = 1, \ldots, n_{cc}, \tag{8.2}$$

where $\mathbb{X}_i := \{x \in \mathbb{R}^{n_x} | c_i^\top x \leq d_i\}$; $c_i \in \mathbb{R}^{n_x}$; $d_i \in \mathbb{R}$; n_{cc} denotes the number of chance constraints considered; and $\beta_i \in (0, 1)$ denotes the lower bound of the desired probability that the i^{th} state constraint should be satisfied in the presence of system uncertainties.

This chapter aims to design an SMPC approach for the system (8.1) such that the stability of the closed-loop system is guaranteed. The SMPC approach incorporates the statistical descriptions of system uncertainties into the control framework. Such a probabilistic control approach allows the shaping of the state PDFs, which is essential to trade off the performance and robustness of the closed-loop system.

Let $N \in \mathbb{N}$ denote the prediction horizon of the control problem, and define $\mathbf{w} := [w_0^\top, \ldots, w_{N-1}^\top]^\top$ as the disturbance sequence over 0 to $N-1$. We consider a full state feedback control policy π defined by

$$\pi := \{\pi_0, \pi_1(\cdot), \ldots, \pi_{N-1}(\cdot)\} \tag{8.3}$$

where $\pi_0 \in \mathbb{U}$ is a control action that is a function of the known current state and $\pi_i(\cdot) : \mathbb{R}^{n_x} \longrightarrow \mathbb{U}$ are feedback control laws for $i = 1, \ldots, N-1$.

Let $\phi_i(x, \pi, \mathbf{w}, \theta)$ denote the solution to (8.1) at time i when the initial state is x at time 0, the control law π_j is applied at time $j = 0, \ldots, i-1$, the disturbance realization is $w_0, \ldots, w_{i-1}$, and the parameter realization is θ, i.e.,

$$\phi_{i+1} = A(\theta)\phi_i + B(\theta)\pi_i(\phi_i) + w_i \tag{8.4}$$

where $\phi_0 = x$. Note $\{\phi_i(x, \pi, \mathbf{w}, \theta)\}_{i=0}^N$ represent model predictions from the observed state x, and that the explicit functional dependencies on the initial condition, control law, and uncertainties will be dropped for notational convenience.

We can now formulate the SMPC problem for the stochastic linear system (8.1)

with time-invariant parametric uncertainty and unbounded process noise as follows.

Problem 8.1 (SMPC with hard input and state chance constraints) *Given the current states x, observed from system* (8.1)*, the stochastic optimal control problem to be solved at each time instance is defined as*

$$
\begin{aligned}
J_N^*(x) := \min_{\pi} \quad & J_N(x,\pi), && && (8.5)\\
\text{s.t.} \quad & \text{dynamics } (8.4), && i \in \mathbb{Z}_{[0,N-1]},\\
& \pi_i \in \mathbb{U}, && i \in \mathbb{Z}_{[0,N-1]},\\
& \mathbb{P}(c_l^\top \phi_i \leq d_l) \geq \beta_l, && i \in \mathbb{Z}_{[1,N-1]},\ l \in \mathbb{Z}_{[1,n_{cc}]},\\
& \phi_0 = x,\ \theta \sim f(\theta),\ \mathbf{w} \sim \mathcal{N}(0, \Sigma_{\mathbf{w}}),
\end{aligned}
$$

where the objective function is defined as

$$
J_N(x,\pi) = \mathbb{E}\left[\sum_{i=0}^{N-1} \|\phi_i\|_Q^2 + \|\pi_i\|_R^2\right], \tag{8.6}
$$

Q and R are symmetric positive definite weighting matrices, and $\Sigma_{\mathbf{w}} = \text{diag}(\Sigma, \cdots, \Sigma)$.

Note that since the system is time-invariant, we are able to define the observed state x (at any time) as the initial state at time 0 for model prediction in the SMPC program. Problem 8.1 cannot be solved directly for three main reasons:

- Cannot optimize over arbitrary functions π.
- Chance constraints are non-convex and intractable.
- Traditional methods for propagating time-invariant uncertainty are inefficient.

In this work, approximations are introduced to Problem 8.1 to tackle the aforementioned issues. First, we introduce a state feedback parametrization of the control policy. Next, the chance constraints are replaced with a deterministic surrogate, in terms of the mean and variance of the predicted state, to yield a tractable expression. These approximations are introduced in Section 8.3 to produce a deterministic version of Problem 8.1

However, the problem is still not yet fully tractable due to the presence of time-invariant uncertainties θ. Generalized polynomial chaos (gPC) is proposed for efficient propagation of these probabilistic uncertainties through the system dynamics. The Galerkin projection, coupled with gPC, directly provides an approximation of the moments of the predicted state ϕ_i that appear in the reformulated chance constraints and objective function of the SMPC problem (Section 8.4). Stability of the unconstrained version of this approximated SMPC problem is then explored in Section 8.5 using methods for Markov processes.

8.3 Deterministic Surrogate

Approximation of Chance Constraints

We use the following result to replace chance constraints in (8.5) with a deterministic expression in terms of the mean and variance of the predicted states.

Theorem 8.1 (Distributionally robust chance constraint [37]) *Consider an individual chance constraint of the form*

$$\mathbb{P}(c^\top l \leq 0) \geq 1 - \beta, \quad \beta \in (0,1), \tag{8.7}$$

where $l \in \mathbb{R}^{n_l}$ are some random quantities with known mean $\tilde{l}$ and covariance Σ_l and $c \in \mathbb{R}^{n_l}$ are some constants. Let $\mathcal{L}$ denote the family of all distributions with mean $\tilde{l}$ and covariance Σ_l. For any $\beta \in (0,1)$, the chance constraint

$$\inf_{l \sim \mathcal{L}} \mathbb{P}(c^\top l \leq 0) \geq 1 - \beta,$$

(where $l \sim \mathcal{L}$ denotes that the distribution of l belongs to the family $\mathcal{L}$) is equivalent to the constraint

$$\mathbb{E}[c^\top l] + \kappa_\beta \sqrt{\mathbf{Var}[c^\top l]} \leq 0, \quad \kappa_\beta = \sqrt{(1-\beta)/\beta}, \tag{8.8}$$

where $\mathbb{E}[c^\top l] = c^\top \tilde{l}$ *and* $\mathbf{Var}[c^\top l] = c^\top \Sigma_l c$.

Using Theorem 8.1, we can replace the chance constraints in (8.5) with a deterministic counterpart

$$c_l^\top \mathbb{E}[\phi_i] + \kappa_{1-\beta_l} \sqrt{c_l^\top \mathbf{Var}[\phi_i] c_l} \leq d_l \tag{8.9}$$

which guarantees the constraint $\phi_i \in \mathbb{X}_l$ is satisfied with at least probability β_l.

State Feedback Parametrization of Control Policy

To incorporate feedback over the prediction, we choose to have the control policy parametrized as an affine function of the state. This leads to policy π having elements of the form

$$\pi_i(\phi_i) := g_i + L_i \phi_i, \quad i \in \mathbb{Z}_{[0,N-1]} \tag{8.10}$$

where $g_i \in \mathbb{R}^{n_u}$ and $L_i \in \mathbb{R}^{n_u \times n_x}$ are the affine terms and feedback gains, respectively.

Let $\tilde{L} = \{L_0, \ldots, L_{N-1}\}$ and $\tilde{g} = \{g_0, \ldots, g_{N-1}\}$ denote the set of set of decision variables in (8.5) to be optimized over the horizon N. A deterministic reformulation of Problem 8.1 is then stated as

Problem 8.2 (Deterministic formulation for SMPC with hard input and state chance constraints)

$$\begin{aligned}
&\min_{(\tilde{L},\tilde{g})} \; J_N(x, \tilde{L}, \tilde{g}), && && (8.11)\\
&\text{s.t.} \quad \phi_{i+1} = A(\theta)\phi_i + B(\theta)\pi_i + w_i, && i \in \mathbb{Z}_{[0,N-1]},\\
&\qquad \pi_i = g_i + L_i \phi_i \in \mathbb{U}, && i \in \mathbb{Z}_{[0,N-1]},\\
&\qquad c_l^\top \mathbb{E}[\phi_i] + \kappa_{1-\beta_l} \sqrt{c_l^\top \mathbf{Var}[\phi_i] c_l} \leq d_l, && i \in \mathbb{Z}_{[1,N-1]},\\
&\qquad && l \in \mathbb{Z}_{[1,n_{cc}]},\\
&\qquad \phi_0 = x, \; \theta \sim f(\theta), \; \mathbf{w} \sim \mathcal{N}(0, \Sigma_{\mathbf{w}}),
\end{aligned}$$

In Problem 8.2, the objective function and chance constraints are only in terms of

the mean and variance of the predicted states ϕ_i. Using the gPC framework, we can propagate uncertainties θ and $\mathbf{w}$ through the system dynamics to approximate these moments using deterministic equations.

Remark 8.1 *In general, it is impossible to guarantee input constraint satisfaction for a state feedback control law in the presence of unbounded disturbances unless $L_i = 0$ for all $i = 0, \cdots, L-1$, meaning that (8.10) takes the form of an open-loop control law. This concept was further elaborated on in Chapter 7 where a method was introduced for handling unbounded disturbances within SMPC. Those methods can also be utilized here, but for simplicity we do not consider hard input constraints for the remainder of the chapter by assuming $\mathbb{U} = \mathbb{R}^{n_u}$.*[1]

8.4 Tractable Stochastic Model Predictive Control Algorithm

Polynomial Chaos for Uncertainty Propagation

The gPC framework enables approximation of a stochastic variable $\psi(\xi)$ in terms of a finite series expansion of orthogonal polynomial basis functions. A detailed overview of gPC and its relationship to Weiner–Hermite chaos is provided in Chapter 3 of this thesis. A brief review is provided here for clarity and to introduce the notation used in this chapter:

$$\psi(\xi) \approx \hat{\psi}(\xi) := \sum_{k=0}^{p} a_k \varphi_k(\xi) = \mathbf{a}^\top \Lambda(\xi), \tag{8.12}$$

where $\mathbf{a} := [a_0, \ldots, a_p]^\top$, $\Lambda(\xi) := [\varphi_0(\xi), \ldots, \varphi_p(\xi)]^\top$ is the vector of φ_k of maximum degree m with respect to the random variables ξ, and $p + 1 = \frac{(n_\xi + m)!}{n_\xi! m!}$ denotes the total number of terms in the expansion. The basis functions belong to the Askey scheme of polynomials, which encompasses a set of orthogonal basis functions in the

[1] Alternative approaches include truncating the distribution of the disturbances to have a finite support or defining input chance constraints (see, e.g., [67]

Hilbert space defined on the support of the random variables [272]. This implies that $\langle \varphi_i(\xi), \varphi_j(\xi) \rangle = \langle \varphi_i^2(\xi) \rangle \delta_{ij}$, where $\langle h(\xi), g(\xi) \rangle = \int_\Omega h(\xi) g(\xi) f(\xi) d\xi$ denotes the inner product induced by $f(\xi)$, and δ_{ij} denotes the Kronecker delta function. Hence, the coefficients a_k in (8.12) are defined by $a_k = \frac{\langle \psi(\xi), \varphi_k(\xi) \rangle}{\langle \varphi_k(\xi), \varphi_k(\xi) \rangle}$.

For linear and polynomial systems, the integrals in the inner products can be computed analytically [81]. Note that the basis functions φ_k are chosen in accordance with the PDFs of the uncertain variables ξ as discussed in Chapter 3.

Evaluation of Multivariate State PDF

The time evolution of the multivariate predicted state PDF, given $\phi_0 = x$, describes the propagation of θ and $\{w_i\}$ through the system dynamics (8.4). For a particular realization of $\{w_i\}$, the propagation of θ through (8.4) can be efficiently described using gPC. This represents the conditional predicted state PDF $f(\phi_{i+1}|w_0, \cdots, w_i)$, which can be integrated over all possible realizations of $w_0, \cdots, w_i$ to obtain the entire predicted state PDF at the next time $f(\phi_{i+1})$, i.e.,

$$f(\phi_{i+1}) = \int_{-\infty}^{\infty} f(\phi_{i+1}|w_0, \cdots, w_i) \prod_{s=0}^{i} f(w_s) dw_s, \tag{8.13}$$

Since $f(w_i)$ for all $i \in \mathbb{Z}_{[0,N-1]}$ is a Gaussian distribution, this integral reduces substantially when evaluating moments of the states, as shown later in this section.

To use the gPC approach, approximate each element of the predicted state ϕ, input π, $A(\theta)$, and $B(\theta)$ in (8.4) with a finite PC expansion of the form (8.12). Define $\Phi_{i,t} = [a_{i_0,t}, \ldots, a_{i_p,t}]^\top$ and $\Pi_{i,t} = [b_{i_0,t}, \ldots, b_{i_p,t}]^\top$ to be the set of PC expansion coefficients for the i^{th} predicted state and input at time t, respectively, and then concatenate these into vectors $\mathbf{\Phi}_t := [\Phi_{1,t}^\top, \ldots, \Phi_{n_x,t}^\top]^\top \in \mathbb{R}^{n_x(p+1)}$ and $\mathbf{\Pi}_t := [\Pi_{1,t}^\top, \ldots, \Pi_{n_u,t}^\top]^\top \in \mathbb{R}^{n_u(p+1)}$. Then, use the Galerkin projection method, as described in [199], to project the error in the truncated expansion approximation of (8.4) onto the space of orthogonal basis

functions $\{\varphi_k\}_{k=0}^p$ to yield

$$\mathbf{\Phi}_{i+1} = \mathbf{A}\mathbf{\Phi}_i + \mathbf{B}\mathbf{\Pi}_i + \mathbf{F}w_i, \tag{8.14}$$

where

$$\mathbf{A} = \sum_{k=0}^{p} A_k \otimes \Psi_k, \ \mathbf{B} = \sum_{k=0}^{p} B_k \otimes \Psi_k, \ \mathbf{F} = F \otimes e_{p+1},$$

$$\Psi_k := \begin{bmatrix} \sigma_{0k0} & \cdots & \sigma_{0kp} \\ \vdots & \ddots & \vdots \\ \sigma_{pk0} & \cdots & \sigma_{pkp} \end{bmatrix},$$

A_k and B_k are the projections of $A(\theta)$ and $B(\theta)$ onto the k^{th} basis function φ_k, $\sigma_{ijk} = \langle \varphi_i, \varphi_j, \varphi_k \rangle / \langle \varphi_i^2 \rangle$, and $e_a = [1, 0, \ldots, 0]^\top \in \mathbb{R}^a$ is an a-dimensional vector whose first element is one and the remaining elements are zero.

We can take advantage of the orthogonality property of the multivariate polynomials to efficiently compute moments of the conditional PDF $P(\phi_{t+1}|\{w_s\}_{s=0}^t)$ using the coefficients $\mathbf{\Phi}_{t+1}$. For example, here are the explicit expressions for the first two moments of the i^{th} predicted state

$$\mathbb{E}[\phi_{i,t+1}|w_0, \cdots, w_t] \approx a_{i_0,t+1}(w_0, \ldots, w_t), \tag{8.15a}$$

$$\mathbb{E}\left[\phi_{i,t+1}^2|w_0, \cdots, w_t\right] \approx \sum_{k=0}^{p} a_{i_k,t+1}^2(w_0, \ldots, w_t)\langle \varphi_k^2 \rangle, \tag{8.15b}$$

Similarly, we can project the state feedback control law (8.10)

$$\mathbf{\Pi}_i = \mathbf{g}_i + \mathbf{L}_i\mathbf{\Phi}_i, \tag{8.16}$$

where $\mathbf{g}_i = g_i \otimes e_{p+1}$ and $\mathbf{L}_i = L_i \otimes I_{p+1}$. Since $w_0 \cdots, w_{N-1}$ is assumed to be Gaussian white noise, $\{\mathbf{\Phi}_i\}_{i=1}^N$ is a Gaussian process, with mean $\bar{\mathbf{\Phi}}_i$ and covariance Γ_i given by

$$\bar{\mathbf{\Phi}}_{i+1} = (\mathbf{A} + \mathbf{B}\mathbf{L}_i)\bar{\mathbf{\Phi}}_i + \mathbf{B}\mathbf{g}_i, \tag{8.17a}$$

$$\Gamma_{i+1} = (\mathbf{A} + \mathbf{B}\mathbf{L}_i)\Gamma_i(\mathbf{A} + \mathbf{B}\mathbf{L}_i)^\top + \mathbf{F}\Sigma\mathbf{F}^\top \tag{8.17b}$$

We can initialize $(\bar{\mathbf{\Phi}}_i, \Gamma_i)$ using the current state $\phi_0 = x$ via projection, i.e., $\bar{\mathbf{\Phi}}_0 = x \otimes e_{p+1}$ and $\Gamma_0 = 0$. Using (8.15), (8.17), and the law of iterated expectation, we can derive tractable expressions for first two moments of $f(\phi_{i,t+1})$

$$\begin{aligned} \mathbb{E}[\phi_{i,t+1}] &= \mathbb{E}\big[\mathbb{E}[\phi_{i,t+1}|w_0, \cdots, w_t]\big], \\ &\approx \mathbb{E}[a_{i_0,t+1}(w_0, \ldots, w_t)], \\ &= \bar{a}_{i_0,t+1}, \end{aligned} \tag{8.18}$$

$$\begin{aligned} \mathbb{E}[\phi^2_{i,t+1}] &= \mathbb{E}\big[\mathbb{E}[\phi^2_{i,t+1}|w_0, \cdots, w_t]\big], \\ &\approx \mathbb{E}\big[\sum_{k=0}^{p} a^2_{i_k,t+1}(w_0, \ldots, w_t)\langle\varphi^2_k\rangle\big], \\ &= \sum_{k=0}^{p} \mathbb{E}[a^2_{i_k,t+1}]\langle\varphi^2_k\rangle, \\ &= \sum_{k=0}^{p} \big[\bar{a}^2_{i_k,t+1} + \Gamma_{i_k i_k,t+1}\big]\langle\varphi^2_k\rangle, \end{aligned} \tag{8.19}$$

Tractable SMPC Formulation using gPC

In this section, our goal is to use gPC to write a tractable approximation of Problem (8.11). As discussed previously, Problem (8.11) is written in terms of moments of $\{\phi_i\}_{i=0}^N$. Therefore, we can use (8.18) and (8.19) to rewrite J_N and the chance constraints in terms of $(\bar{\mathbf{\Phi}}_i, \Gamma_i)$. First, rewrite (8.6) as

$$J_N = \mathbb{E}\left[\sum_{i=0}^{N-1} \mathbb{E}[\|\phi_i\|^2_Q | w_0, \cdots, w_{i-1}] + \mathbb{E}[\|\pi_i\|^2_R | w_0, \cdots, w_{i-1}]\right],$$

and approximate the conditional moments using gPC (8.15)

$$J_N \approx V_N := \mathbb{E}\left[\sum_{i=0}^{N-1} \|\mathbf{\Phi}_i\|_{\mathbf{Q}}^2 + \|\mathbf{\Pi}_i\|_{\mathbf{R}}^2\right]. \tag{8.20}$$

where $\mathbf{Q} = Q \otimes W$, $\mathbf{R} = R \otimes W$, and $W = \text{diag}(\langle\varphi_0^2\rangle, \langle\varphi_1^2\rangle, \ldots, \langle\varphi_p^2\rangle)$. Substituting the propagated control law (8.16) and rearranging gives

$$\begin{aligned} V_N = \sum_{i=0}^{N-1} &\|\bar{\mathbf{\Phi}}_i\|_{\mathbf{Q}+\mathbf{L}_i^\top \mathbf{R}\mathbf{L}_i}^2 + \text{tr}\left((\mathbf{Q} + \mathbf{L}_i^\top \mathbf{R}\mathbf{L}_i)\Gamma_i\right) \\ &+ \|\mathbf{g}_i\|_{\mathbf{R}}^2 + 2\mathbf{g}_i^\top \mathbf{R}\mathbf{L}_i \Omega^\top \bar{\mathbf{\Phi}}_i + \|\bar{\mathbf{\Phi}}_N\|_{\mathbf{S}}^2 + \text{tr}\left(\mathbf{S}\Gamma_N\right), \end{aligned} \tag{8.21}$$

where $\Omega = (I_{n_x} \otimes e_{p+1})$. The mean and variance can be approximated, from (8.18) and (8.19), as

$$\mathbb{E}[\phi_i] \approx \Omega^\top \bar{\mathbf{\Phi}}_i, \tag{8.22}$$

and

$$\begin{aligned} \mathbb{E}[\phi_i \phi_i^\top] \approx &\left[\{(I_{n_x} \otimes \mathbf{1}_{p+1}) \circ (\bar{\mathbf{\Phi}}_i \mathbf{1}_{n_x}^\top)\}^\top (\mathbf{I}_{n_x} \otimes W)\{(I_{n_x} \otimes \mathbf{1}_{p+1}) \circ (\bar{\mathbf{\Phi}}_i \mathbf{1}_{n_x}^\top)\}\right] \\ &+ M(I_{n_x} \otimes \text{vec}(\Gamma_i)), \end{aligned} \tag{8.23}$$

where

$$M = \begin{bmatrix} \text{vec}(E_{1,1} \otimes W)^\top & \cdots & \text{vec}(E_{1,n_x} \otimes W)^\top \\ \vdots & \ddots & \vdots \\ \text{vec}(E_{n_x,1} \otimes W)^\top & \cdots & \text{vec}(E_{n_x,n_x} \otimes W)^\top \end{bmatrix},$$

and $E_{i,j} \in \mathbb{R}^{n_x \times n_x}$ is a binary matrix with a value 1 in only the $(i,j)^{th}$ position. Note that the covariance matrix is obtained from $\mathbf{Var}[\phi_i] = \mathbb{E}[\phi_i \phi_i^\top] - \mathbb{E}[\phi_i]\mathbb{E}[\phi_i]^\top$. The tractable formulation of Problem 8.2 is then given by:

Problem 8.3 (Tractable SMPC Formulation with State Chance Constraints)

$$\min_{(\tilde{L},\tilde{g})} \; V_N(x,\tilde{L},\tilde{g}) \tag{8.24}$$

$$\begin{aligned}
\text{s.t.} \quad & \bar{\mathbf{\Phi}}_{i+1} = (\mathbf{A}+\mathbf{B}\mathbf{L}_i)\bar{\mathbf{\Phi}}_i + \mathbf{B}\mathbf{g}_i, && i \in \mathbb{Z}_{[0,N-1]}, \\
& \Gamma_{i+1} = (\mathbf{A}+\mathbf{B}\mathbf{L}_i)\Gamma_i(\mathbf{A}+\mathbf{B}\mathbf{L}_i)^\top + \mathbf{F}\Sigma\mathbf{F}^\top, && i \in \mathbb{Z}_{[0,N-1]}, \\
& f_{cc}^{[l]}(\bar{\mathbf{\Phi}}_i,\Gamma_i) \leq 0, && i \in \mathbb{Z}_{[0,N-1]},\ l \in \mathbb{Z}_{[1,n_{cc}]}, \\
& \bar{\mathbf{\Phi}}_0 = x \otimes e_{p+1},\ \Gamma_0 = 0,
\end{aligned}$$

where $f_{cc}^{[l]}$ *denotes the* l^{th} *chance constraint as a function of* $(\bar{\mathbf{\Phi}}_i,\Gamma_i)$*, which is straightforwardly derived by substituting* (8.22) *and* (8.23) *into the deterministic chance constraint* (8.9).

8.5 Stability Analysis for the Unconstrained Case

Our initialization strategy, $(\bar{\mathbf{\Phi}}_0 = x \otimes e_{p+1}, \Gamma_0 = 0)$, uses the current state observation x, and is equivalent to having the SMPC problem optimize state predictions conditioned on recent data x. However, the closed-loop state, is influenced by an unbounded disturbance w so that it is impossible to assert convergence of the states to any compact set under any control policy. In other words, there will almost surely be excursions of the states beyond any compact set infinitely often over an infinite time horizon [43].

The fact that x can jump anywhere in $\mathbb{R}^{n_x}$ also makes it difficult to guarantee feasibility of chance constraints (must know there always exists some affine state feedback control law can recover from any arbitrary initial condition). An alternative method for guaranteeing feasibility is to switch the initialization strategy to correspond to open-loop predictions while adding appropriate terminal constraints (discussed in [67]). We avoid these methods here by proving stability for the unconstrained version of the SMPC problem.

Here, we focus on discrete-time Markov processes $\{x_t\}_{t\in\mathbb{N}_0}$ where the PDF of the

future state x_{t+1} is conditionally independent of the past $x_0, \cdots, x_t$ given the present state x_t. We are concerned with a type of stability that concerns boundedness of sequences of the form $\{\mathbb{E}[h(x_t)|x_0 = x]\}_{t\in\mathbb{N}_0}$ where h is some norm-like function [43]. The theory of stability for discrete-time Markov processes entails the notion of a *negative drift condition* [170].

Theorem 8.2 (Geometric Drift) *Let $\{x_t\}_{t\in\mathbb{N}_0}$ denote a Markov process. Suppose there exists a measurable function $V : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}_{\geq 0}$, a compact set $D \subset \mathbb{R}^{n_x}$ such that $\mathbb{E}[V(x_1)|x_0 = x]$, $\forall x \notin D$, and $\sup_{x\in D} \mathbb{E}[V(x_1)|x_0 = x] = b$ for some constants $b \geq 0$ and $\lambda \in [0, 1)$. Then, $\mathbb{E}[V(x_t)|x_0 = x] \leq \lambda^t V(x) + b(1-\lambda)^{-1}$ for all $x \in \mathbb{R}^{n_x}$ and $t \in \mathbb{N}_0$. This implies the sequence $\{\mathbb{E}[V(x_t)|x_0 = x]\}_{t\in\mathbb{N}_0}$ is bounded for all $x \in \mathbb{R}^{n_x}$. A "geometric drift condition" is also satisfied for states outside a compact set i.e.,*

$$\mathbb{E}[V(x_1)|x_0 = x] - V(x) \leq -(1-\lambda)V(x), \ \forall x \notin D.$$

Proof. A proof of this theorem is provided in the Appendix of [43] and is repeated here for completeness and clarity. First, $\mathbb{E}[V(x_t)|x_0 = x] = \mathbb{E}[\mathbb{E}[V(x_t)|\{x_s\}_{s=1}^{t-1}]|x_0 = x] = \mathbb{E}\left[\mathbb{E}[V(x_t)|x_{t-1}]|x_0 = x\right]$ from the law of iterated expectations. Next, we can derive the bound $\mathbb{E}[V(x_t)|x_{t-1}] \leq \lambda V(x_{t-1})\mathcal{I}_{\mathbb{R}^{n_x}\setminus\mathbb{D}}(x_{t-1}) + b\mathcal{I}_{\mathbb{D}}(x_{t-1})$ for all $x_{t-1} \in \mathbb{R}^{n_x}$ from the hypothesis of the theorem. Combining these gives $\mathbb{E}[V(x_t)|x_0 = x] \leq \lambda\mathbb{E}[V(x_{t-1})|x_0 = x] + b\mathbb{P}(x_{t-1} \in \mathbb{D}|x_0 = x)$. Repeating these steps for $\{\mathbb{E}[V(x_s)|x_0 = x]\}_{s=1}^{t-1}$ and recursively substituting into the last inequality gives

$$\begin{aligned}
\mathbb{E}[V(x_t)|x_0 = x] &\leq \lambda^t V(x) + b\sum_{i=0}^{t-1} \lambda^i \mathbb{P}(x_{t-1-i} \in \mathbb{D}|x_0 = x), \\
&\leq \lambda^t V(x) + b\sum_{i=0}^{t-1} \lambda^i, \\
&\leq \lambda^t V(x) + b\sum_{i=0}^{\infty} \lambda^i, \\
&\leq \lambda^t V(x) + b(1-\lambda)^{-1},
\end{aligned}$$

where we have used the fact that $\mathbb{P}(x_{t-1-i} \in \mathbb{D}|x_0 = x) \leq 1$ and the geometric series

expression $1 + \lambda + \lambda^2 + \cdots = (1 - \lambda)^{-1}$, which is convergent for $|\lambda| < 1$ as assumed. From this result, we see that $\sup_{t \in \mathbb{N}_0} \mathbb{E}[V(x_t)|x_0 = x] \leq V(x) + b(1 - \lambda)^{-1} < \infty$ is bounded as claimed. □

The main stability results for stochastic predictive control, that we extend in this work to also handle parameter uncertainty, are presented in detail in [43]. The main goal is to select appropriate cost functions such that a drift condition on the optimal value function can be established.

Preliminaries

Let $n := n_x(p + 1)$ and $r := n_u(p + 1)$ denote the dimension of the gPC projected states and inputs, respectively.

We include a terminal cost $\|\mathbf{\Phi}_N\|_P^2$ in the objective where $P = P^\top > 0$ is the solution to the Lyapunov equation

$$(\mathbf{A} + \mathbf{BK})^\top P(\mathbf{A} + \mathbf{BK}) - P = -(1 + \delta)\mathbf{M}, \tag{8.25}$$

$\delta > 0$, $\mathbf{M} := \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K}$, and $\mathbf{K} := K \otimes I_{p+1}$. The objective function of interest V_N is now stated as

$$V_N(\mathbf{\Phi}, \tilde{L}, \tilde{g}) = \mathbb{E}\left[\sum_{i=0}^{N-1} \|\mathbf{\Phi}_i\|_{\mathbf{Q}}^2 + \|\mathbf{\Pi}_i\|_{\mathbf{R}}^2 + \|\mathbf{\Phi}_N\|_P^2 \middle| \mathbf{\Phi}_0 = \mathbf{\Phi}\right], \tag{8.26}$$

Since the pair $(A(\theta), B(\theta))$ is assumed to be stabilizable for all realizations of θ, there exists at feedback gain K and $P > 0$ that satisfies (8.25) [73]. Note that the argument $\mathbf{\Phi} \in \mathbb{R}^n$ represents the initial condition in the PC expansion coefficient space. We compactly denote the stage cost $c : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}_{\geq 0}$ and final cost $c_f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ in (8.26) as

$$c(\mathbf{\Phi}, \mathbf{\Pi}) = \|\mathbf{\Phi}\|_{\mathbf{Q}}^2 + \|\mathbf{\Pi}\|_{\mathbf{R}}^2, \tag{8.27}$$

$$c_f(\mathbf{\Phi}) = \|\mathbf{\Phi}\|_P^2. \tag{8.28}$$

Problem 8.4 (Unconstrained SMPC) *For any initial condition* $\mathbf{\Phi} \in \mathbb{R}^n$, *the unconstrained* N*-horizon stochastic optimal control problem (to be solved at every time step) can be stated as*

$$\min_{(\tilde{L},\tilde{g})} V_N(\mathbf{\Phi}, \tilde{L}, \tilde{g}), \quad \text{s.t.} \quad (8.14) \text{ and } (8.16), \ \forall i \in \mathbb{Z}_{[0,N-1]}. \tag{8.29}$$

Let π^* denote the optimal state feedback policy computed from Problem 8.4 with parametrization (8.10) i.e., $\pi_i^\star(x) = g_i^\star(x) + L_i^\star(x)x$ for $i \in \mathbb{Z}_{[0,N-1]}$. Given the state x_t at time t, implementing Problem 8.4 in receding-horizon consists of three steps:

- Solving (8.29) for $\pi^\star$ with $\mathbf{\Phi} = x_t \otimes e_{p+1}$.
- Giving the first element $\pi_0^\star$ to the system (8.1).
- Shifting time to $t+1$, and repeating the preceding steps.

As the true system (8.1) has fixed parameter values $\theta = \hat{\theta}$, it evolves as a Markov process. Uner the policy $\{\pi_0^\star, \pi_0^\star, \ldots\}$, (8.1) generates a state trajectory $\{x_t\}_{t\in\mathbb{N}_0}$ via the recursion

$$x_{t+1} = \hat{A}x_t + \hat{B}\pi_0^\star(x_t) + Fw_t, \ x_0 \text{ given}, \ t \in \mathbb{N}_0 \tag{8.30}$$

where $\hat{A} = A(\hat{\theta})$ and $\hat{B} = B(\hat{\theta})$ are the true plant matrices.

Stability through Boundedness of the Value Function

Let $(\tilde{L}^\star, \tilde{g}^\star)$ denote the optimal control parameters, corresponding to optimal policy $\pi^\star$ with parametrization (8.10), obtained by solving (8.29) for a given initial condition. Denote the optimal value function by $V_N^\star(\mathbf{\Phi}) := V_N(\mathbf{\Phi}, \tilde{L}^\star, \tilde{g}^\star)$. The following two lemmas form the basis for the main stability result for Problem 8.4, which is presented in Theorem 8.3.

Lemma 8.1 *The stage cost* (8.27), *final cost* (8.28), *and controller* $\mathbf{K\Phi}$ *satisfy*

$$\sup_{\mathbf{\Phi}\in D}\left\{c(\mathbf{\Phi},\mathbf{K\Phi})-c_f(\mathbf{\Phi})+\mathbb{E}\big[c_f\big((\mathbf{A}+\mathbf{BK})\mathbf{\Phi}+\mathbf{F}w_0\big)|\mathbf{\Phi}\big]\right\}\leq b, \tag{8.31a}$$

$$c(\mathbf{\Phi},\mathbf{K\Phi})-c_f(\mathbf{\Phi})+\mathbb{E}\big[c_f\big((\mathbf{A}+\mathbf{BK})\mathbf{\Phi}+\mathbf{F}w_0\big)|\mathbf{\Phi}\big]\leq 0,\ \forall \mathbf{\Phi}\notin D, \tag{8.31b}$$

for some constant $b\geq 0$ *and bounded measurable set* $D:=\big\{z\in\mathbb{R}^n|z^\top\mathbf{M}z\leq \frac{1}{\delta}\mathrm{tr}(\mathbf{F}^\top P\mathbf{F}\Sigma)\big\}$.

Proof. Substituting in the definitions of the cost functions, we derive $c(\mathbf{\Phi},\mathbf{K\Phi})=\|\mathbf{\Phi}\|^2_{\mathbf{Q}+\mathbf{K}^\top\mathbf{RK}}$ and $\mathbb{E}\big[c_f\big((\mathbf{A}+\mathbf{BK})\mathbf{\Phi}+\mathbf{F}w_0\big)|\mathbf{\Phi}\big]=\|(\mathbf{A}+\mathbf{BK})\mathbf{\Phi}\|^2_P+\mathrm{tr}(\mathbf{F}^\top P\mathbf{F}\Sigma)$. Using the Lyapunov equation (8.25), we have

$$\begin{aligned}
&c(\mathbf{\Phi},\mathbf{K\Phi})-c_f(\mathbf{\Phi})+\mathbb{E}\big[c_f\big((\mathbf{A}+\mathbf{BK})\mathbf{\Phi}+\mathbf{F}w_0\big)|\mathbf{\Phi}\big]\\
&=\mathbf{\Phi}^\top((\mathbf{A}+\mathbf{BK})^\top P(\mathbf{A}+\mathbf{BK})-P+\mathbf{M})\mathbf{\Phi}+\mathrm{tr}(\mathbf{F}^\top P\mathbf{F}\Sigma)\\
&=-\delta\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi}+\mathrm{tr}(\mathbf{F}^\top P\mathbf{F}\Sigma)
\end{aligned}$$

We know that $\delta\inf_{\mathbf{\Phi}\in\mathbf{D}}(\mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi})=0$ such that the supremum of this expression is $\mathrm{tr}(\mathbf{F}^\top P\mathbf{F}\Sigma)>0$. Therefore, there exists a number $b\geq 0$ that satisfies assertion (8.31a). For all $\mathbf{\Phi}\notin D$, this expression will be less than or equal to zero such that assertion (8.31b) is also satisfied. □

Lemma 8.2 *For all* $\mathbf{\Phi}\in\mathbb{R}^n$, *the optimal value function satisfies the inequality* $V_N^\star(\mathbf{\Phi})\leq c_f(\mathbf{\Phi})+Nb$.

Proof. Define the N-length sequences $\tilde{L}^s:=\{K,\ldots,K\}$ and $\tilde{g}^s:=\{0,\ldots,0\}$. Let $\{\mathbf{\Phi}_i^s\}_{i=1}^N$ denote the sequence obtained by applying the policy $(\tilde{L}^s,\tilde{g}^s)$ to the gPC system (8.14), i.e., $\mathbf{\Phi}_{i+1}^s=(\mathbf{A}+\mathbf{BK})\mathbf{\Phi}_i^s+\mathbf{F}w_i$ for all $i\in\mathbb{Z}_{[0,N-1]}$ with initial condition $\mathbf{\Phi}_0^s=\mathbf{\Phi}$ for any fixed $\mathbf{\Phi}\in\mathbb{R}^n$, and $V_N^s(\mathbf{\Phi}):=V_N(\mathbf{\Phi},\tilde{L}^s,\tilde{g}^s)$. From Lemma 8.1, we

can derive

$$
\begin{aligned}
c_f(\mathbf{\Phi}) &\geq \mathbb{E}\big[c_f(\mathbf{\Phi}_1^f)|\mathbf{\Phi}_0^s = \mathbf{\Phi}\big] + c(\mathbf{\Phi}, \mathbf{K}\mathbf{\Phi}) - b \\
c_f(\mathbf{\Phi}_1^s) &\geq \mathbb{E}\big[c_f(\mathbf{\Phi}_2^s)|\mathbf{\Phi}_1^s\big] + c(\mathbf{\Phi}_1^s, \mathbf{K}\mathbf{\Phi}_1^s) - b \\
&\vdots \\
c_f(\mathbf{\Phi}_{N-1}^s) &\geq \mathbb{E}\big[c_f(\mathbf{\Phi}_N^s)|\mathbf{\Phi}_{N-1}^s\big] + c(\mathbf{\Phi}_{N-1}^s, \mathbf{K}\mathbf{\Phi}_{N-1}^s) - b
\end{aligned}
$$

Recursively substituting these expressions into one another gives $c_f(\mathbf{\Phi}) \geq V_N^s(\mathbf{\Phi}) - Nb$. Subtracting $V_N^\star(\mathbf{\Phi})$ from both sides of this inequality gives $c_f(\mathbf{\Phi}) - V_N^\star(\mathbf{\Phi}) \geq V_N^s(\mathbf{\Phi}) - V_N^\star(\mathbf{\Phi}) - Nb$. The assertion follows by noting that $V_N^s(\mathbf{\Phi}) \geq V_N^\star(\mathbf{\Phi})$, since $(\tilde{L}^s, \tilde{g}^s)$ is a suboptimal policy to $(\tilde{L}^\star, \tilde{g}^\star)$ for arbitrary $\mathbf{\Phi} \in \mathbb{R}^n$. □

Given the optimal parameters $(\tilde{L}^\star, \tilde{g}^\star)$, we denote the following feasible sequences of the control parameters at the next time instant as

$$
\begin{aligned}
\tilde{L}^f &:= \{L_1^\star, \ldots, L_{N-1}^\star, K\}, \\
\tilde{g}^f &:= \{g_1^\star, \ldots, g_{N-1}^\star, 0\},
\end{aligned}
$$

where we have taken the last $N-1$ elements of the optimal policy $(\tilde{L}^\star, \tilde{g}^\star)$ and added to it the state feedback law designed to satisfy the Lyapunov equation (8.25). Let $V_N^f(\mathbf{\Phi}) := V_N(\mathbf{\Phi}, \tilde{L}^f, \tilde{g}^f)$ and define $\mathbf{\Pi}^\star := \{\mathbf{\Pi}_0^\star, \ldots, \mathbf{\Pi}_{N-1}^\star\}$ to be the optimal propagated policy of the form (8.16). We denote the "optimal" states $\{\mathbf{\Phi}_i^\star\}_{i=0}^N$, generated by applying $\mathbf{\Pi}^\star$ to the gPC system (8.14), as

$$
\mathbf{\Phi}_{i+1}^\star = \mathbf{A}\mathbf{\Phi}_i^\star + \mathbf{B}\mathbf{\Pi}_i^\star(\mathbf{\Phi}_i^\star) + \mathbf{F}w_i, \ \mathbf{\Phi}_0^\star \text{ given.} \tag{8.32}
$$

Using these definitions, we make the following assumption about how the true states, from (8.30), relate to the "optimal" PCE coefficient states in (8.32):

$$
\mathbb{E}\big[V_N^f(x_1 \otimes e_{p+1})|x_0 = x\big] \leq \mathbb{E}\big[V_N^f(\mathbf{\Phi}_1^\star)|\mathbf{\Phi}_0^\star = x \otimes e_{p+1}\big], \ \forall x \in \mathbb{R}^{n_x}. \tag{8.33}
$$

Theorem 8.3 *Consider the system* (8.1) *at a fixed* $\theta = \hat{\theta}$ *and the stochastic optimal control problem* (8.29). *Suppose that assumption* (8.33) *holds. Then,* $\{\mathbb{E}[V_N^\star(x_t \otimes e_{p+1})|x_0 = x]\}_{t\in\mathbb{N}_0}$ *is bounded for each* $x \in \mathbb{R}^{n_x}$.

Proof. From the definitions of the value function (8.26), the recursion (8.32), and V_N^f, we know that

$$\begin{aligned}
&\mathbb{E}\big[V_N^f(\boldsymbol{\Phi}_1^\star)|\boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\big] - V_N^\star(x \otimes e_{p+1}),\\
&= \mathbb{E}\Big[- \|\boldsymbol{\Phi}_0^\star\|_{\mathbf{Q}}^2 - \|\boldsymbol{\Pi}_0^\star(\boldsymbol{\Phi}_0^\star)\|_{\mathbf{R}}^2 + \|\boldsymbol{\Phi}_N^\star\|_{\mathbf{Q}}^2 + \|\mathbf{K}\boldsymbol{\Phi}_N^\star\|_{\mathbf{R}}^2\\
&+ \|(\mathbf{A}+\mathbf{B}\mathbf{K})\boldsymbol{\Phi}_N^\star + \mathbf{F}w_N\|_P^2 - \|\boldsymbol{\Phi}_N^\star\|_P^2 \big| \boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\Big].
\end{aligned}$$

The first two terms above can be taken out of the expected value and derived to be $\|x \otimes e_{p+1}\|_{\mathbf{Q}}^2 + \|\boldsymbol{\Pi}_0^\star(x \otimes e_{p+1})\|_{\mathbf{R}}^2 = \|x\|_Q^2 + \|\pi_0^\star(x)\|_R^2$. We can apply the law of iterated expectation for Markov processes on the remaining terms above, and then use Lemma 8.1 to obtain a bound on this expression

$$\begin{aligned}
&\mathbb{E}\Big[\mathbb{E}\Big[\|\boldsymbol{\Phi}_N^\star\|_{\mathbf{Q}}^2 + \|\mathbf{K}\boldsymbol{\Phi}_N^\star\|_{\mathbf{R}}^2 + \|(\mathbf{A}+\mathbf{B}\mathbf{K})\boldsymbol{\Phi}_N^\star + \mathbf{F}w_N\|_P^2\\
&\qquad - \|\boldsymbol{\Phi}_N^\star\|_P^2 \big| \{\boldsymbol{\Phi}_s^\star\}_{s=0}^N\Big] \big| \boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\Big],\\
&\leq \mathbb{E}\big[b\mathcal{I}_D(\boldsymbol{\Phi}_N^\star)\big|\boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\big],\\
&= b\mathbb{P}\big(\boldsymbol{\Phi}_N^\star \in D \big| \boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\big),\\
&\leq b.
\end{aligned}$$

We can apply these results to our starting expression to derive $\mathbb{E}\big[V_N^f(\boldsymbol{\Phi}_1^\star)|\boldsymbol{\Phi}_0^\star = x \otimes e_{p+1}\big] - V_N^\star(x \otimes e_{p+1}) \leq -(\|x\|_Q^2 + \|\pi_0^\star(x)\|_R^2) + b$. From the optimality of $(\tilde{L}^\star, \tilde{g}^\star)$ and from assumption (8.33), we know that

$$\begin{aligned}
\mathbb{E}\big[V_N^\star(x_1 \otimes e_{p+1})|x_0 = x\big] - V_N^\star(x \otimes e_{p+1}) &\leq \mathbb{E}\big[V_N^f(x_1 \otimes e_{p+1})|x_0 = x\big] - V_N^\star(x \otimes e_{p+1}),\\
&\leq \mathbb{E}\big[V_N^f(\boldsymbol{\Phi}_1^\star)|\boldsymbol{\Phi}_0 = x \otimes e_{p+1}\big] - V_N^\star(x \otimes e_{p+1}),\\
&\leq -\big(\|x\|_Q^2 + \|\pi_0^\star(x)\|_R^2\big) + b,\\
&\leq -\|x\|_Q^2 + b,
\end{aligned}$$

For some constant $\alpha \in [0,1)$, we define the set $D' := \{z \in \mathbb{R}^{n_x} | z^\top Q z \leq \alpha(z \otimes e_{p+1})^\top P(z \otimes e_{p+1})\}$ such that $\mathbb{E}\big[V_N^\star(x_1 \otimes e_{p+1}) | x_0 = x\big] - V_N^\star(x \otimes e_{p+1}) \leq -\alpha c_f(x \otimes e_{p+1}) + b$ for all $x \notin D'$. From Lemma 8.2, we have $-\alpha c_f(x \otimes e_{p+1}) \leq -\alpha V_N^\star(x \otimes e_{p+1}) + \alpha N b$ for all $x \in \mathbb{R}^{n_x}$ such that $\mathbb{E}\big[V_N^\star(x_1 \otimes e_{p+1}) | x_0 = x\big] - V_N^\star(x \otimes e_{p+1}) \leq -\alpha V_N^\star(x \otimes e_{p+1}) + b(1 + \alpha N)$, for all $x \notin D'$. Since $\lim_{\|z\| \to +\infty} c(z \otimes e_{p+1}, \pi_0^\star(z) \otimes e_{p+1}) = +\infty$, we find, from the definition (8.26), that $\lim_{\|z\| \to +\infty} V_N^\star(z \otimes e_{p+1}) = +\infty$. From the definition of a limit, there must exists a closed ball D'' around the origin $0 \in \mathbb{R}^{n_x}$ of a radius large enough such that $V_N^\star(z \otimes e_{p+1}) \geq 2b(\alpha^{-1} + N)$ for all $z \notin D''$ [43]. Substituting this into the previous expression gives

$$\mathbb{E}\big[V_N^\star(x_1 \otimes e_{p+1}) | x_0 = x\big] - V_N^\star(x \otimes e_{p+1}) \leq -\frac{\alpha}{2} V_N^\star(x \otimes e_{p+1}), \quad \forall x \notin D'',$$

The sets D' and D'' should satisfy $D' \subseteq D'' \subset \mathbb{R}^{n_x}$ such that $x \notin D'' \Rightarrow x \notin D'$. This represents a geometric drift condition outside the compact set D''. The assertion directly follows from Theorem 8.2. □

As shown in the proof of Theorem 8.3, $V_N^\star$ satisfies a geometric drift condition outside of some compact set of $\mathbb{R}^{n_x}$. Therefore, we can claim that the receding-horizon controller (under the proper assumptions) results in a bounded objective for all time such that the discrete-time Markov system is stochastically stable.

8.6 Example: Van de Vusse Reactor

The Van de Vusse series of reactions [57]

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C, \quad 2A \xrightarrow{k_3} D,$$

taking place in an isothermal continuous stirred-tank reactor is considered to evaluate the performance of the SMPC approach. The dynamic evolution of the concentration

of A and B (denoted by C_A and C_B, respectively) is described by

$$\begin{aligned} \dot{C}_A &= -k_1 C_A - k_3 C_A^2 - C_A u, \\ \dot{C}_B &= k_1 C_A - k_2 C_B - C_B u, \end{aligned} \tag{8.34}$$

where k_1, k_2, and k_3 denote reaction rate constants and u is the dilution rate (i.e., cpntrol action). Linearizing the system (8.34) around a desired steady state operating point and discretizing the linearized model with a sampling time of 0.002 (see [226]) results in a linear system of the form (8.1) with

$$A = \begin{pmatrix} \theta_1 & 0 \\ 0.088 & 0.819 \end{pmatrix} \qquad B = \begin{pmatrix} -0.005 \\ -0.002 \end{pmatrix},$$

where θ_1 is a random variable with PDF given by the four-parameter β distribution $\beta(0.923, 0.963, 2, 5)$. The noise matrix in (8.1) is assumed to be identity, and $\Sigma = 10^{-4} I$ with I being a 2×2 identity matrix. The states of the linearized model are defined in terms of the deviation variables x_1 and x_2. The initial states are assumed to be random variables with Gaussian PDFs, i.e., $x_1(0) \sim \mathcal{N}(0.5, 0.01)$ and $x_2(0) \sim \mathcal{N}(0.1, 0.01)$. The control objective is to retain both states at the desired steady state (hence $x_{1_{sp}} = 0$ and $x_{2_{sp}} = 0$) in the presence of time-invariant probabilistic uncertainties and process noise. In addition, x_2 should remain below the limit 0.17.

To formulate an SMPC problem of the form (8.24), fifth-order Jacobi polynomials are used to propagate the time-invariant uncertainties through the system dynamics. The weight matrix Q is set equal to identity with $R = 0$. The probability β_i in the chance constraint imposed on x_2 is 0.95, which indicates that at least in 95% of occurrences the constraint $x_2 < 0.17$ should be satisfied.

The performance of the proposed SMPC approach is evaluated based on 100 closed-loop simulations in the presence of probabilistic uncertainties and process noise, and is compared with that of a nominal MPC approach with terminal constraints. Figure 5-1 shows the histograms of x_1 for both MPC approaches at three different times. The SMPC approach clearly leads to smaller mean (i.e., deviation with respect

to the steady state value) and smaller variance. This indicates that the SMPC approach can effectively deal with the system uncertainties and process noise. Figure 8-1 shows that the state approaches its steady state value (x_1 approaches zero). To assess the satisfaction of the state constraint, the time profiles of x_2 for the 100 runs are shown in Figure 8-2. The state constraint is fulfilled in over 95% of simulations, whereas it is violated in nearly 46% of closed-loop simulations of the nominal MPC approach. Hence, the inclusion of the chance constraint into the SMPC approach leads to effective state constraints satisfaction.

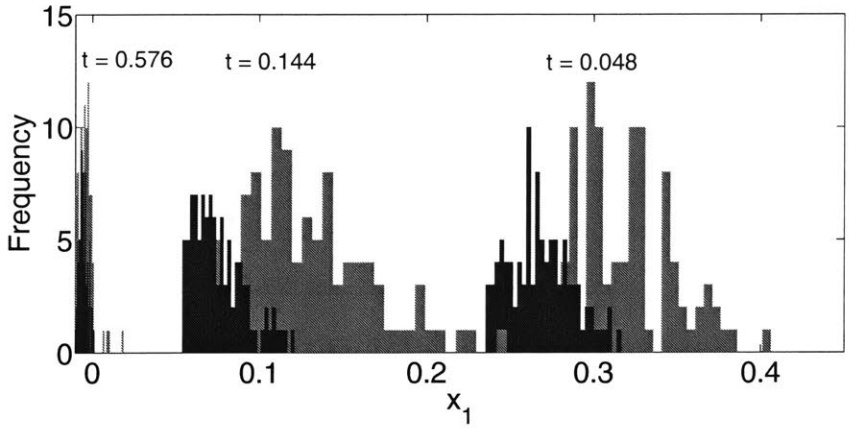
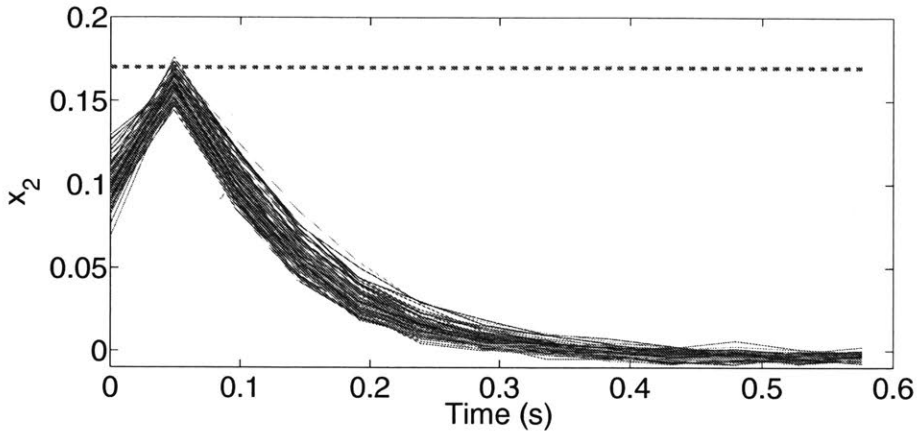


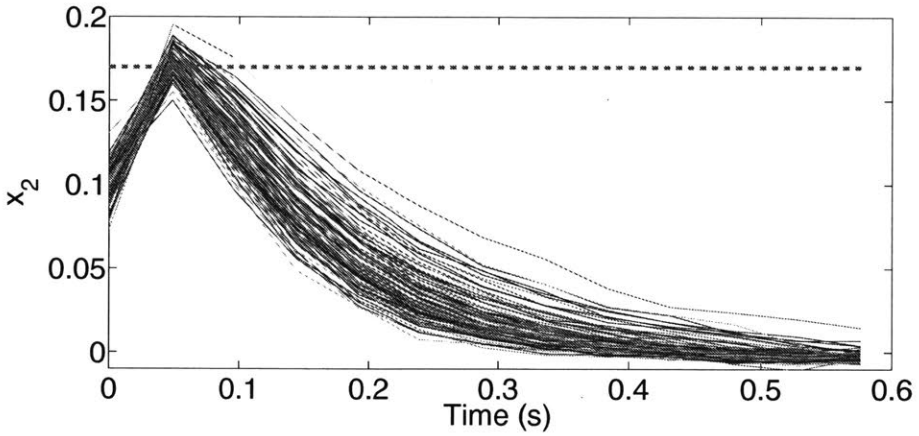
Figure 8-1: Histograms of x_1 at different times obtained from 100 closed-loop simulations of the receding-horizon SMPC (blue) and MPC (red). The proposed SMPC approach leads to smaller mean and variance of x_1 in the presence of probabilistic uncertainties and process noise.

8.7 Conclusions

This chapter presents an SMPC approach with state chance constraints for linear systems subject to time-invariant probabilistic uncertainty in the model and additive Gaussian process noise. A tractable formulation for this type of SMPC problem is presented using deterministic surrogates to simplify the problem. Conditions for closed-loop stability are established in the unconstrained case.



(a) Proposed SMPC approach



(b) Nominal MPC approach

Figure 8-2: Time profiles of x_2 obtained from 100 closed-loop simulations of the proposed SMPC approach and nominal MPC. The red-dashed line represents the state constraint. Nominal MPC results in violation of the constraint in 46% of the cases.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 9

Improved Output Feedback with Bayesian Learning

9.1 Introduction

Mathematical models of physical systems are never exact: parameters are subject to uncertainties and variations, states and outputs underly noise and external disturbances, and not all states can be accessed directly from measurements. From a control perspective this leads to many questions such as: How can one efficiently and reliably obtain estimates of the parameters and states? How can one account for such uncertainties in the controller?

In this chapter, we exploit polynomial chaos methods (see Chapter 3) to derive solutions to these questions in certain cases by a incorporating a new recursive Bayesian estimation scheme into stochastic MPC methods discussed and referenced in previous chapters of this thesis. This is first motivated by the fact that the interpretation considered in many works in the literature [169, 199, 238, 72], in which the parameter distribution $f(\theta)$ remains constant at every iteration of the MPC problem, is suboptimal in most cases. When $f(\theta)$ is interpreted as a frequency of occurrence (for example, a collection of a number of batches/systems each corresponding to a different value of θ), chance constraints may not be satisfied in the closed-loop system even when they are satisfied in every iteration of a receding horizon control problem

(this is rigorously proved below).

Alternatively, $f(\theta)$ might be a quantitative description of our confidence in a particular value of θ being realized (as would be the most common case in continuous processes). In this case, the true system is deterministic (evolving at fixed, but unknown values of the parameters) so that chance constraints must also be interpreted as trying to enforce a constraint with a particular confidence, even though the real system either meets constraints or does not [198]. We explore the latter interpretation in this chapter. It is important to realize that the online measurements are providing new information about the process parameters that should be incorporated into the PDF representation of our confidence. The more data/knowledge that we have, the tighter we would expect this distribution to become. Bayes' theorem is the main tool that we can use to recursively update our parameter PDFs.

Polynomial chaos can be used to efficiently propagate and quantify probabilistic uncertainties by approximation of the PDFs of random variables [271]. This technique has been exploited in many contexts within control and estimation. With respect to estimation, in [156] for example, the location of a contamination source is obtained solving a single Bayesian estimation problem with multiple measurements. In [28], recursive schemes using polynomial chaos in conjunction with the Galerkin projection method are developed to estimate PDFs of uncertain parameters. Similar to [153], this chapter uses Bayes' theorem to generate moments that are nonlinear functions of polynomial chaos expansion (PCE) coefficients. The solution to a nonlinear least-squares problem then yields the updated PCE coefficients (which is directly used for numerical approximation of the updated parameter PDF).

The use of polynomial chaos in stochastic MPC has been explored in a number of works, e.g., [66, 110, 199, 239]. Algorithms were also developed/extended for various situations in Chapters 6–8 of this thesis. The focus of this chapter is to use the combination of Bayesian estimation and stochastic MPC to drive a deterministic (but unknown) plant to a desired reference. This approach makes the most sense in the context of continuous processes with models whose structure is accurate, but the parameters are not exactly known and do not change much during operation.

Organization The next section proves that chance constraints are not guaranteed to be satisfied in receding horizon control when the parameter PDF $f(\theta)$ is not updated at each iteration, which motivates the exploration of Bayesian updating methods. Section 9.3 presents the proposed recursive Bayesian estimation problem and describes a method for efficiently solving this problem numerically using polynomial chaos. Section 9.4 briefly describes how to incorporate this estimation problem into stochastic MPC algorithms. Lastly, the method is illustrated on an example problem composed of a series of continuously stirred tank reactors in Section 9.5 and conclusions are drawn in Section 9.6.

Parts of this chapter were originally published in the *Proceedings of the American Control Conference* [176].

9.2 Failure of Chance Constraints in Receding Horizon Control

The majority of research on stochastic MPC with probabilistic time-invariant parameter uncertainty has focused on developing practical algorithms with reduced computational cost (so that they can be implemented in real-time) and have demonstrated improved robustness compared to nominal MPC algorithms. The theoretical development of these algorithms (see [169, 199, 238, 72] and Chapter 6 for examples), however, could be significantly improved.

In the previous chapter of this thesis, an algorithm was developed that can handle both stochastic disturbances and parameter uncertainty. Although chance constraints are included in each step of the receding horizon implementation of the algorithm, the stability analysis was performed in the unconstrained case due to the complexity added by the chance constraints. In Chapter 7, we were able to guarantee satisfaction of chance constraints since the uncertainty was assumed to be time-varying and independent of all past and future values. This allows us to apply the Markov property (and results derived for Markov systems) to the state space description of the

system. Consideration of time-invariant uncertainty, however, is more challenging as the current information is fully correlated with the past. This can be illustrated using a simple example.

Example Consider the following one-state discrete time linear system

$$x_{k+1} = \theta x_k, \tag{9.1}$$

where θ is a random gain with prior distribution $f(\theta)$. The PDF of x_2 given a particular value x_1 is

$$f(x_2|x_1) = \frac{1}{|x_1|} f\left(\theta = \frac{x_2}{x_1}\right). \tag{9.2}$$

However, the PDF of x_2 given two previous states x_1 and x_0 is

$$f(x_2|x_1, x_0) = \delta\left(x_2 - \frac{x_1^2}{x_0}\right), \tag{9.3}$$

where δ denotes the Dirac delta function. The latter equation is derived based on the fact that the value of the parameter $\theta = x_1/x_0$, so that the value of x_2 can be determined with certainty.

Chance constraints have been incorporated into a number of stochastic MPC problems that consider time-invariant parameter uncertainty. Although these chance constraints are fulfilled in each instance of the open-loop control problem (solved at every iteration), it has not been shown that these the desired chance constraints are guaranteed to be fulfilled by the *closed-loop* system (that is, taking into effect the recursive solution to the MPC problem). We prove that a standard formulation of these problems do not provide chance constraint guarantees on the closed-loop system by counterexample below.

Theorem 9.1 *Consider the following discrete-time stochastic system*

$$x_{k+1} = g(x_k, u_k, \theta), \tag{9.4}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state vector, $u_k \in \mathbb{R}^{n_u}$ is the input vector, $\theta \in \mathbb{R}^{n_\theta}$ is a real-valued random vector of parameter values with PDF $f(\theta)$, and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_x}$ is some known algebraic function. Given the observed state $x \in \mathbb{R}^{n_x}$, define the stochastic optimal control problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & J(x, \mathbf{u}), && && (9.5) \\ \text{s.t.} \quad & z_{k+1} = g(z_k, u_k, \theta), && k = 0, \cdots, N-1, \\ & u_k \in \mathbb{U}, && k = 0, \cdots, N-1, \\ & \mathbb{P}(z_k \in \mathbb{X}) \geq \beta, && k = 1, \cdots, N, \\ & z_0 \sim \delta(x), \quad \theta \sim f(\theta), \end{aligned}$$

where J is the objective function, $\mathbf{u} = (u_0, \ldots, u_{N-1})$ are the input values to be optimized over a prediction horizon N, z_k are the model predictions which are random variables due to the uncertainty in θ, $\mathbb{U}$ are the input constraints, $\mathbb{X}$ are the state constraints, and $z_0 \sim \delta(x)$ indicates that z_0 has a PDF of a Dirac delta at x (meaning that z_0 is a known value equal to the measured states x).

Let $\mathbf{u}^\star(x)$ be the optimal input trajectory obtained by solving (9.5) for any $x \in \mathbb{R}^{n_x}$, and let $\kappa_N : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ be the receding horizon control law $\kappa_N(x) = u_0^\star(x)$ defined as the first element of the optimal trajectory $\mathbf{u}^\star(x)$. Then, the stochastic closed-loop system, derived by applying κ_N to (9.4)

$$x_{k+1} = g(x_k, \kappa_N(x_k), \theta), \tag{9.6}$$

may not satisfy the state chance constraints $\mathbb{P}(x_k \in \mathbb{X}) \geq \beta$ even though they are feasible/satisfied in (9.5) at every time step.

Proof. This is shown by selecting any dynamic system of the form (9.4), parameter distribution $f(\theta)$, state constraints $\mathbb{X}$, probability level β, input constraints $\mathbb{U}$, and prediction horizon N such the states generated by (9.6) do not satisfy $\mathbb{P}(x_k \in \mathbb{X}) \geq \beta$ even though a feasible solution exists to (9.5) at every time step. Consider the simple

scalar system

$$
\begin{aligned}
&x_{k+1} = \theta x_k + u_k, \quad x_0 = 1, \\
&f(\theta) = 0.5\delta(\theta - 1) + 0.5\delta(\theta - 2), \\
&\mathbb{U} = \mathbb{R}, \quad \mathbb{X} = \{x \in \mathbb{R} : x \leq 2\}, \quad \beta = 0.5, \quad N = 2.
\end{aligned}
$$

From any initial condition x, the PDFs of the predicted states z_1 and z_2, as functions of input values u_0 and u_1, can be derived to be

$$
\begin{aligned}
f(z_1) &= 0.5\delta(z_1 - x - u_0) + 0.5\delta(z_1 - 2x - u_0), \\
f(z_2) &= 0.5\delta(z_2 - x - u_0 - u_1) + 0.5\delta(z_2 - 4x - 2u_0 - u_1).
\end{aligned}
$$

For the first iteration of (9.5), we know that $x = 1$ since $x_0 = 1 \sim \delta(1)$. A feasible input sequence is given by $\mathbf{u} = [-1.1, 0.3]^\top$ as this results in $(z_1 = -0.1, z_2 = 0.2)$ with 50% probability and $(z_1 = 0.9, z_2 = 2.1)$ with 50% probability, which satisfies the state chance constraint over the horizon. Only the first input will be supplied to the system as we are implementing this strategy in receding horizon.

There are two cases to explore in the second iteration of (9.5) as θ can take on either the value 1 or 2. Let us first consider the case that $\theta = 1$ in the true system. As a result, $x_1 = 1 - 1.1 = -0.1$ such that (9.5) should be initialized with $x = -0.1$. For this problem, we can select $\mathbf{u} = [2.2, -1]^\top$ as a feasible input sequence since this results in $(z_1 = 2.1, z_2 = 1.1)$ with 50% probability and $(z_1 = 2.0, z_2 = 3.0)$ with 50% probability. Supplying the first element of this sequence $u_1 = 2.2$ to the true system, $x_2 = 2.1$ so the desired constraint $x_2 \leq 2$ is violated when $\theta = 1$.

Next, consider the case that $\theta = 2$ in the true system. Here, the closed-loop state is $x_1 = 2 - 1.1 = 0.9$ such that (9.5) should be initialized with $x = 0.9$. For this problem, we can select $\mathbf{u} = [1.1, 0]^\top$ as a feasible input sequence since this results in $(z_1 = 2.0, z_2 = 2.0)$ with 50% probability and $(z_1 = 2.9, z_2 = 5.8)$ with 50% probability. Supplying the first element of this sequence $u_1 = 1.1$ to the true system, $x_2 = 2.9$ so the desired constraint $x_2 \leq 2$ is violated when $\theta = 2$.

For this particular example, the receding horizon control policy was selected to ensure that the chance constraints were satisfied in each instance of the optimal control problem. Under this feasible receding horizon policy, the true system reaches $x_2 = 2.1$ with 50% probability and $x_2 = 2.9$ with 50% probability, meaning that $\mathbb{P}(x_2 \leq 2) = 0$. As such, the desired state chance constraint is violated by the closed-loop system. As these feasible input sequences may be selected by some choice of objective J (e.g., a trivial case is when $J = 0$), the assertion directly follows. □

As discussed in detail in Chapter 3, uncertainty descriptions typically come from some sort of parameter estimation algorithm combined with uncertainty quantification techniques. Although we treat the parameter as a random variable in our models, the PDF $f(\theta)$ can be interpreted as a representation of our belief about the value of θ. It is common to assume that the true parameters are fixed, they are just now known exactly (especially when modeling systems with time-invariant uncertainty as the parameter does not change at all over time). In this case, the distribution of the predicted states cannot be interpreted as a frequency of occurrence, meaning that satisfaction of chance constraints does not provide a direct guarantee on the true system (as it either meets the constraint or not; there is no inherent stochasticity in the system).

The MPC formulation, posed in (9.5), could be vastly improved by updating the prior $f(\theta)$ based on measurements as opposed to reusing the same prior at every step. In theory, this update should be performed using Bayes' rule based on all measurements, i.e.,

$$f(\theta|D_k) = \frac{f(y_k|\theta, D_{k-1})f(\theta|D_{k-1})}{f(y_k|D_{k-1})}, \tag{9.7}$$

where y_k are the system measurements and $D_k = \{\cdots, y_0, y_1, \cdots, y_k\}$ is the set of all measurements over time. This route is explored in the rest of this chapter.

9.3 Recursive Bayesian Estimation using Polynomial Chaos

Problem Formulation

Consider a discrete-time linear system of the form

$$x_{k+1} = A(\theta)x_k + B(\theta)u_k, \tag{9.8a}$$

$$y_k = C(\theta)x_k + v_k, \tag{9.8b}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state vector, $u_k \in \mathbb{R}^{n_u}$ is the input vector, $y_k \in \mathbb{R}^{n_y}$ is the output/measurement vector, $v_k \in \mathbb{R}^{n_y}$ is the measurement noise vector, and $\theta \in \mathbb{R}^{n_\theta}$ is the set of uncertain parameters. The inputs are assumed to be constrained to a convex set $\mathbb{U}$. The parameters θ and noise v_k are assumed to be random variables with PDFs $f(\theta)$ and $f(v_k)$, respectively. For simplicity, the noise is assumed to be an independent and identically distributed (i.i.d.) random process. Assume that there exists a diffeomorphism $T : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\theta}$ such that $\theta = T\xi$ where $\xi \in \mathbb{R}^{n_\xi}$ is composed of standard random variables [126].

The prior distribution $f(\theta)$ (with support Ω_θ) represents our *confidence* in the true system being at a particular value in Ω_θ. It is assumed that the true plant dynamics are described by (9.8) with parameters $\theta = \theta^\star \in \Omega_\theta$ corresponding to some unknown realization, meaning that the true dynamics are within the set of possible dynamics under consideration.

Polynomial Chaos Spectral Representation

Polynomial chaos, described in Chapter 3, is a spectral approximation method for uncertainty propagation that allows for (i) efficient sampling and approximation of a PDF, (ii) analytic expressions for moments of a random variable, and (iii) a deterministic reformulation of a stochastic process. The truncated polynomial chaos

expansion (PCE) can be used to approximate θ

$$\theta(\xi) \approx \hat{\theta}(\xi) = \sum_{i=0}^{L} a_i \Phi_i(\xi), \tag{9.9}$$

where a_i are the coefficients, $\Phi_i(\xi)$ are the polynomial basis functions chosen based on the distribution $f(\xi)$, and $L+1 = \frac{(n_\xi+d)!}{n_\xi! d!}$ is the total number of terms in the expansion with d being the maximum order of the polynomial basis functions retained in the truncated expansion. This basis must satisfy the orthogonality condition

$$\mathbb{E}[\Phi_i \Phi_j] = \langle \Phi_i, \Phi_j \rangle = \int_{\Omega_\xi} \Phi_i(\xi)\Phi_j(\xi) f(\xi) d\xi = \langle \Phi_i^2 \rangle \delta_{ij}, \tag{9.10}$$

where Ω_ξ is the support of ξ. We define the vector of PCE coefficients in (9.9) to be $\mathbf{a} = (a_0, \cdots, a_L)$. As detailed in the previous chapters of this thesis, the moments of θ can be approximated using (9.9). A general expression for the PCE approximation of the r^{th} moment for the j^{th} element of θ, denoted as $\mathscr{M}_j^r$, is given by

$$\mathbb{E}[\theta_j^r] \approx \mathbb{E}[\hat{\theta}_j^r] = \sum_{i_1=0}^{L} \cdots \sum_{i_r=0}^{L} a_{i_1,j} \cdots a_{i_r,j} \langle \Phi_{i_1}, \cdots, \Phi_{i_r} \rangle \triangleq \mathscr{M}_j^r(\mathbf{a}), \tag{9.11}$$

where $a_{i,j}$ is the j^{th} element of vector a_i (corresponds to the i^{th} PCE coefficient of the j^{th} element of θ). The orthogonality condition (9.10) greatly simplifies the computation of the moment expressions $\mathscr{M}_j^r$, which can be calculated as a function of the coefficients $\mathbf{a}$ offline.

Approximate Bayesian Estimation using Moment Matching

We now look to use PCE as a tool for solving (9.7) approximately. The approach we use involves approximating the prior distribution $f(\theta|D_{k-1})$ and the posterior distribution $f(\theta|D_k)$ with PCEs. Since the parameters are assumed to be time-invariant, the posterior distribution at time step $k-1$ is equal to the prior used at time step k. This allows for the problem to be solved recursively while initializing the prior as $f(\theta|D_{-1}) = f(\theta)$, which could be based on past measurements and/or operator

knowledge of the process. Since v_k is assumed to be an i.i.d. process, the likelihood reduces to $f(y_k|\theta, D_{k-1}) = f(y_k|\theta)$ as the measurements $\{y_k\}$ are independent of one another in this conditional space.

Denote the vector of PCE coefficients for $f(\theta|D_{k-1})$ and $f(\theta|D_k)$ as $\mathbf{a}_{k-1}$ and $\mathbf{a}_k$, respectively. At a given time step k, the unknowns in the approximated version of (9.7) are then $\mathbf{a}_k \in \mathbb{R}^{n_\theta(L+1)}$. One way to solve for these unknowns is to determine $\mathbf{a}_k$ that ensures the moments of the right-hand side (RHS) of (9.7) best match the moments of the left-hand side (LHS) of (9.7). Analytic expressions for the moments of the LHS are given by (9.11), which are found once offline. At least $n_\theta(L+1)$ moments $\mathscr{M}_j^r(\mathbf{a}_k)$ need to be determined so that there are more data points than unknowns.

The RHS is more challenging to evaluate analytically in general because of the product of distributions. The likelihood function $f(y_k|\theta)$ expresses the probability of measuring a particular y_k given θ, but is utilized within Bayes' theorem by substituting the measured y_k values as a function of θ. Recalling (9.8b), the likelihood can be stated as $f(y_k|\theta) = f(v_k = y_k - C(\theta)x_k)$ where x_k is a function of θ, some known initial condition x_0, and input sequence $u_0, \cdots, u_{k-1}$. The evidence $f(y_k|D_{k-1})$ is a normalizing factor that can lead to computational challenges in Bayesian estimation problems since marginalization is involved, i.e.,

$$f(y_k|D_{k-1}) = \int_{\Omega_\theta} f(y_k|\theta) f(\theta|D_{k-1}). \tag{9.12}$$

Closed-form solutions exists in special cases when the prior and likelihood are conjugate (for example, the Kalman filter). Monte Carlo integration methods are suitable for approximating this integral since sampling PCEs are cheap as discussed previously. Assuming N_{samp} samples of the standard random variables $\xi \sim f(\xi)$ are drawn, denoted by $\xi^{(j)}$ for $j = 1, \cdots, N_{samp}$, we can approximate the j^{th} sample of θ at time

k, i.e., samples of θ from the distribution $f(\theta|D_k)$), using the PCE from (9.9)

$$\theta_k^{(j)} \approx \hat{\theta}_k^{(j)} = \sum_{i=0}^{L} a_{i,k}\Phi_i(\xi^{(j)}), \tag{9.13}$$

where $a_{i,k}$ is the i^{th} PCE coefficient at time k, which make up the elements of $\mathbf{a}_k$. The numerical approximation of the evidence can now be written as

$$f(y_k|D_{k-1}) \approx \frac{1}{N_{samp}} \sum_{j=1}^{N_{samp}} f(y_k|\theta_{k-1}^{(j)}). \tag{9.14}$$

Moments of the RHS of (9.7) can be evaluated using a similar Monte Carlo approach by combining these results. The r^{th} moment of the j^{th} element of $\theta \sim f(\theta|D_k) = \frac{f(y_k|\theta)f(\theta|D_{k-1})}{f(y_k|D_{k-1})}$ numerically approximated from the RHS of (9.7) is denoted by m_j^r.

We can now write a nonlinear least-squares (NLS) optimization problem to find the $\mathbf{a}_k$ that ensures the LHS moments $\mathscr{M}_j^r$ best match the RHS moments m_j^r

$$\min_{\mathbf{a}_k} \sum_j \sum_r \left(m_j^r - \mathscr{M}_j^r(\mathbf{a}_k)\right)^2, \quad \text{s.t.} \quad H\mathbf{a}_k \leq k, \tag{9.15}$$

where (H, k) are linear constraints that could be included to ensure physicality of the parameter values. At least $n_\theta(L+1)$ terms should be included in the objective to ensure the problem is not underdetermined. Note that joint moments (e.g., $\mathbb{E}[\theta_i\theta_j]$ where θ_i is the i^{th} component of θ) can easily be incorporated into this optimization problem. These joint moments are calculated in the same manner as (9.11). These moments should be included when strong correlation is expected between parameters. Also note that the number of PCE coefficients $L+1$ considered can be significantly reduced in certain situations. An example is when all the cross-terms are assumed to be zero, for which the number of PCE coefficients in $\mathbf{a}_k$ (and thus number of moments that one needs to consider) is reduced from $n_\theta(L+1)$ to $n_\theta(d+1)$ where d is the highest order of the polynomial basis functions retained in the truncated expansion

The solution to the NLS problem (9.15) is denoted by $\mathbf{a}_k^\star$, which must be determined online. As mentioned previously, the state vector appears in the likelihood

function evaluation. By assumption, the state is a random process driven only by θ. As such, we apply the Galerkin projection method (discussed in detail in Chapters 6 and 8) to (9.8) (where the noise is negelected as it is zero-mean) [199]

$$\mathbf{X}_{k+1} = \mathbf{A}(\mathbf{a}_k)\mathbf{X}_k + \mathbf{B}(\mathbf{a}_k)u_k, \tag{9.16a}$$

$$\mathbf{Y}_k = \mathbf{C}(\mathbf{a}_k)\mathbf{X}_k, \tag{9.16b}$$

where $\mathbf{X}_k$ and $\mathbf{Y}_k$ are the concatenated vectors of the PCE coefficients of the state and output vectors, respectively, and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are the projected matrices (see Chapters 6 and 8 for derivations and explicit expressions for these matrices). The key thing to note is that $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are functions of the PCE coefficients $\mathbf{a}_k$ since (9.9) is substituted for θ in the expressions for $A(\theta)$, $B(\theta)$, and $C(\theta)$. Note that the recursively updated posterior distribution $f(\theta|D_k)$, parametrized by coefficients $\mathbf{a}_k$, is always used to update the state distribution in (9.16) so this is also always being updated with the most recent measurements.

9.4 Output Feedback MPC with Bayesian Learning

The concept of recursive Bayesian estimation using polynomial chaos, introduced in the previous section, can be straightforwardly incorporated into the many possible variations of stochastic MPC algorithms introduced and referenced in Chapters 6–8. Integral action can be attained by rewriting the model in a *velocity form* as discussed in Chapter 4.

Chance constraints can be enforced using a number of different methods including the Cantelli-Chebyshev inequality. These chance constraints do not have a physical interpretation in terms of frequency of occurrence since the process is only run once indefinitely (i.e., it is assumed to be a deterministic, but unknown process) and the parameters are assumed to be time-invariant (as such the true process either satisfies the constraint or it violates the constraint). However, using the Bayesian framework presented in the previous section, we can impose constraints based on our confidence

(aka subjective belief) described by a PDF. This approach is more rigorous than many standard receding horizon control methods in the form of (9.5) as highlighted by Theorem 9.1. An example problem illustrating some of these points is presented next.

9.5 Example: Reactors in Series

The proposed algorithm of SMPC with Bayesian learning is applied to a reactor problem consisting of two continuously stirred rank reactors (CSTRs) and a nonadiabatic flash [254] as illustrated in Figure 9-1. In each CSTR the reaction $A \rightarrow B \rightarrow C$ occurs, where B is the desired product. The system is nonlinear with twelve states (liquid level, temperature, mole fraction A, B for each subsystem), six inputs (flow rates F_I, F_{II}, D and cooling loads Q_r, Q_m, Q_b) and five outputs (all temperatures, liquid level H_b and mole fraction $x_{B,b}$). The initial conditions of the system are considered to be known. The system is linearized around a steady state and discretized using a zero-order hold with a sampling time of 0.2 s. Two valve parameters $\theta_1 = k_r$ and $\theta_2 = k_m$ are assumed to be unknown with $\theta_1^* = \theta_2^* = 2.5$. The valves are mounted on the pipes connecting the CSTRs and the second CSTR with the flash. Lagrange polynomials with $d = 4$ are used, because θ_1 and θ_2 are each distributed uniformly between $[2.0, 2.75]$ upon initialization.

The MPC objective function for this problem was seleced to penalize weighted deviations between the mean value and a setpoint for each state, weighted penalties on the variance of each state, and weighted penalties on the change in input values. The mean deviation from the setpoint and variance weights were selected to be 1 for all states while the input flow rate deviations weights were chosen to be 1 and input cooling load weights were chosen to be 0.1. The measurement noise v_k was selected to be a zero-mean normal distribution with covariance matrix equal to $\text{diag}(1/3^2, 1/3^2, 1/3^2, 0.005^2/3^2, 0.05^2/3^2)$. The prediction and control horizons were both selected to be 10. Constraints on the input and change in input values were also considered. Readers are referred to [176] for further details.

Figure 9-2 shows the PCE-approximated histograms of the estimates for each parameter θ_1 and θ_2. No significant improvement is observed for $t > 4$, which is due to measurement noise in the system. The calculated inputs are depicted in Figure 9-3 and the resulting states in Figure 9-4. In the beginning, the flow rate constraints are enforced, leading to fast regulation of the heights. Simultaneously, the cooling is adjusted such that the temperatures are regulated to the steady state. Because the mole fractions are weakly coupled to the inputs and exhibit fairly large time constants, it takes considerably longer for them to return to their steady state values. On a Windows 7 machine with 16 GB of RAM it took less than 1 ms to solve each NLS problem (9.15) using MEIGO [64] while each MPC problem (formulated as a QP) was solved in on the order of 1 ms using qpOASES [69].

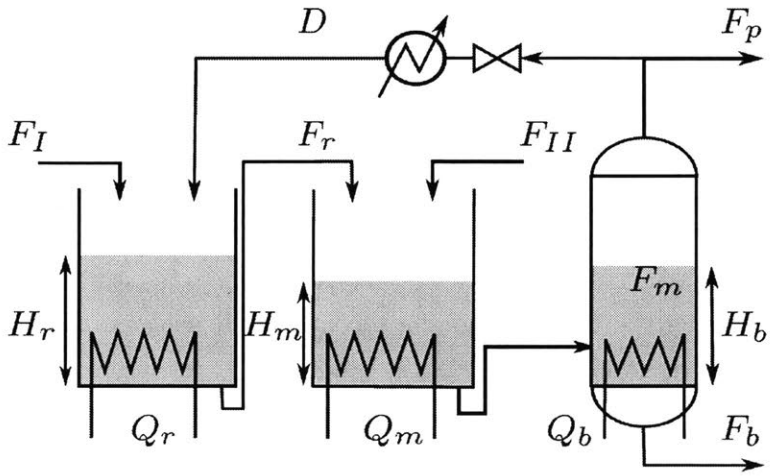


Figure 9-1: Cascade of two CSTRs with reaction $A \to B \to C$ and nonadiabatic flash with purge and recycle, e.g., [254].

9.6 Conclusions

This chapter considers the design of an efficient output feedback control strategy combining stochastic model predictive control and recursive Bayesian state and parameter estimation. This is motivated by the fact that standard receding horizon control methods fail to give probabilistic guarantees even when chance constraints

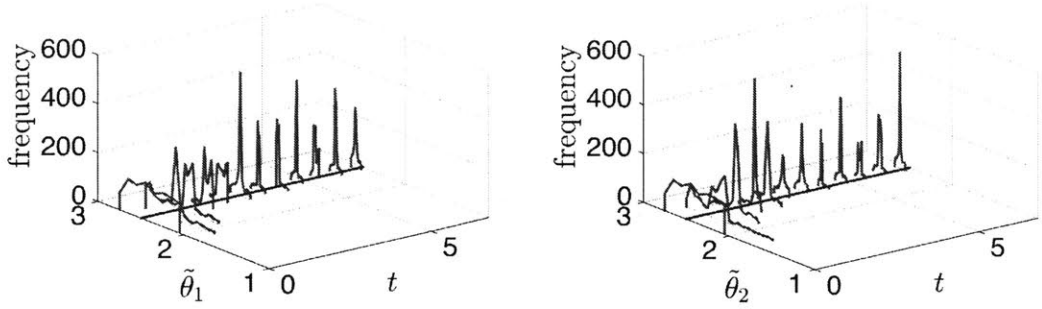


Figure 9-2: Histograms of PDF ($M = 1000$) of parameters $\tilde{\theta}_1$ and $\tilde{\theta}_2$ for cascaded CSTRs and flash over time t with $\theta_1^* = \theta_2^* = 2.5$ depicted as solid line.

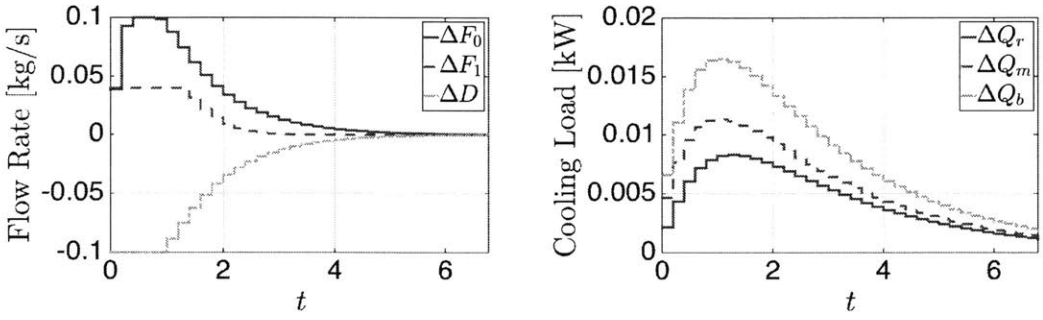


Figure 9-3: Stochastic MPC inputs (as deviations from steady state) to cascaded CSTRs and flash.

are satisfied in each open-loop problem. This is proven by counterexample and highlights viewing the parameter uncertainty as a confidence (aka quantification of subjective belief). Stochastic noise, probabilistic parametric uncertainty, and hard input constraints are considered. Uncertainties are incorporated in the form of probability density functions. Adopting the Bayesian framework, the uncertain parameters are treated as a realization of a random vector. Consequently, the controlled plant becomes a realization of a stochastic process. The efficient propagation of uncertainties is addressed using polynomial chaos. The performance and efficiency of the approach is underlined considering a reactor example.

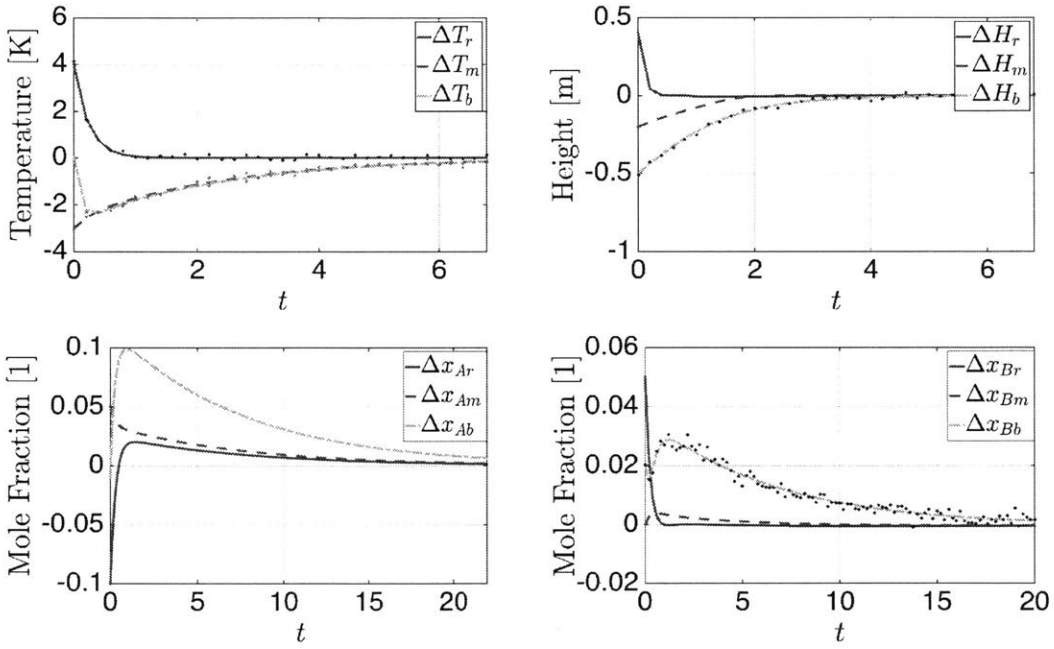


Figure 9-4: States (as deviations from steady state) of cascaded CSTRs and flash under inputs from Figure 9-3. Measurements depicted as dots.

Part V

Fault/Failure Tolerant Process Control

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 10

Multi-objective Failure Tolerant Controller Design

10.1 Introduction

The design of the control structure, which is the specification of the interconnection of measurements, exogenous inputs, and manipulated variables, greatly influences the performance achievable by a control system. In practice, control systems are usually required to fulfill multiple objectives that are poorly described using a single performance measure as commonly proposed in the literature. This realization has led to the development of numerous control structures that have multiple degrees-of-freedom (e.g., see [92, 36, 144, 206, 283, 58, 256, 146] and the citations therein), where each controller degree-of-freedom is tasked with addressing some subset of the control objectives. As failures in system components inevitably occur in practice, an important practical consideration is to design control structures and their associated controllers to have graceful performance degradation during component failures [30, 112, 281].

Internal model control (IMC) is a control design method developed in the 1970s–1980s with several useful features, including that it provides a convenient theoretical framework for the design of two degrees-of-freedom control systems [175]. The basic idea is to combine an optimal controller obtained from the nominal process model

with a low-pass filter to tradeoff closed-loop performance with robustness to model uncertainties. Another feature of the IMC structure is that it simplifies the task of controller design by employing the Youla parameterization to write the nominal closed-loop transfer functions as an affine function of the to-be-designed controller(s). The IMC structure can also be implemented in the a manner that ensures internal nominal stability of the closed-loop system in the presence of actuator constraints [175, 33].

This chapter presents a systematic procedure for the design of multiple degrees-of-freedom controllers based on an extension of the internal model control design method. The cornerstone of the design procedure is a general control structure that enables separate formulation of control objectives for each exogenous input. This leads to independent design of multi-objective controllers with optimal failure tolerance, as the global optimality of the control system is preserved when a controller(s) is taken off-line (e.g., due to actuator and/or sensor failures). The control approach is shown to alleviate most of the tradeoffs inherent in a classical feedback control structure without compromising on the best achievable performance. The approach is extended to multi-loop control systems by deriving the most general control structure for multi-loop cascade and coordinated control systems. In addition, analytical expressions are presented for the design of H_2-optimal controllers for single-input single-output (SISO) systems that have objectives associated with tracking reference trajectories, rejection of measured and unmeasured load and output disturbances, and suppression of measurement noise.

Notation and Preliminaries Throughout the chapter, a finite-dimensional multi-input multi-output process is denoted by $P(s) \in \mathcal{RH}_\infty$ where s is the Laplace variable and $\mathcal{RH}_\infty$ denotes the real rational subspace of $\mathcal{H}_\infty$ consisting of all proper and rational stable transfer matrices. The exogenous inputs $r(t)$, $l_m(t)$, $l_u(t)$, $d_m(t)$, and $d_u(t)$ are bounded signals (i.e., $r(t), l_m(t), l_u(t), d_m(t), d_u(t) \in \mathcal{L}_p[0,\infty)$, where $\mathcal{L}_p[0,\infty)$ encompasses all signal sequences on $[0,\infty)$ which have finite p-norm). The real-valued function $\|\cdot\|$ denotes any norm defined over the linear vector space of the signals. The

induced system norm $|\cdot|$ is defined as the supremum of the output signal norm over a norm-bounded set of input signals [282].

Definition 10.1 (Internal Stability [175]) *A continuous-time linear time-invariant (LTI) closed-loop system is* ***internally stable*** *if the transfer functions between any two points of the closed-loop system are stable (have all poles in the open left-half plane).*

Definition 10.2 (Robust Stability [175]) *A closed-loop system is* ***robustly stable*** *if the controller C ensures the internal stability of the closed-loop system for all $P \in \mathcal{P}$, where $\mathcal{P}$ is the set of uncertain processes.*

10.2 Systematic Design of Multi-objective Controllers

The fundamental questions central to the design of a control strategy can be summarized as:

1. Control structure: Does the control structure limit the achievable performance?
2. Controller design: Do the closed-loop performance measures reflect the control objectives?
3. Controller implementation: Is it feasible to realize all the control objectives given the available degrees of freedom?

This section addresses each of these questions in order.

Control Structure

The most general control structure for a process P with manipulated variable u, reference r, measured load disturbance l_m, unmeasured load disturbance l_u, measured output disturbance d_m, unmeasured output disturbance d_u, and measurement noise n is shown in Figure 10-1[1]. All variables that can be measured are fed directly into the

[1]To simplify the analytical expressions, explicit transfer functions for the various disturbances are not shown; the generalization of the results of this chapter to include such transfer functions is straightforward.

controller $\mathbf{C}$ that is to be designed to ensure (i) internal stability of the closed-loop system, (ii) the output y closely tracks the reference r (i.e., small error $e = y - r$), and (iii) the effects of the measurement noise and measured and unmeasured disturbances on the closed-loop error e are suppressed. The mapping between all of the inputs to the closed-loop system and the process output and manipulated variable is given by

$$\begin{bmatrix} y \\ u \end{bmatrix} = H(P, \mathbf{C}) \begin{bmatrix} r \\ l_m \\ l_u \\ d_m \\ d_u \\ n \end{bmatrix} \tag{10.1}$$

where the transfer matrix $H(P, \mathbf{C})$ is

$$\begin{bmatrix} PC_r(I+PC_y)^{-1} & PC_{lm}(I+PC_y)^{-1} + (I+PC_y)^{-1}P & (I+PC_y)^{-1}P \\ C_r(I+PC_y)^{-1} & C_{lm}(I+PC_y)^{-1} - C_y(I+PC_y)^{-1}P & -C_y(I+PC_y)^{-1}P \end{bmatrix}$$

$$\begin{bmatrix} PC_{dm}(I+PC_y)^{-1} + (I+PC_y)^{-1} & (I+PC_y)^{-1} & -PC_y(I+PC_y)^{-1} \\ C_{dm}(I+PC_y)^{-1} - C_y(I+PC_y)^{-1} & -C_y(I+PC_y)^{-1} & -C_y(I+PC_y)^{-1} \end{bmatrix}. \tag{10.2}$$

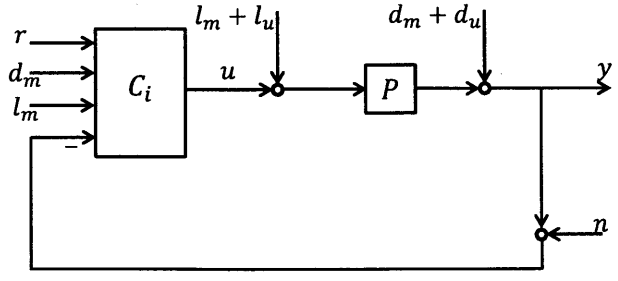


Figure 10-1: General classical feedback control structure.

A standard approach in optimal control is to formulate a single performance measure in terms of an overall norm such as a weighted H_2 or H_∞-norm on $H(P, \mathbf{C})$ [282]. However, a drawback of this approach is that a typical control problem has multiple objectives that are independently defined in terms of relationships between specific inputs and specific outputs. Several of the closed-loop transfer functions in (10.2) that relate the system inputs to the output y and manipulated variable u are functions of multiple controller transfer functions, so that the designs of these controller transfer functions to satisfy multiple independently defined control objectives are not independent. Next, an alternative control structure is presented that is provably general while having each term in the relationship between an input and output being a function of only one controller transfer function.

Consider the internal model control structure in Figure 10-2 for the formulation of the control design problem where $\mathbf{Q}$ is the IMC controller. For the class of stable linear time-invariant systems, Theorem 10.1 states that Figure 10-2 provides a non-restrictive control structure for the design of multi-objective controllers.

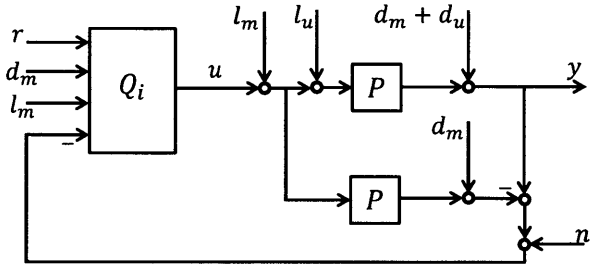


Figure 10-2: General control structure for a stable LTI system.

Theorem 10.1 *Consider a stable LTI system P with measured output y, manipulated input u, measurement noise n, and disturbances l_m, l_u, d_m, and d_u. The IMC structure in Figure 10-2 is the most general LTI control structure for the design of multiple degrees-of-freedom control systems that internal stabilize the closed-loop system.*

Proof. Let $P = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ where $\{M, N\}$ and $\{\tilde{M}, \tilde{N}\}$ are right and left coprime factorizations of P over $\mathcal{RH}_\infty$, respectively. Define $C_{y0} = UV^{-1} = \tilde{V}^{-1}\tilde{U}$ as a stabilizing controller such that $\tilde{V}$ and $\tilde{U}$ satisfy the Bezout identity $\tilde{V}M + \tilde{U}N = I$.

It is well-known that all stabilizing LTI feedback controllers are parameterized by

$$C_y = (\tilde{V} - Q_y\tilde{N})^{-1}(\tilde{U} + Q_y\tilde{M}) \tag{10.3}$$

where Q_y is any stable LTI transfer function [282]. For a stable LTI process ($P \in \mathcal{RH}_\infty$) choose $\tilde{N} = P$, $\tilde{M} = I$, $\tilde{U} = 0$, and $\tilde{V} = I$ [175]. This results in the Youla parameterization of all stabilizing feedback controllers for a process P [274]:

$$C_y = (I - Q_yP)^{-1}Q_y = Q_y(I - PQ_y)^{-1}. \tag{10.4}$$

Insertion of this equation into (10.2) defines the set of all possible LTI closed-loop transfer functions that are internally stable. Hence, $H(P, \mathbf{C})$ takes the form

$$\begin{bmatrix} PC_r(I + PC_y)^{-1} & PC_{lm}(I + PC_y)^{-1} + (I + PC_y)^{-1}P & (I - PQ_y)P \\ C_r(I + PC_y)^{-1} & C_{lm}(I + PC_y)^{-1} - C_y(I + PC_y)^{-1}P & -Q_yP \\ PC_{dm}(I + PC_y)^{-1} + (I + PC_y)^{-1} & I - PQ_y & -PQ_y \\ C_{dm}(I + PC_y)^{-1} - C_y(I + PC_y)^{-1} & -Q_y & -Q_y \end{bmatrix}. \tag{10.5}$$

The expression in (10.5) implies that internal stability of the overall closed-loop system requires that the transfer functions

$$\begin{aligned} &C_r(I + PC_y)^{-1} \\ &C_{lm}(I + PC_y)^{-1} \\ &C_{dm}(I + PC_y)^{-1} \end{aligned} \tag{10.6}$$

are stable[2]. Define

$$\begin{aligned} Q_r &= C_r(I + PC_y)^{-1} \\ Q_{lm} &= C_{lm}(I + PC_y)^{-1} - C_y(I + PC_y)^{-1}P \\ Q_{dm} &= C_{dm}(I + PC_y)^{-1} - C_y(I + PC_y)^{-1}. \end{aligned} \tag{10.7}$$

It follows from the definition of C_y (see (10.4)) that C_r, C_{lm}, and C_{dm} in (10.7) encompass the set of all stabilizing feedback controllers for any stable Q_r, Q_{lm}, and Q_{dm}, respectively. This implies that (10.5) is internally stable.

Insertion of (10.7) into (10.5) simplifies the transfer matrix $H(P, \mathbf{C})$ to

$$\begin{bmatrix} PQ_r & P(I + Q_{lm}) & (I - PQ_y)P & I + PQ_{dm} & I - PQ_y & -PQ_y \\ \\ Q_r & Q_{lm} & -Q_yP & Q_{dm} & -Q_y & -Q_y \end{bmatrix}. \tag{10.8}$$

The block diagram in Figure 10-2 has the same closed-loop transfer matrix as in (10.8) and, hence, is equivalent to the general classical feedback control system in Figure 10-1. The control structure in Figure 10-2 is non-restrictive for stable LTI systems since it entails the set of all stabilizing controllers $\mathbf{C}$ for any stable $\mathbf{Q}$. □

The control structure in Figure 10-2 can be used for IMC implementation [175] by replacing the lower P in Figure 10-2 with a process model $\tilde{P}$. This structure is an extension of the IMC structure to systems with four degrees of freedom. Theorem 10.1 indicates that the proposed IMC control structure does not restrict the set of closed-loop transfer functions that ensure internal stability of the closed-loop system. A consequence of this result is that the use of the IMC control structure does not limit the achievable closed-loop performance, regardless of the closed-loop performance measure(s) used to encode the control objectives. These characteristics are in contrast to most of the control structures that have been proposed for control systems with multiple degrees-of-freedom (e.g., see [232] and the references therein). Next, we

[2]The fact that $(I + PC_y)^{-1}$ is stable for all stable Q_y indicates that the only unstable poles allowed in C_r, C_{lm}, and C_{dm} for internal stability must also be unstable zeros of $(I + PC_y)^{-1}$.

discuss how multiple control objectives can be met by independent controllers design.

Controller Design and Implementation

The control structure in Figure 10-2 provides a convenient framework for the design of multi-objective controllers because the closed-loop transfer matrix (10.8) depends on each of the controllers in an affine manner, and all columns depend on only one controller. Hence, the optimal controllers Q_r, Q_{lm}, Q_{dm}, and Q_y for multiple objectives between the system inputs and outputs can be designed independently.

Based on the closed-loop mapping obtained from (10.8) between $[e, u]^T$ and each of the system inputs, the following multiple control objectives can be defined:

- Reference tracking:

$$\inf_{Q_r} \left\| \begin{bmatrix} PQ_r - I \\ Q_r \end{bmatrix} r \right\|. \tag{10.9}$$

- Measured load disturbance rejection:

$$\inf_{Q_{lm}} \left\| \begin{bmatrix} P(I + Q_{lm}) \\ Q_{lm} \end{bmatrix} l_m \right\|. \tag{10.10}$$

- Unmeasured load disturbance rejection:

$$\inf_{Q_y} \left\| \begin{bmatrix} (I - PQ_y)P \\ -Q_yP \end{bmatrix} l_u \right\|. \tag{10.11}$$

- Measured output disturbance rejection:

$$\inf_{Q_{dm}} \left\| \begin{bmatrix} I + PQ_{dm} \\ Q_{dm} \end{bmatrix} d_m \right\|. \tag{10.12}$$

- Unmeasured output disturbance rejection:

$$\inf_{Q_y} \left\| \begin{bmatrix} I - PQ_y \\ -Q_y \end{bmatrix} d_u \right\|. \tag{10.13}$$

- Measurement noise suppression:

$$\inf_{Q_y} \left\| \begin{bmatrix} -PQ_y \\ -Q_y \end{bmatrix} n \right\|. \tag{10.14}$$

In practice, a single signal

$$v = Pl_u + d_u \tag{10.15}$$

is used to represent the combined effect of the unmeasured load and unmeasured output disturbances. This implies that the two performance measures (10.11) and (10.13) can be replaced with a single expression

$$\inf_{Q_y} \left\| \begin{bmatrix} I - PQ_y \\ -Q_y \end{bmatrix} v \right\|. \tag{10.16}$$

As such, the above norm expressions are applicable to both continuous-time and discrete-time transfer functions and for different norms used as performance measures. For each control objective, an alternative performance measure is to replace the signal norm with an induced system norm. For example, the measurement noise suppression objective is often expressed in terms of its induced system norm as

$$\inf_{Q_y} \left\| \begin{bmatrix} -PQ_y \\ -Q_y \end{bmatrix} \right\|. \tag{10.17}$$

The performance measures in (10.9)–(10.14) indicate that Q_r, Q_{lm}, and Q_{dm} only influence one control objective. Hence, the latter controller transfer functions can be designed independently of each other and independently of Q_y. The only tradeoff in each of the designs of Q_r, Q_{lm}, and Q_{dm} is that fast speed of response (the effect on y)

will be associated with faster and larger changes in the manipulated variable u. This tradeoff can be implemented by placing a weight on one or both of the closed-loop error e and manipulated variable u signals. In addition, the design of Q_y requires prioritizing the multiple control objectives in view of their importance, as Q_y is the only controller to suppress the effects of unmeasured disturbances and measurement noise (see expressions (10.14) and (10.16)).

Section 10.5 illustrates the design of H_2-optimal controllers for a general class of stable SISO systems. Next, we discuss how the control structure in Figure 10-2 enables controllers to be implemented such that they lead to optimal achievable performance in the presence of system failures.

10.3 Design of Failure-tolerant Controllers

A common approach to failure-tolerant control is to design a single control system using robust control techniques to deal with all potential actuator and/or sensor failures (e.g., see the discussion in [283]). Since this approach designs the control system for the worst-case performance, it may lead to very conservative performance when no actuator and/or sensor failures occur.

The special feature of the proposed control structure that the controllers Q_r, Q_{lm}, Q_{dm}, and Q_y are designed independently of each other is particularly significant for failure-tolerant control when a system component (actuator or sensor) needs to be taken out of service due to a failure. The design of the controllers in Section 10.2 can be posed as the multi-objective optimization

$$\inf_{\mathbf{Q}} \{w_1 F_1(Q_r) + w_2 F_2(Q_{lm}) + w_3 F_3(Q_{dm}) + w_4 F_4(Q_y)\} \tag{10.18}$$

where $w_k \geq 0$ are weights and the objective functions F_k are the signal or induced system norms of the columns of the closed-loop mapping between $[e, u]^T$ and the system inputs (see (10.9)–(10.17)) for $k = 1, 2, 3, 4$. In (10.18), the vector $\mathbf{Q}$ consists of the to-be-designed controllers Q_r, Q_{lm}, Q_{dm}, and Q_y.

Theorem 10.2 *Consider the multi-objective optimization* (10.18). *The solution to the optimization remains globally optimal as the controller(s)* Q_r, Q_{lm}, Q_{dm}, *and/or* Q_y *and the respective objective function(s)* F_1, F_2, F_3, *and/or* F_4 *are eliminated from the multiple-objective function in* (10.18).

Proof. Let the optimal solution to the convex optimization (10.18) be

$$\mathbf{Q}^* = \arg\min_{\mathbf{Q}} \{w_1 F_1(Q_r) + w_2 F_2(Q_{lm}) + w_3 F_3(Q_{dm}) + w_4 F_4(Q_y)\}, \tag{10.19}$$

where $\mathbf{Q}^\star = [Q_r^\star, Q_{lm}^\star, Q_{dm}^\star, Q_y^\star]$. Since each objective function depends on only one decision variable (either Q_r, Q_{lm}, Q_{dm}, or Q_y), the optimal solution to every objective function is independent of the other objective functions, i.e.,

$$Q_r^* = \arg\min_{Q_r} F_1(Q_r), \quad Q_{lm}^* = \arg\min_{Q_{lm}} F_2(Q_{lm}),$$
$$Q_{dm}^* = \arg\min_{Q_{dm}} F_3(Q_{dm}), \quad Q_y^* = \arg\min_{Q_y} F_4(Q_y).$$

Hence, when any of the objectives is eliminated from the multiple-objective function, the optimal solution remains the same as that in (10.19) for the rest of the remaining controllers. □

Theorem 10.2 indicates that when optimal control is used to design each controller independently, the rest of the controllers remain optimal if one or more of the other controllers are taken out of service (set to 0) due to an actuator or sensor failure. In other words, the overall control system remains optimal for the multiple achievable objectives when any controller is taken out of service. This implies that the remaining controllers need not be redesigned to realize optimal failure-tolerant control. Such an approach to optimal failure tolerance leads to vastly superior performance under most conditions than designing a controller to optimize the worst-case performance for all possible failure conditions.

The proposed control structure also possesses a distinct feature for fault-tolerant control when abnormal system operation results from a change in process dynamics and/or disturbance characteristics. When a change in process dynamics can be

characterized by model uncertainty, the lower P in Figure 10-2 is replaced with a process model $\tilde{P} \in \mathcal{P}$. In this case, the internal stability of the closed-loop system in Figure 10-2 for any $\tilde{P} \in \mathcal{P}$ depends on only Q_y.

Theorem 10.3 *For stable Q_r, Q_{lm}, and Q_{dm}, robust stability of the closed-loop system in Figure 10-2 depends on only Q_y, $\tilde{P}$, and $\mathcal{P}$, where $\mathcal{P}$ is the set of uncertain processes. In particular, robust stability does not depend on Q_r, Q_{lm}, or Q_{dm}.*

Proof. Suppose that the process dynamics are described by the model $\tilde{P}$ that belongs to the uncertainty set $\mathcal{P}$. Replace the lower P in the general control structure of Figure 10-2 with $\tilde{P}$. The closed-loop transfer matrix $H(P, \mathbf{Q})$ for the mapping between $[y, u]^T$ and the system inputs (see (10.1)) takes the form

$$\begin{bmatrix} PQ_rS & PQ_{lm}S + (I - \tilde{P}Q_y)SP & (I - \tilde{P}Q_y)SP & PQ_{dm}S + (I - \tilde{P}Q_y)S \\ Q_rS & Q_{lm}S - Q_ySP & -Q_ySP & Q_{dm}S - Q_yS \end{bmatrix}$$
$$\begin{bmatrix} (I - \tilde{P}Q_y)S & -PQ_yS \\ -Q_yS & -Q_yS \end{bmatrix} \tag{10.20}$$

where S is

$$S = \left(I + (P - \tilde{P})Q_y\right)^{-1}. \tag{10.21}$$

Robust stability of the control structure in Figure 10-2 requires that all transfer functions in (10.20) are stable for any $\tilde{P} \in \mathcal{P}$. For stable Q_r, Q_{lm}, and Q_{dm}, (10.20) indicates that the robust stability of the closed-loop system is determined by the stability of S. Hence, the robust stability of the system in Figure 10-2 depends on only Q_y, $\tilde{P}$, and $P \in \mathcal{P}$. □

Theorem 10.3 implies that any stable Q_r, Q_{lm}, and Q_{dm} do not influence the internal stability of the closed-loop system in the presence of model uncertainties. This observation motivates an approach similar to [283] in which Q_y can be designed to be robust to some modest fault conditions as well as to model uncertainties, while the rest

of the controllers are designed independently of these fault conditions. The control structure therefore alleviates the need for redesigning Q_r, Q_{lm}, and Q_{dm} when faults occur in the closed-loop system. What distinguishes the proposed control structure from that presented in [283] is its ability to realize optimal failure-tolerant control for multi-objective controllers.

10.4 Design of Multi-objective Controllers for Multi-loop Systems

Cascade Control

Cascade control systems are commonly used in chemical industry to improve the dynamic response of the closed-loop system by effectively reducing the impact of process disturbances, in particular load disturbances [150, 132]. A cascade control system is typically designed to tightly control a secondary process variable that closely relates to the property of interest and is readily available from on-line measurements, while scarce and delayed measurements of the primary process variable are used to correct the control action. The majority of cascade control systems are of the series type where the manipulated variable influences the primary output through the secondary output (see Figure 10-3a). In general, cascade control is most effective when the dynamics of the primary process exhibit nonminimum phase behavior (right-half plane zeros or a time delay) and the secondary loop has a faster dynamic response [175].

The general structure for cascade control of stable LTI systems is obtained by replacing the plant P in the control structure of Figure 10-2 with

$$\begin{bmatrix} P_1 P_2 \\ P_2 \end{bmatrix}. \tag{10.22}$$

This leads to the control structure depicted in Figure 10-3b, which possesses 7 degrees of freedom. The closed-loop mapping between the exogenous inputs and the two

process outputs and the manipulated variable is defined by

$$\begin{bmatrix} y_1 \\ y_2 \\ u \end{bmatrix} = H(P, \mathbf{Q}) \begin{bmatrix} r \\ l_{m1} \\ l_{m2} \\ l_{u1} \\ l_{u2} \\ d_{m1} \\ d_{m2} \\ d_{u1} \\ d_{u2} \\ n_1 \\ n_2 \end{bmatrix} \tag{10.23}$$

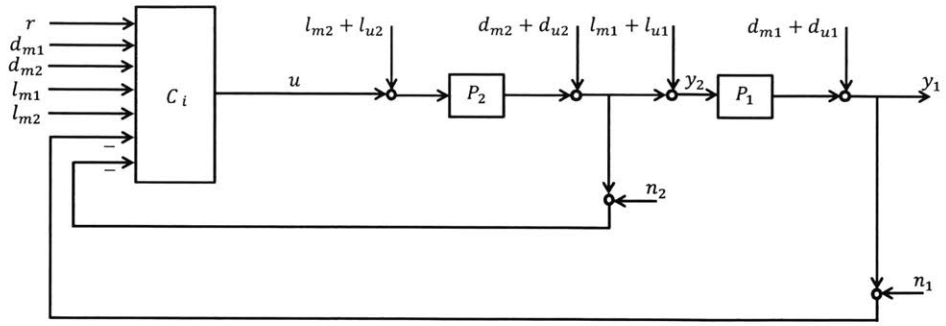
where the transfer matrix $H(P, \mathbf{Q})$ is

$$\begin{bmatrix} P_1P_2Q_r & P_1 + P_1P_2Q_{lm1} & P_1P_2(I + Q_{lm2}) & (I - P_1P_2Q_{y1})P_1 & P_1(I - P_2Q_{y2})P_2 & I + P_1P_2Q_{dm1} \\ P_2Q_r & I + P_2Q_{lm1} & P_2(I + Q_{lm2}) & I - P_2Q_{y1}P_1 & (I - P_2Q_{y2})P_2 & P_2Q_{dm1} \\ Q_r & Q_{lm1} & Q_{lm2} & -Q_{y1}P_1 & -Q_{y2}P_2 & Q_{dm1} \end{bmatrix}$$

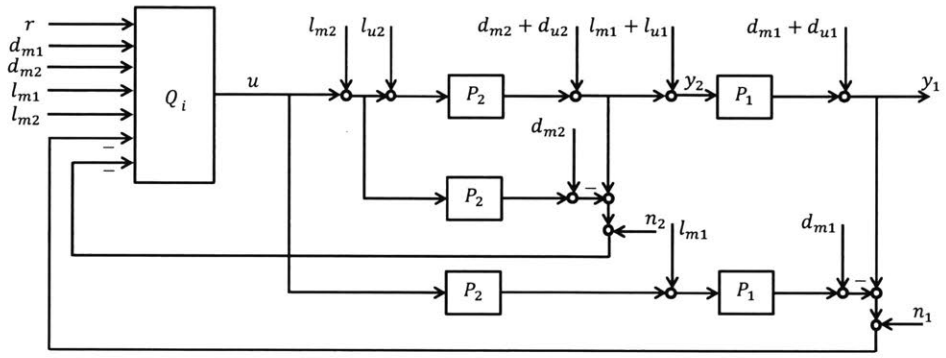
$$\begin{matrix} P_1 + P_1P_2Q_{dm2} & I - P_1P_2Q_{y1} & P_1(I - P_2Q_{y2}) & -P_1P_2Q_{y1} & -P_1P_2Q_{y2} \\ I + P_2Q_{dm2} & -P_2Q_{y1} & I - P_2Q_{y2} & -P_2Q_{y1} & -P_2Q_{y2} \\ Q_{dm2} & -Q_{y1} & -Q_{y2} & -Q_{y1} & -Q_{y2} \end{matrix} \Bigg]. \tag{10.24}$$

Equation (10.24) indicates that each column depends on only one controller transfer function, so that the optimal designs of the controllers can be performed independently. The general cascade control structure in Figure 10-3b extends the IMC cascade structure presented in [175] by including the effects of load disturbances and measured output disturbances on both control loops.

Another widely used variant of cascade control is the parallel cascade structure in which the manipulated variable affects the primary and secondary outputs through parallel actions (see Figure 10-4) [150, 275]. In [229], it was demonstrated (using block diagram transformations) that the parallel cascade structure is equivalent to



(a) Classical structure



(b) General structure

Figure 10-3: Cascade control system.

the series cascade structure when the primary process transfer function is replaced with P_1/P_2. Hence, the general control structure pertaining to parallel cascade control can be obtained by defining the plant in Figure 10-2 as $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$. This implies that in Figure 10-3b and the closed-loop mapping (10.24) P_1 will be replaced with P_1/P_2. Note that the general structure for parallel cascade control holds only when P_2 is minimum phase.

Coordinated Control

Coordinated control typically refers to a class of control problems where two manipulated variables are used to control one output [205, 99, 83, 77]. A general representation of coordinated control systems, also known as mid-ranging control, is depicted

in Figure 10-5a. A coordinate control system is designed such that the manipulated variable u_1, which has a more direct effect on y (faster dynamics and a smaller time delay), rapidly regulates the process output for setpoint and disturbance changes. However, since the manipulation of u_1 is more expensive than u_2, the control system gradually resets the fast input u_1 to its desired setpoint u_r as the slower input u_2 begins to affect the output.

To obtain the general structure for coordinate control of stable LTI systems, the plant P in the control structure of Figure 10-2 is replaced with

$$\begin{bmatrix} P_1 & P_2 \end{bmatrix}. \tag{10.25}$$

The resulting control structure with 6 degrees of freedom has the closed-loop mapping

$$\begin{bmatrix} y \\ u_1 \\ u_2 \end{bmatrix} = H(P, \mathbf{Q}) \begin{bmatrix} r \\ u_r \\ l_{m1} \\ l_{m2} \\ l_{u1} \\ l_{u2} \\ d_m \\ d_u \\ n \end{bmatrix} \tag{10.26}$$

with $H(P, \mathbf{Q})$ defined by

$$\begin{bmatrix} (P_1+P_2)Q_r & (P_1+P_2)Q_{ur} & P_1+(P_1+P_2)Q_{lm1} & P_2+(P_1+P_2)Q_{lm2} & (I-(P_1+P_2)Q_y)P_1 \\ Q_r & Q_{ur1} & Q_{lm1} & Q_{lm2} & -Q_yP_1 \\ Q_r & Q_{ur2} & Q_{lm1} & Q_{lm2} & -Q_yP_1 \end{bmatrix}$$

$$\begin{bmatrix} (I-(P_1+P_2)Q_y)P_2 & I+(P_1+P_2)Q_{dm} & I-(P_1+P_2)Q_y & -(P_1+P_2)Q_y \\ -Q_yP_2 & Q_{dm} & -Q_y & -Q_y \\ -Q_yP_2 & Q_{dm} & -Q_y & -Q_y \end{bmatrix}. \tag{10.27}$$

Equation (10.27) suggests that the optimal controllers, designed for multiple objectives between the system inputs and outputs, in a coordinated control system can be performed independently. Note that the feedforward controller

$$Q_{ur} = \begin{bmatrix} Q_{ur1} \\ Q_{ur2} \end{bmatrix} \tag{10.28}$$

is designed such that $P_1 Q_{ur1} + P_2 Q_{ur2} = 0$.

10.5 H_2-optimal Controllers for SISO Systems

Although the previous results primarily considered finite-dimensional continuous-time LTI systems (that is, with all transfer functions in $\mathcal{RH}_\infty$), the results are also applicable to infinite-dimensional systems and to discrete-time systems (the former by defining the appropriate infinite-dimensional algebra [53] and the latter by replacing the Laplace transform with z-transform and the location of the poles for specifying stability of a transfer function). This section deals with single-input single-output processes with time delays to illustrate the derivation of analytical expressions for optimal controllers for one class of infinite-dimensional systems. The results in this section can be generalized to other infinite-dimensional systems using the mathematical machinery in [166].

In its most general form, a stable LTI single-input single-output process $p(s)$ can be written as

$$p(s) = p_a(s) p_m(s) \tag{10.29}$$

where $p_a(s)$ and $p_m(s)$ are the all-pass and minimum-phase parts of $p(s)$, respectively. $p_a(s)$ includes all the right-half plane zeros as well as time delays of $p(s)$ and generally takes the form

$$p_a(s) = e^{-\theta s} \Pi_i \frac{-s + \zeta_i}{s + \zeta_i^*} \tag{10.30}$$

where the superscript $*$ denotes complex conjugate [175].

Next, optimal controllers in the control structure in Figure 10-2 are designed for

the SISO process (10.29). The control objectives are defined in terms of minimization of the H_2-norm of the error signal (see expressions (10.9)–(10.16)). Since the derivation of analytical expressions for controllers requires partial fraction expansion, the main result of partial fraction expansion is summarized in the following definition.

Definition 10.3 (Partial Fraction Expansion [204]) *Suppose* $A(s) = \prod_{i=1}^{N}(s - \lambda_i)^{n_i}$, $\lambda_i \neq \lambda_j$ *for* $i \neq j$, *with integers* n_i *and* $\deg B(s) \leqslant \deg A(s)$. *The partial fraction expansion of* $A^{-1}(s)B(s)$ *is defined by*

$$A^{-1}(s)B(s) = a_0 + \sum_{i=1}^{N}\sum_{j=1}^{n_i}\frac{a_{ij}}{(s-\lambda_i)^j} \tag{10.31}$$

where

$$B(s) = a_0 A(s) + \sum_{i=1}^{N}\sum_{j=1}^{n_i} a_{ij}\Big(\prod_{k\neq i}^{N}(s-\lambda_k)^{n_k}\Big)(s-\lambda_i)^{n_i-j}, \tag{10.32}$$

with the coefficients of this expression given by

$$\begin{aligned}
a_0 &= \lim_{s\to\infty}\frac{B(s)}{A(s)}, \\
a_{in_i} &= \lim_{s\to\lambda_i}(s-\lambda_i)^{n_i}\frac{B(s)}{A(s)}, \qquad i = 1,\ldots,N, \\
a_{ij} &= \lim_{s\to\lambda_i}\left[\frac{B(s)}{A(s)} - \sum_{k=j+1}^{n_i}\frac{a_{ik}}{(s-\lambda_j)^k}\right], \quad i = 1,\ldots,N, \quad j = 1,\ldots,n_i-1.
\end{aligned}$$

The following theorems give analytic expressions for H_2-optimal controllers that can be directly applied to Q_r, Q_{lm}, Q_{dm}, and Q_y, which are derived using the same mathematical approach described in [175] (interested readers are referred to this reference for explicit proofs of these theorems).

Theorem 10.4 *Suppose that* $p(s)$ *is the stable process* (10.29). *Let a proper weight function* $r(s)$ *be factored into an all-pass part and a minimum-phase part*

$$r(s) = r_a(s)r_m(s) = r_a(s)\frac{r_n(s)}{r_d(s)}. \tag{10.33}$$

Then the optimal solution to $\inf_{Q_r} \|(pQ_r - 1)r\|_2$ *is*

$$Q_r = \frac{B(s)}{p_m(s)r_n(s)}, \tag{10.34}$$

where $B(s)$ *is calculated from* (10.32) *for*

$$\frac{A(s)}{B(s)} = \frac{r_n(s)}{p_a(s)r_d(s)}, \tag{10.35}$$

with λ_i *being the roots of* $r_d(s)$.

Theorem 10.5 *Suppose that* $p(s)$ *is the stable process* (10.29). *Then the optimal solution to* $\inf_{Q_{lm}} \|p(1 + Q_{lm})l\|_2$ *is*

$$Q_{lm} = -1. \tag{10.36}$$

Theorem 10.6 *Suppose that* $p(s)$ *is the stable process* (10.29). *Let the minimum-phase part of a proper weight function* $d(s)$ *be written as* $d_n(s)/d_d(s)$. *Then the optimal solution to* $\inf_{Q_{dm}} \|(1 + pQ_{dm})d\|_2$ *is*

$$Q_{dm} = \frac{-B(s)}{p_m(s)d_n(s)}, \tag{10.37}$$

where $B(s)$ *is calculated from* (10.32) *for*

$$\frac{A(s)}{B(s)} = \frac{d_n(s)}{p_a(s)d_d(s)}, \tag{10.38}$$

with λ_i *being the roots of* $d_d(s)$.

Theorem 10.7 *Suppose that* $p(s)$ *is the stable process* (10.29). *Let the minimum-phase part of a proper weight function* $v(s) = p(s)l_u(s)+d_u(s)$ *be written as* $v_n(s)/v_d(s)$. *Then the optimal solution to* $\inf_{Q_y} \|(1 - pQ_y)v\|_2$ *is*

$$Q_y = \frac{B(s)}{p_m(s)v_n(s)}, \tag{10.39}$$

where $B(s)$ is calculated from (10.32) *for*

$$\frac{A(s)}{B(s)} = \frac{v_n(s)}{p_a(s)v_d(s)}, \tag{10.40}$$

with λ_i being the roots of $v_d(s)$.

The above analytical expressions obtained for the optimal controllers are stable but may be improper and, as a result, the controllers may be physically unrealizable. In the IMC design method [175], the optimal controllers are augmented with a low-pass filter such as

$$J_f(s) = \frac{1}{(\lambda_f s + 1)^{n_f}}, \tag{10.41}$$

with n_f just large enough such that the controllers $\mathbf{Q}$ are proper, at the expense of suboptimality[3]. In (10.41), λ_f is an adjustable parameter, with small values leading to a very fast response and large values resulting in manipulated variable moves that are slower and have smaller peak values during sharp changes in the inputs.

10.6 Example: Continuous Thin-film Dryer

Consider the thin-film composition control problem in a continuous dryer used for manufacturing of pharmaceutical thin-film tablets [165]. In this process, the drug formulation solution is cast as thin films that are dried to remove solvents (volatile components) of the solution through evaporation. Among the critical quality attributes of thin films are the solvent concentration remaining in the film, which heavily affects the mechanical characteristics and adhesion properties of the dried films. Hence, controlling the solvent concentration of the dried films is crucial to the overall process of thin-film tablet formation.

In the thin-film dryer investigated here, the manipulated variables used to control the solvent concentration in the film are flow rate of the formulation solution pumped into the dryer and temperature of hot air exposed to the film. The on-line mea-

[3]Somewhat more complicated filters are more appropriate in the presence of load disturbances when the closed-loop dynamics are much faster than the open-loop process dynamics [107].

surements available for control consist of film temperature and solvent concentration in the film. Next, we study several single-loop and multi-loop control systems for regulating the solvent concentration in the dried thin films.

Single-loop Control

A single-loop multi-objective control system is designed to regulate the solvent concentration in the film by manipulating the temperature of hot air blown into the dryer. The thin-film drying dynamics are described by the first-order-plus-dead-time (FOPDT) model

$$p_1(s) = \frac{-0.0003e^{-5s}}{50s+1}. \tag{10.42}$$

The process is affected by measured and unmeasured disturbances such that

$$y(s) = p_1(s)\big(u(s) + p_{lm}(s)l_m(s)\big) + p_{dm}(s)d_m(s) + d_u(s) \tag{10.43}$$

where $p_{lm} = \frac{5s+10}{50s^2+10s+1}$ and $p_{dm} = \frac{s+0.01}{s+1}$ are the measured load and output disturbance transfer functions, respectively. The output measurements are corrupted by stochastic sensor noise having a zero mean Gaussian distribution with $\sigma^2 = 10^{-4}$.

The control objective is to track a desired solvent concentration setpoint r while the process is perturbed by measured load disturbance l_m, measured output disturbance d_m, and unmeasured output disturbance d_u. The control structure in Section 10.2 was used to cast the control problem as a multi-objective controller design problem. Four independent control objectives were formulated to realize adequate reference tracking while rejecting the measured and unmeasured disturbances (see expressions (10.9)–(10.16)). The optimal IMC controllers Q_r, Q_{lm}, Q_{dm}, and Q_y were designed using Theorems 10.4 to 10.7 for a step change in r, l_m, d_m, and d_u, respectively. The IMC controllers were made proper so as to be physically realizable by augmenting with the first-order low-pass filter (10.41) with $\lambda_f = 2.0$. The time-delay term in (10.42) was approximated by a first-order Padé approximation [190].

The controllers are listed in Table 10.1. Figure 10-6 shows the profile of the

measured solvent concentration remaining in the film at the exit of the dryer when a step change is applied to the inputs r, l_m, d_m, and d_u of the closed-loop system. The solvent concentration should be maintained at 0.5 wt.% to achieve the desired extent of drying. Figure 10-6 indicates that Q_r enables very good reference tracking and the controllers Q_{lm}, Q_{dm}, and Q_y adequately reject the measured and unmeasured disturbances. The suppression of the measured load disturbance l_m is perfect. The closed-loop response for the measured and unmeasured output disturbance are limited by the same nonminimum phase behavior of the process. The closed-loop speed of response for the measured output disturbance and reference tracking is the same, as the two inputs act through the same controller transfer function Q_y.

Table 10.1: Multi-objective controllers for the single-loop control system.

$Q_r = -\frac{50s+1}{0.0006s+0.0003}$	$Q_{dm} = \frac{-16.215s^2+0.1757s+0.01}{0.0006s^2+0.000306s+0.000003}$
$Q_{lm} = -1$	$Q_y = -\frac{50s+1}{0.0006s+0.0003}$

Next, we investigate the performance of the single-loop multi-objective control system in response to failures in the sensors of the measurable variables. It is assumed that the sensor failures can be detected using fault detection and diagnosis methods (e.g., as described in [47, 79], and references therein). When a sensor fails, its measurements can no longer be used for control and, therefore, the respective controller is switched off. Figure 10-7 indicates the system responses in the event of sensor failures for the measurable variables r, l_m, d_m, and y. The system output (solvent concentration) measurements suggest that the closed-loop responses with respect to the measurable variables with working sensors are unaffected by removal of the failed sensors, as suggested by the analysis in Section 10.3.

Figure 10-7a shows the output response for a loss in the reference signal, which could occur due to loss in a communication line between an upper level supervisory control loop and a lower level regulatory control system. The comparison between Figures 10-6 and 10-7a reveals that the output responses to the disturbances l_m, d_m, and d_u are completely unaffected by the loss of reference signal, as they are just

shifted to a different baseline. On the other hand, Figure 10-7b suggests that the output responses to the inputs r, l_m, and d_m are completely unaffected by a failure in the output sensor y. This results from the fact that the feedforward controllers Q_r, Q_{lm}, and Q_{dm} remain intact by the loss in the feedback of y. Under these conditions, only the output response to the unmeasured disturbance is influenced by the loss of y, as the measurement of the output is the only way by which the control system can detect the presence of d_u.

Figure 10-7c indicates that losing the controller Q_{lm} affects the output response at $t = 150$ s while having no effects on the output responses to the reference r and unmeasured output disturbance d_u. The integrating action of Q_y forces the output response to follow the reference signal r after the measured load disturbance perturbs the system. Figure 10-7c also suggests that the loss of measured output disturbance d_m and, consequently, the controller Q_{dm} does not influence the output responses of the working sensors Q_r and Q_y. This numerical example clearly demonstrates the optimal failure tolerance of the proposed control structure and design method. It illustrates that the control structure alleviates the need to redesign controllers for optimal failure-tolerant control.

Cascade Control

The thin-film temperature is used as secondary process output to design a multi-objective cascade control system. The dynamic effect of hot air temperature (manipulated variable) on the thin-film temperature is described by the FOPDT model

$$p_2(s) = \frac{0.8}{2s+1}. \tag{10.44}$$

Equation (10.44) implies that manipulation of the hot air temperature has a much faster influence on the film temperature dynamics than on the solvent concentration dynamics (see (10.42)). This suggests that the film temperature, which is closely related to the solvent concentration in the film (the primary process output) and is readily available from on-line measurements, can be used to improve the dynamic

response of the closed-loop system in terms of disturbance rejection. The process is under the influence of various disturbances such that the secondary process output (film temperature) and the primary process output (solvent concentration) are

$$y_2(s) = p_2(s)\big(u(s) + l_{m2}(s)\big) + d_{u2}(s) \tag{10.45}$$

and

$$y_1(s) = p_1(s)\big(y_2(s) + l_{m1}(s)\big) + p_{dm1}(s)d_{m1}(s) + d_{u1}(s), \tag{10.46}$$

respectively, where $p_{dm1} = \frac{s+0.01}{s+1}$ (see Figure 10-3b for depiction of the closed-loop system). Both process outputs are corrupted by stochastic sensor noise defined as shown previously.

The control objective is to maintain the solvent concentration at a predetermined setpoint r while the different measured and unmeasured disturbances (l_{m1}, l_{m2}, d_{u1}, d_{u2}, and d_{m1}) affect the process. The general cascade control structure in Section 10.4 was applied to formulate the multi-objective controllers design problem for series and parallel cascade control systems. The optimal IMC controllers were designed using Theorems 10.4 to 10.7 for a step change in the input and, subsequently, augmented with the first-order low-pass filter (10.41) with $\lambda_f = 2.0$ to be made physically realizable. The controllers for the series and parallel cascade control systems are given in Table 10.2 and Table 10.3, respectively.

Table 10.2: Multi-objective controllers for the series cascade control system.

$Q_r = -\frac{100s^2+52s+1}{0.000953s^2+0.000953s+0.000238}$	$Q_{lm1} = -\frac{2.5s+1.25}{2s+1}$	$Q_{dm1} = \frac{42.42s^3+21.06s^2-0.1s-0.01}{4s^3+4.04s^2+1.04s+0.01}$
$Q_{y1} = -\frac{100s^2+52s+1}{0.000953s^2+0.000953s+0.000238}$	$Q_{lm2} = -1$	$Q_{y2} = \frac{2.5s+1.25}{2s+1}$

Figure 10-8a shows the closed-loop response of the primary process output for both cascade control systems. The optimal design of Q_r results in perfect reference tracking while the feedforward controllers Q_{lm1}, Q_{lm2}, and Q_{dm1} along with the feedback controllers Q_{y1} and Q_{y2} adequately reject all the disturbances affecting the

Table 10.3: Multi-objective controllers for the parallel cascade control system.

$Q_r = -\frac{50s+1}{0.000606s+0.000303}$	$Q_{lm1} = -\frac{2.5s+1.25}{2s+1}$	$Q_{dm1} = \frac{-16.215s^2+0.1757s+0.01}{0.0006s^2+0.000306s+0.000003}$
$Q_{y1} = -\frac{50s+1}{0.000606s+0.000303}$	$Q_{lm2} = -1$	$Q_{y2} = \frac{2.5s+1.25}{2s+1}$

system. For the thin-film process investigated here, Figure 10-8a indicates that the series and parallel cascade control systems exhibit comparable performance under the nominal process operation. Note that the proposed control structure provides a consistent framework to evaluate the performance of the cascade control systems as the performance comparison is independent of the choice of controllers tuning.

Figure 10-8b depicts the closed-loop system response for the two cascade control systems when the sensors used to measure the secondary load disturbance (l_{m2}) and the secondary process output (y_2) failed. Failure of the latter sensors rendered the secondary control loop in the cascade structures dysfunctional, as Q_{lm2} and Q_{y2} were switched off. Figure 10-8b suggests that the series cascade control system outperforms the parallel cascade system in the event of sensor failures. This is because of the longer response time of the parallel control system to restore the performance (bring the solvent concentration to its setpoint) after the load disturbance l_{m2} occurred at 350 s. Yet, the performance of the rest of the optimal controllers in both control systems remains intact due to optimal failure tolerance of the control structure.

Coordinated Control

The solvent concentration in the film can be controlled by manipulating the hot air temperature and feed flow rate in a coordinated manner. The effect of feed flow rate manipulation on the solvent concentration is described by

$$p_3(s) = \frac{0.001e^{-5s}}{6s+1}. \tag{10.47}$$

It is evident from the comparison between (10.42) and (10.47) that manipulation of the feed flow rate exhibits much faster dynamics than that of the hot air temperature.

Hence, manipulating the feed flow rate enables obtaining a better closed-loop response in terms of setpoint tracking and disturbance rejection. However, the feed flow rate should be reset to a predetermined setpoint during process operation to achieve a desired production rate.

Consider the process to be affected by various disturbances such that the process output (solvent concentration) is defined by

$$y(s) = p_1(s)\big(u_1(s) + l_{m1}(s)\big) + p_2(s)\big(u_2(s) + l_{m2}(s)\big) + p_{dm}(s)d_m(s) + d_u(s) \quad (10.48)$$

where $u_1(s)$ and $u_2(s)$ are the fast input (feed flow rate) and slow input (hot air temperature), respectively. The closed-loop control system is depicted in Figure 10-4b. The control objective is not only to regulate the solvent concentration in the presence of disturbances (l_{m1}, l_{m2}, d_m, and d_u), but also to reset the feed flow rate to its desired setpoint u_r as the slow input begins to influence the output. The general coordinated control structure in Section 10.4 was used to design the multi-objective control system. The analytic expressions for the optimal IMC controllers were derived for a step input. The tuning parameter of the first-order low-pass filter (10.41) was set to $\lambda_f = 2.0$. The controllers are listed in Table 10.4.

Table 10.4: Multi-objective controllers for the coordinated control system.

$Q_r = \frac{300s^2+56s+1}{0.0964s^2+0.0468s+0.0007}$	$Q_{lm1} = \frac{-0.05s-0.001}{0.0482s+0.0007}$	$Q_{dm} = \frac{300s^2+56s+1}{0.0964s^2+0.0496s+0.0007}$
$Q_y = \frac{300s^2+56s+1}{0.0964s^2+0.0468s+0.0007}$	$Q_{lm2} = \frac{0.0018s+0.0003}{0.0482s+0.0007}$	$Q_{ur1} = \frac{166.5s+3.33}{6s+1}$

Figure 10-9 shows the closed-loop response of the system for the nominal process operation and the case of system failures due to loss of the measured load disturbance signal l_{m1} and the input setpoint signal u_r. Figure 10-9a suggests that the loss of l_{m1} at 100 seconds is merely detrimental to the ability of the control system to reject the load disturbance affecting the fast control loop, as the rest of the controllers fulfill their objectives adequately. It is shown in Figure 10-9b that losing the input setpoint at 300 seconds (e.g., due to a communication failure between supervisory and

regulatory control levels) only makes the control system unable to maintain the feed flow rate at its desired level 1550 cm^3/s. The simulation results indicate that, in the event of sensor failures, the optimal performance of the control system is preserved with respect to the working controllers.

10.7 Conclusions

A general control structure is presented for the design of multiple degrees-of-freedom controllers for stable linear time-invariant multi-input multi-output processes. Through a Youla parameterization of all stabilizing controllers, it is demonstrated that the control structure is non-restrictive in terms of the achievable performance. The proposed control structure is an extension of the internal model control structure to systems with four degrees of freedom. The distinct feature of the control structure is that the multi-objective controllers can be designed independently of each other, as the control objectives are defined separately for each exogenous input. This is particularly significant for failure-tolerant control since the global optimality of a multi-objective control system remains intact when a controller is switched off due to actuator and/or sensor failures.

The control structure is applied for the design of multi-objective controllers for multi-loop control systems (cascade and coordinated control structures). Analytic expressions are obtained for optimal controllers design for stable single-input single-output processes with general inputs. Simulation of several single-loop and multi-loop control systems for a thin-film dryer indicates that the proposed approach enables designing optimal failure-tolerant controllers without compromising the best achievable performance.

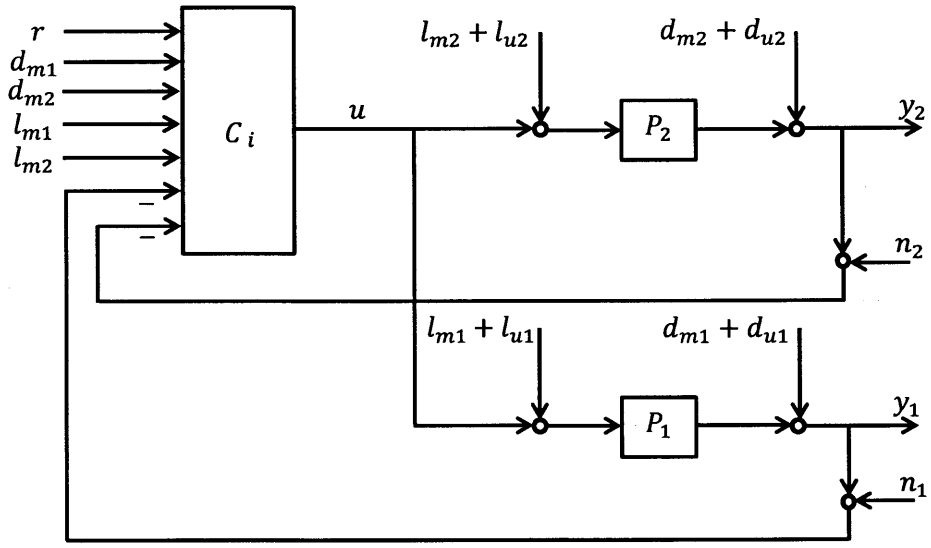
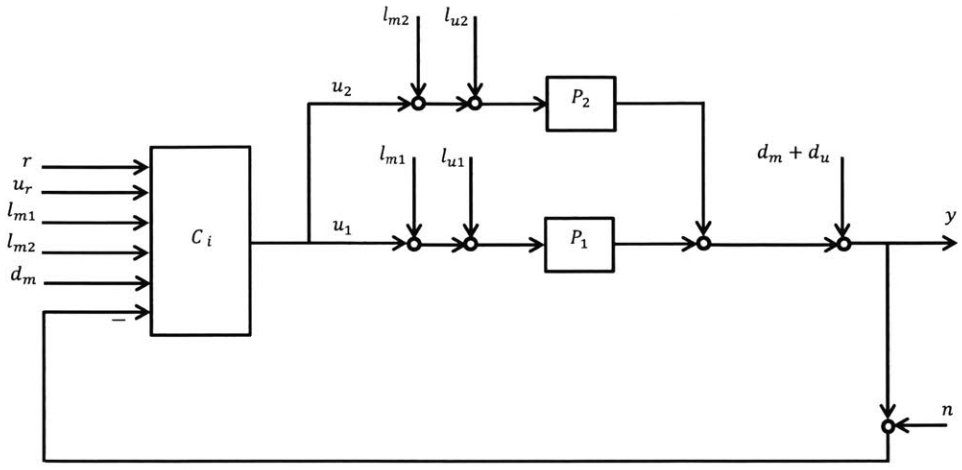
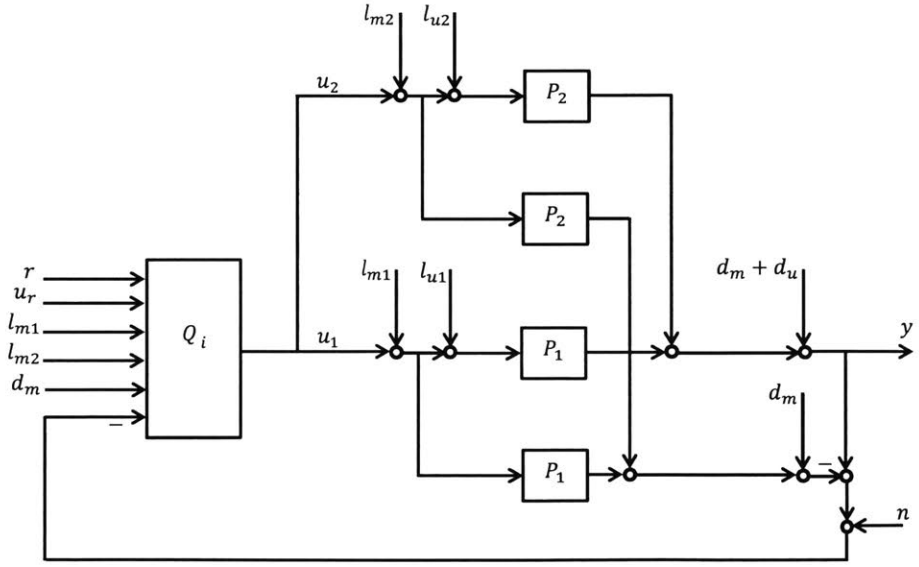


Figure 10-4: Classical parallel cascade structure.



(a) Classical structure



(b) General structure

Figure 10-5: Coordinated control system.

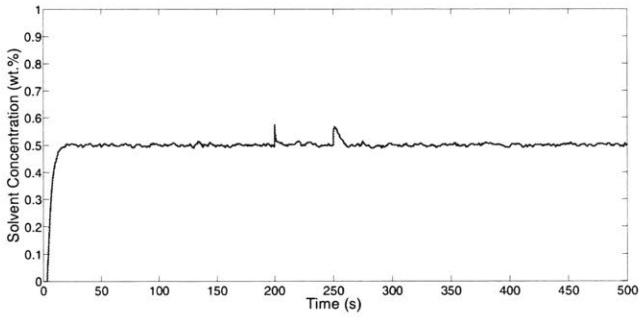
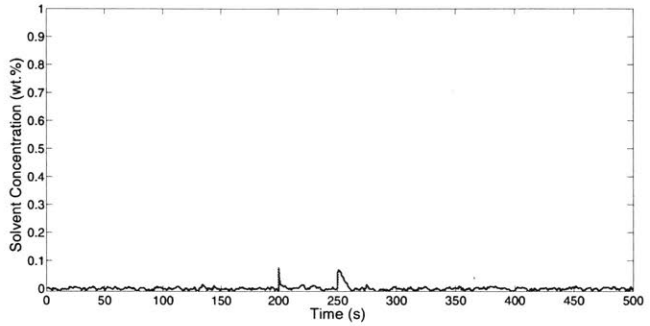
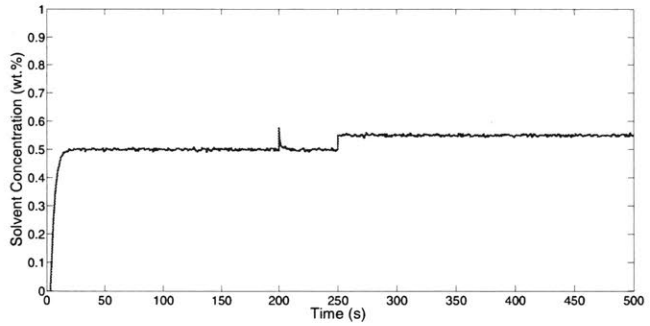


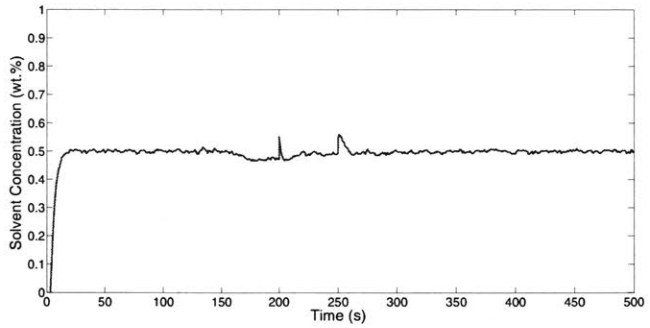
Figure 10-6: Dynamic behavior of the single-loop control system for a step change in the reference r, the measured load disturbance l_m, the measured output disturbance d_m, and the unmeasured output disturbance d_u at $t = 0$, 150, 200, and 250 s, respectively.



(a) Loss of the reference signal r; with Q_r taken out of service

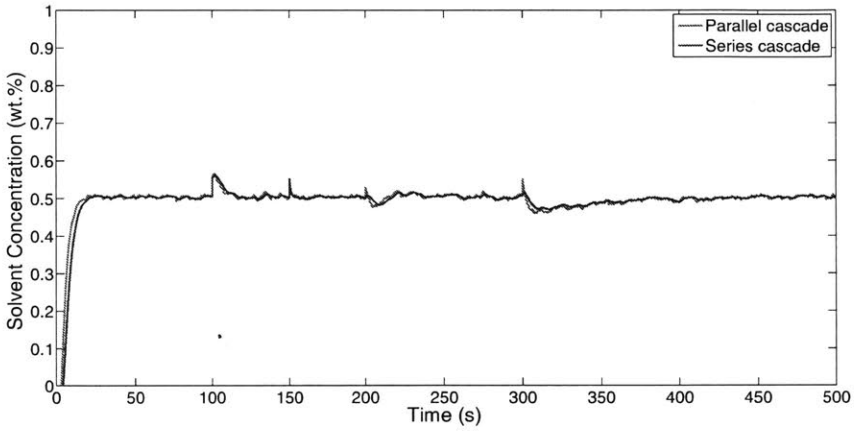


(b) Sensor failure for y; with Q_y taken out of service

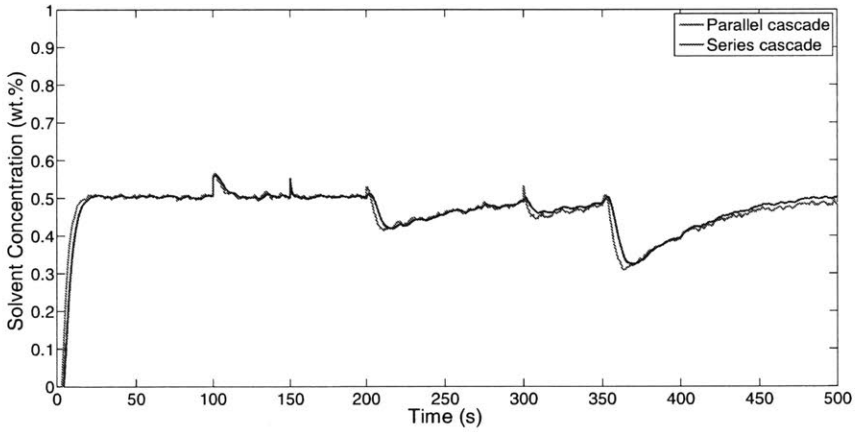


(c) Sensor failures for l_m and d_m; with Q_{lm} and Q_{dm} taken out of service

Figure 10-7: Dynamic behavior of the single-loop control system during various failures in the sensors of the measurable variables for a step change in the reference r, the measured load disturbance l_m, the measured output disturbance d_m, and the unmeasured output disturbance d_u at $t = 0$, 150, 200, and 250 s, respectively.

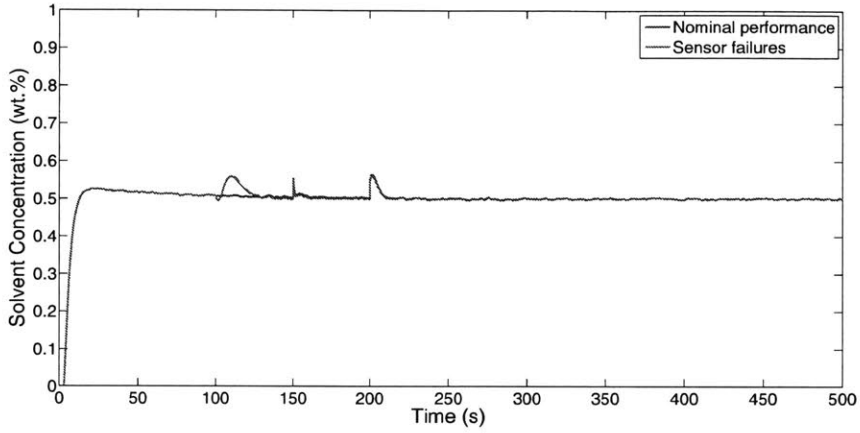


(a) Nominal performance

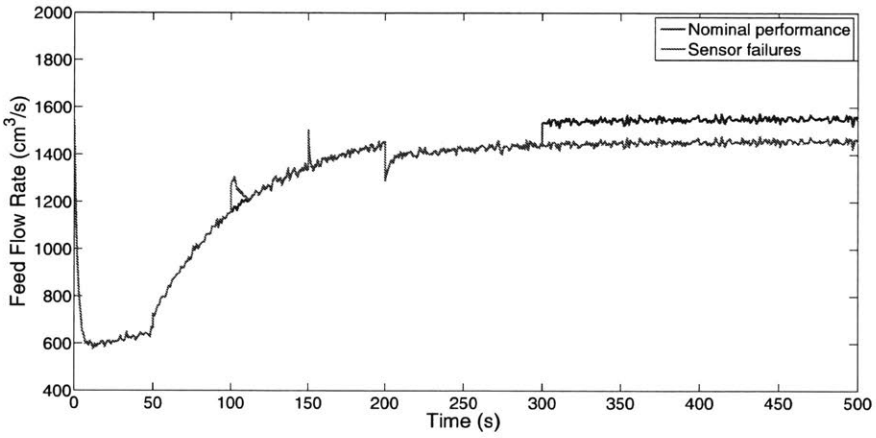


(b) Sensor failures for l_{m2} and y_2; with Q_{lm2} and Q_{y2} taken out of service

Figure 10-8: Dynamic behavior of the series and parallel control systems for a step change in the reference r, the unmeasured output disturbance in the primary loop d_{u1}, the measured output disturbance in the primary loop d_{m1}, the unmeasured output disturbance in the secondary loop d_{u2}, the measured load disturbance in the primary loop l_{m1}, and the measured load disturbance in the secondary loop l_{m2} at $t = 0$, 100, 150, 200, 300, and 350 s, respectively.



(a) System output (solvent concentration)



(b) Fast system input (feed flow rate)

Figure 10-9: Dynamic behavior of the coordinated control system for a step change in the reference r, the measured load disturbance for the slow input l_{m2}, the measured load disturbance for the fast input l_{m1}, the measured output disturbance d_m, the unmeasured output disturbance d_u, and the input setpoint u_r at $t = 0$, 50, 100, 150, 200, and 300 s, respectively.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 11

Active Fault Diagnosis for Nonlinear Systems

11.1 Introduction

Fault detection and isolation (FDI) has become increasingly important in maintaining stable, reliable, and profitable operations in the presence of component malfunctions, drifting parameters, and other abnormal events. Process disturbances, measurement noise, model nonlinearities, and other sources of uncertainty make fault diagnosis a challenging task that is further complicated by the steadily increasing complexity of industrial systems. By now, many methods have been proposed to address these challenges, such as residual- and observer-based methods [47, 197, 59, 30], set-based approaches [187, 221], and data-based methods [47]. The majority of these methods are *passive*, meaning that the inputs are not actively changed and that the fault status of the system is deduced only on the basis of measurements obtained during standard operation, compared with model predictions or historical data. However, faults may not be detectable or isolable at the current operating conditions, or when faults are obscured by the corrective action of the control system itself. Then it is necessary to inject a signal into the system to improve fault detectability and isolability, which is an approach known as *active fault diagnosis* [279]. Although active fault diagnosis can significantly improve fault isolation, the required excitations can have adverse

effects on the process that must be minimized.

The active input design problem has been addressed using deterministic [189, 188, 40, 227, 62, 3], stochastic [168, 230, 279, 207], and hybrid stochastic-deterministic [228] approaches. With few exceptions, the work is restricted to linear systems whereas, in reality, almost all systems exhibit nonlinear dynamics.

This chapter presents a deterministic model-based approach for active fault diagnosis of nonlinear systems with polynomial or rational dynamics subject to unknown-but-bounded uncertainties and disturbances. This work builds on a framework for set-based analysis of nonlinear systems (e.g., [237]) and allows for a deterministic formulation of the fault diagnosis problems. To decide whether a fault has occurred with certainty, the measured outputs are compared with the set of outputs from the nominal and faulty models [236].

The main contribution in this work is a method to determine optimal inputs for guaranteed active fault diagnosis despite process nonlinearities and uncertainties. To achieve this, the input design problem is formulated as a bilevel optimization problem in which the outer program minimizes the two-norm input, while the inner program guarantees that only inputs that separate the output sets are selected (i.e., inputs that guarantee the output measurements are consistent with at most one model). In order to solve this nonconvex bilevel problem, we propose a method that takes advantage of the fact that the relaxed inner program is convex at a fixed input (e.g., [236]). As suboptimality of this input is directly related to the tightness of the convex relaxation (i.e., how close the relaxation approximates the true set), different ways to tighten the relaxation are presented and discussed. Moreover, to provide a measure of output set separation, the inner convex program is reformulated in terms of an optimization-based consistency measure (e.g., [237]).

Organization The considered system class, the active input design problem, and the feasibility-based fault detection and isolation approach are presented in Section 11.2. Section 11.3 introduces the convex relaxation techniques used to determine outer approximations of the separating inputs. Section 11.4 presents the bilevel

optimization approach to determine the optimal separating input. The active input design method is illustrated for a two-tank system in Section 11.5 and the conclusions drawn in Section 11.6.

The majority of this chapter was published in the *Proceedings of the European Control Conference* [201].

11.2 Problem Formulation

Given a process subject to n_f possible faults, consider discrete-time models of the form

$$f^{[i]} : \begin{cases} x_{k+1} = g(x_k, u_k, w_k, p) \\ y_k = h(x_k, u_k, v_k, p) \end{cases} \tag{11.1}$$

representing the nominal and all the possible faulty dynamics. The superscript $i \in \mathcal{J} := \{0, 1, \ldots, n_f\}$, denotes the various fault scenarios $\mathcal{F} := \{f^{[0]}, f^{[1]} \ldots, f^{[n_f]}\}$ considered, with $f^{[0]}$ corresponding to the nominal model.

In (11.1), $x_k \in \mathbb{R}^{n_x}$, $y_k \in \mathbb{R}^{n_y}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_w}$, $v_k \in \mathbb{R}^{n_v}$ denote the system states, outputs, inputs, process noise, and measurement noise at time-point k, respectively. The possibly uncertain model parameters are denoted by $p \in \mathbb{R}^{n_p}$. The functions g and h are assumed to be polynomial or rational. Many other nonlinearities (e.g., exponential, transcendental, quasi-polynomial) can be approximated or exactly expressed by polynomial or rational functions (e.g., see references in [237]). In general, each fault model $f^{[i]}$ has its own set of variables $x_k^{[i]}, w_k^{[i]}, v_k^{[i]}, y_k^{[i]}, p^{[i]}$ and functions $g^{[i]}$ and $h^{[i]}$ with possibly different dimensions. However, to shorten and simplify the notation, the superscript $[i]$ is omitted for all variables and functions appearing in (11.1).

The model parameters are assumed to be unknown-but-bounded. Furthermore, the states as well as the process and measurement noise are assumed bounded, leading to the overall uncertainty description:

$$p \in \mathcal{P},\ x_k \in \mathcal{X}_k,\ w_k \in \mathcal{W}_k,\ v_k \in \mathcal{V}_k,\ \ \forall k \in \mathcal{T} \tag{11.2}$$

where $\mathcal{T} = \{0, 1 \ldots, n_t\}$ is the collection of time instances considered. To simplify the presentation, only one model is assumed to be active during $\mathcal{T}$. The framework can easily be extended to include fault sequences.

Set-based Fault Diagnosis

This work employs the notion of *consistency* for the design of separating inputs. The core idea is to check consistency by means of a feasibility problem that takes into account the model and the uncertainty description. The feasibility problem is derived next.

Denote sequences on $\mathcal{T}$ by $\widetilde{\sigma} := [\sigma_0, \sigma_1, \ldots, \sigma_{n_t}]^\top$. Then, given a sequence of inputs and measured outputs, $(\widetilde{u}, \widetilde{y})$, combining (11.1) with the uncertainty description in (11.2) into a single feasibility problem (FP) for each model $i \in \mathcal{J}$ gives

$$\mathrm{FP}^{[i]}(\widetilde{u}) : \begin{cases} \text{find } \xi_{\mathrm{FP}} & \\ \text{s.t. } x_{k+1}{=}g(x_k, u_k, w_k, p), & \forall k \in \mathcal{T}\backslash n_t \\ \quad y_k{=}h(x_k, u_k, v_k, p), & \forall k \in \mathcal{T} \\ \quad x_k \in \mathcal{X}_k, w_k \in \mathcal{W}_k, v_k \in \mathcal{V}_k\,, & \forall k \in \mathcal{T} \\ \quad p \in \mathcal{P} & \end{cases}$$

where $\xi_{\mathrm{FP}} = [x_0, \ldots, x_{n_t}, w_0, \ldots, w_{n_t}, y_0, \ldots, y_{n_t}, v_0, \ldots, v_{n_t}, p]^\top$ lumps all the variables except the input into a single vector. The superscript $[i]$ on the variables in $\mathrm{FP}^{[i]}(\widetilde{u})$ has been omitted to simplify notation.

Definition 11.1 (Consistency) *Input and output sequences $(\widetilde{u}, \widetilde{y})$ are said to be consistent with a fault candidate $f^{[i]}$ if the problem $FP^{[i]}(\widetilde{u})$ admits a solution (meaning that there exists a set of states, process noise, measurement noise, and parameters such that the input sequence $\widetilde{u}$ leads to that particular output sequence $\widetilde{y}$). Otherwise, the sequence $(\widetilde{u}, \widetilde{y})$ is called inconsistent.*

Optimal Input Design for Active Fault Diagnosis

Passive set-based fault detection and isolation based on $\text{FP}^{[i]}(\widetilde{u})$ has been proposed in [221] considering fixed or unknown-but-bounded inputs. The main focus of this work is the design of separating inputs for active fault detection and isolation in the presence of uncertainties, which is formally stated below.

Definition 11.2 (Robust Separating Input) *An input $\widetilde{u}$ separates models $f^{[i]}$ and $f^{[j]}$ at time t_{n_t}, if $\nexists \widetilde{y}$ such that $(\widetilde{u}, \widetilde{y})$ is consistent with both $f^{[i]}$ and $f^{[j]}$. If $\widetilde{u}$ separates all model output sets subject to output, process, and parametric uncertainties, then $\widetilde{u}$ is a robust separating input.*

The following combined feasibility problem (e.g., [236]) is defined as

$$\text{FP}^{[i,j]}(\widetilde{u}) : \begin{cases} \text{find} & \xi_{\text{FP}}^{[i]}, \xi_{\text{FP}}^{[j]} \\ \text{s.t.} & \text{constraints in } \text{FP}^{[i]}(\widetilde{u}) \\ & \text{constraints in } \text{FP}^{[j]}(\widetilde{u}) \\ & y_{n_t}^{[i]} = y_{n_t}^{[j]} \end{cases}$$

which checks if there exists $\left(\xi_{\text{FP}}^{[i]}, \xi_{\text{FP}}^{[j]}\right)$ such that the outputs of models $f^{[i]}, f^{[j]} \in \mathcal{F}$ intersect at time step n_t. Both models have their own distinct set of states, process and measurement noise, parameters, and outputs that appear as free variables in the program. This problem provides a *robustness certificate* when the problem is shown to be infeasible for all possible model combinations [236]. The set of all robust separating inputs is defined as $\mathcal{U}^* := \{\widetilde{u} : \text{FP}^{[i,j]}(\widetilde{u}) \text{ is infeasible } \forall (i,j) \in \mathcal{J},\, i > j\}$.

Note that $\text{FP}^{[i,j]}(\widetilde{u})$ is defined using only the output at the final time meaning any $\widetilde{u} \in \mathcal{U}^*$ applied to the system guarantees that the output sets at the final time do not intersect. This can be replaced with requirements in which the output sets either do not intersect either at all, or at least at one time instance of $\mathcal{T}$. The first requirement can be straightforwardly obtained by replacing the condition $y_{n_t}^{[i]} = y_{n_t}^{[j]}$ with $y_k^{[i]} = y_k^{[j]},\ \forall k \in \mathcal{T}$, and the second requirement is considered in [237].

Note that for unique fault diagnosis, every possible pairwise comparison of the

models of interest must have separated output sets [236], which requires a check of $\binom{n_f+1}{2}$ distinct combinations. This number can be reduced by applying a hierarchical approach that first selects inputs that separate a number of the models and then refines the input to improve separability for closely related models. An alternative is a hierarchical approach that first selects inputs that separate a number of the models and then refines the input in a second step to improve separability for closely related models. The hierarchical approach can significantly improve separability, since one input sequence is not required to separate all models, but only a smaller number of the models, which can be used to limit the performance loss resulting from the input used for separation.

In this work, we not only want to separate the models, we also want to derive an input that is optimal with respect to a certain performance index. Below it is assumed that the performance loss can be classified in terms of a quadratic penalty of the separating input $\widetilde{u}$. The overall problem considered in this chapter is stated as:

Problem 11.1 (Optimal Separating Input) *Find a separating input $\widetilde{u}$ that solves*

$$\begin{aligned} \inf \quad & \widetilde{u}^{\mathsf{T}} R \widetilde{u} \\ \text{s.t.} \quad & \widetilde{u} \in \mathcal{U} \cap \mathcal{U}^* \end{aligned} \tag{11.3}$$

where R is a positive-semidefinite weighting matrix and $\mathcal{U}$ is a convex set that represents the input constraints.

The solution $\widetilde{u}$, assuming that it exists, is guaranteed to provide robust fault diagnosis within n_t time steps. However, due to the nonlinearities and uncertainties, the combined feasibility problem $\mathrm{FP}^{[i,j]}(\widetilde{u})$ is generally nonconvex, making $\mathcal{U}^*$ very difficult to characterize exactly. Next these problems are efficiently tackled using convex relaxations and bilevel optimization.

11.3 Convex Relaxations

Providing a robustness certificate is not a trivial task for nonlinear and uncertain systems due to the general nonconvexity of $\mathrm{FP}^{[i,j]}(\widetilde{u})$. Although difficult, this is a necessary task for characterizing the set of robust separating inputs $\mathcal{U}^*$ that appears in the optimization (11.3).

For polynomial systems, we can convexly relax the single feasibility problems, $\mathrm{FP}^{[i]}(\widetilde{u})$, into a semidefinite or linear program that provides provable inconsistency certificates for the existence of solutions. These convex outer approximations of the original, generally nonconvex, feasible sets can be used in $\mathrm{FP}^{[i,j]}(\widetilde{u})$ so that solutions can efficiently be computed [236]. The drawback of such an approach is that the outer approximations introduce conservatism into the problem, so they only characterize a subset of $\mathcal{U}^*$. Nevertheless, robustness certificates derived from these convex relaxations allow the computation of an optimal and robust separating input where the tightness of the relaxation controls the degree of conservatism in the solution.

Relaxation Methodology

The first step is to transform the $\mathrm{FP}^{[i]}(\widetilde{u})$ into a quadratically constrained program (QCP) by expressing all the dynamic and output equations as $\xi^\top A \xi$, where $\xi \in \mathbb{R}^{n_\xi}$ is a minimal basis of monomials for the equations of model $f^{[i]}$ and $A \in \mathbb{R}^{n_\xi \times n_\xi}$ is a symmetric matrix. The vector ξ contains the elements of ξ_{FP}, the constant term 1, and any additional monomials greater than degree two necessary to represent all equations. Such a quadratic decomposition can always be found; however, it is in general not convex. By introducing a symmetric matrix $X = \xi\xi^\top$ and replacing the resulting $\mathrm{trace}(X) \geq 1$ and $\mathrm{rank}(X) = 1$ constraints with the weaker positive-semidefinite constraint $X \succeq 0$, the $\mathrm{FP}^{[i]}(\widetilde{u})$ is relaxed into a semidefinite program (SDP) denoted by $\mathrm{SDP}^{[i]}(\widetilde{u})$. It is important to note that this relaxation only increases the solution set. See [237] for further details and references regarding this relaxation method.

Using the index set $\mathcal{L} = \{1, \ldots, n_{\mathrm{eq}}\}$ where n_{eq} represents the total number of equality constraints in model $f^{[i]}$ and assuming that (11.2) can be written as n_{ineq}

linear inequality constraints in terms of the uncertain variables (i.e., $B\xi \leq 0$ where $B \in \mathbb{R}^{n_{\text{ineq}} \times n_\xi}$) results in

$$\text{SDP}^{[i]}(\widetilde{u}): \begin{cases} \text{find} & X \\ \text{s.t.} & \text{trace}\,(A_l(\widetilde{u})X) = 0, \quad \forall l \in \mathcal{L} \\ & \text{trace}(ee^\top X) = 1 \\ & B(\widetilde{u})Xe \leq 0 \\ & X \succeq 0 \end{cases}$$

where $e = [1, 0, \ldots, 0]^\top \in \mathbb{R}^{n_\xi}$. Note that the matrices $B(\widetilde{u})$ and $A(\widetilde{u})$ depend on the input $\widetilde{u}$; however, at this level, the input is treated as a constant (the input is minimized in an outer loop optimization explained below).

To deal with larger problems that involve many constraints and variables, the $\text{SDP}^{[i]}(\widetilde{u})$ can be relaxed to a linear program (LP) denoted by $\text{LP}^{[i]}(\widetilde{u})$, which is done by simply dropping the constraint $X \succeq 0$.

Solving $\text{SDP}^{[i]}(\widetilde{u})$ instead of the original $\text{FP}^{[i]}(\widetilde{u})$ usually leads to the inclusion of false solutions. As a result, the input set $\mathcal{U}^*$ for which separation is guaranteed gets overly restricted. To alleviate this effect, constraints can be added that are redundant in the ξ_{FP} basis, but are not necessarily redundant in the higher dimensional basis X. Such redundant constraints can be constructed by [237]:

$$\text{trace}(B(\widetilde{u})_i^\top B(\widetilde{u})_j X) \geq 0, \forall\, (i,j) \in \{1, \ldots, n_{\text{ineq}}\},\ i \geq j \tag{11.4}$$

where $B(\widetilde{u})_i \in \mathbb{R}^{1 \times n_\xi}$ represents the ith row of $B(\widetilde{u})$. Eq. (11.4) includes the McCormick relaxations for bilinear monomials. Including these redundant constraints in the $\text{SDP}^{[i]}(\widetilde{u})$ adds $n_{\text{ineq}}(n_{\text{ineq}} - 1)/2$ constraints that make the solution more demanding to compute but can significantly tighten the solution set [237]. The tradeoff between speed and conservatism is discussed below in the context of a numerical example.

Robustness Certificates Using Convex Relaxations

This section shows how to efficiently determine robustness certificates, i.e., guaranteed output set separation, using the convexly relaxed feasibility problems.

Define the problems $\text{SDP}^{[i,j]}(\widetilde{u})$ and $\text{LP}^{[i,j]}(\widetilde{u})$ similarly as $\text{FP}^{[i,j]}(\widetilde{u})$ (see Section 11.2). Since the convexly relaxed problems must contain the original output sets (i.e., provide infeasibility certificates for $\text{FP}^{[i]}(\widetilde{u})$), they can also be used to check if a given $\widetilde{u}$ is a robust separating input (see [236]). We can now similarly define the sets $\mathcal{U}^*_{\text{SDP}} := \{\widetilde{u} : \text{SDP}^{[i,j]}(\widetilde{u}) \text{ is infeasible}, \forall (i,j) \in \mathcal{J}, i > j\}$ and $\mathcal{U}^*_{\text{LP}} := \{\widetilde{u} : \text{LP}^{[i,j]}(\widetilde{u}) \text{ is infeasible}, \forall (i,j) \in \mathcal{J}, i > j\}$. Since the relaxed output sets must contain the actual set, but not vice versa, it directly follows that $\mathcal{U}^*_{\text{LP}} \subseteq \mathcal{U}^*_{\text{SDP}} \subseteq \mathcal{U}^*$. Note that $\mathcal{U}^*_{\text{LP}} \subseteq \mathcal{U}^*_{\text{SDP}}$ holds only when the same set of constraints are included in both relaxations.

11.4 Optimal Robust Separating Input

In this section, Problem 11.1 is tackled by employing robustness certificates and bilevel optimization. The inner program of the bilevel optimization certifies, for a given input, that the output sets do not overlap and are therefore consistent with at most one model, while the outer program determines the minimally harmful input. $\text{FP}^{[i,j]}(\widetilde{u})$ could be directly solved to determine if the sets overlap at a given input. However, the result from the feasibility test provides no suitable measure for the outer program to determine a direction that will improve the objective function value. This situation can be avoided by reformulating the $\text{FP}^{[i,j]}(\widetilde{u})$ in terms of an optimization-based consistency measure δ whose value provides a direct measure of output set separation for a fixed input. This measure provides useful information for the outer solver to guide the input toward a minimum, as discussed below.

Below, the mathematical definition of δ is provided along with an explanation of how this can interpreted as a measure of output set separation. Then, the bilevel optimization strategy for minimizing the distance between output sets is provided. The presented approach to determine output set separation does not require the

explicit computation of reachable sets as, for example, done in [222].

Measure of Output Set Separation

The determination of the output set separation requires a reformulation of $\mathrm{FP}^{[i,j]}(\widetilde{u})$ in terms of a scaling parameter δ. Depending on δ, the bounds will be either inflated ($\delta > 0$) or shrank ($\delta < 0$) until the output sets intersect. Intersection of the output sets is checked by solving $\mathrm{FP}^{[i,j]}(\widetilde{u})$ for a given δ and input $\widetilde{u}$. Since many δ values will satisfy this relationship, δ is minimized under the constraints that the output sets still overlap. δ then provides a measure of separation in the sense that the larger the distance between the output sets, the larger the minimum value of δ. If an inflation of the bounds, i.e. $\delta > 0$, was required for output set separation, then the provided input separates the output sets despite all bounded uncertainties.

The inflated bounds (i.e., inequality constraints) can be written as $B_{\mathrm{in}}\xi_{\mathrm{FP}} \leq b_{\mathrm{in}}(\widetilde{u}) + \delta$, where B_{in} is a sparse matrix that relates each of the variables in ξ_{FP} to their lower and upper bounds in the vector $b_{\mathrm{in}}(\widetilde{u})$, which might depend on the input. Using this definition, $\mathrm{FP}^{[i,j]}(\widetilde{u})$ can be reformulated in terms of the minimum inflation parameter δ:

$$
\begin{aligned}
\widehat{\delta}^{[i,j]}(\widetilde{u}) := \min_{\delta, \xi_{\mathrm{FP}}^{[i]}, \xi_{\mathrm{FP}}^{[j]}} \quad & \delta \\
\text{s.t.} \quad & \text{equality constraints in } \mathrm{FP}^{[i]}(\widetilde{u}) \\
& \text{equality constraints in } \mathrm{FP}^{[j]}(\widetilde{u}) \\
& B_{\mathrm{in}}^{[i]}\xi_{\mathrm{FP}}^{[i]} \leq b_{\mathrm{in}}^{[i]}(\widetilde{u}) + \delta \\
& B_{\mathrm{in}}^{[j]}\xi_{\mathrm{FP}}^{[j]} \leq b_{\mathrm{in}}^{[j]}(\widetilde{u}) + \delta \\
& y_{n_t}^{[i]} = y_{n_t}^{[j]}.
\end{aligned}
\tag{11.5}
$$

Theorem 11.1 $\mathrm{FP}^{[i,j]}(\widetilde{u})$ *is infeasible if and only if* $\widehat{\delta}^{[i,j]}(\widetilde{u}) > 0$.

Proof. Choose any $i, j \in \mathcal{J}$ and any $\widetilde{u} \in \mathcal{U}$. If $\widehat{\delta}^{[i,j]}(\widetilde{u}) > 0$, then (11.5) does not have a feasible point with $\delta \leq 0$. Therefore, $\nexists(\xi_{\mathrm{FP}}^{[i]}, \xi_{\mathrm{FP}}^{[j]})$ such that the constraints stated in $\mathrm{FP}^{[i,j]}(\widetilde{u})$ hold. Thus, $\mathrm{FP}^{[i,j]}(\widetilde{u})$ is infeasible. Conversely, assume that $\mathrm{FP}^{[i,j]}(\widetilde{u})$ is

infeasible. Then $\nexists(\xi_{\text{FP}}^{[i]}, \xi_{\text{FP}}^{[j]})$ such that the constraints in $\text{FP}^{[i,j]}(\widetilde{u})$ hold, which implies that (11.5) does not have a feasible solution with $\delta \leq 0$. Thus, $\widehat{\delta}^{[i,j]}(\widetilde{u}) > 0$. □

From Theorem 11.1, we can express the set of separating inputs as $\mathcal{U}^* = \{\widetilde{u} : \widehat{\delta}^{[i,j]}(\widetilde{u}) > 0,\ \forall (i,j) \in \mathcal{J},\ i > j\}$. It directly follows from Section 11.3 that:

$$\begin{aligned} \mathcal{U}_{\text{SDP}}^* &= \{\widetilde{u} : \widehat{\delta}_{\text{SDP}}^{[i,j]}(\widetilde{u}) > 0,\ \forall (i,j) \in \mathcal{J},\ i > j\} \\ \mathcal{U}_{\text{LP}}^* &= \{\widetilde{u} : \widehat{\delta}_{\text{LP}}^{[i,j]}(\widetilde{u}) > 0,\ \forall (i,j) \in \mathcal{J},\ i > j\} \end{aligned} \tag{11.6}$$

and $\widehat{\delta}_{\text{LP}}^{[i,j]}(\widetilde{u}) \leq \widehat{\delta}_{\text{SDP}}^{[i,j]}(\widetilde{u}) \leq \widehat{\delta}^{[i,j]}(\widetilde{u})$ at a fixed $\widetilde{u}$ where $\widehat{\delta}_{\text{SDP}}^{[i,j]}(\widetilde{u})$ and $\widehat{\delta}_{\text{LP}}^{[i,j]}(\widetilde{u})$ are defined in the same manner as (11.5) with the convexly relaxed constraints used in place of the actual constraints.

Determining the Optimal Input

The previous section derived a method for characterizing the set of separating inputs with the measure of output set separation $\widehat{\delta}^{[i,j]}(\widetilde{u})$. Using this framework, Problem 11.1 can be redefined as:

$$\begin{aligned} \inf_{\widetilde{u} \in \mathcal{U}} \quad & \widetilde{u}^\top R \widetilde{u} \\ \text{s.t.} \quad & \widehat{\delta}^{[i,j]}(\widetilde{u}) > 0,\ \forall (i,j) \in \mathcal{J},\ i > j. \end{aligned} \tag{11.7}$$

$\widehat{\delta}^{[i,j]}(\widetilde{u})$ is defined as the solution to a nonconvex optimization which clearly makes (11.7) a nonlinear nonconvex bilevel program (BLP). Only very few results for BLPs with nonconvex inner programs exist [172]. The computational complexity can be reduced by convexly relaxing the $\widehat{\delta}^{[i,j]}(\widetilde{u})$ constraint in (11.7), using the relaxations in Section 11.3 for example, to form a convex BLP that can be solved using existing algorithms [9]. However, we solve this problem differently as explained below.

The resulting convexly relaxed problem is

$$\begin{aligned} \min_{\widetilde{u}\in\mathcal{U}} \quad & \widetilde{u}^\top R\widetilde{u} \\ \text{s.t.} \quad & \widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u}) \geq \epsilon, \ \forall (i,j) \in \mathcal{J}, \ i > j \end{aligned} \tag{11.8}$$

where the subscript CR stands for convex relaxation (e.g., SDP or LP). In (11.8), a minimum separation threshold $\epsilon > 0$ is introduced to ensure that there exists a $\widetilde{u}^\star$ that attains the minimum. The program for $\widehat{\delta}_{\text{CR}}^{[i,j]}$ has the same structure as (11.5) with the constraints defined in terms of the particular relaxation. Note that $\widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u})$ (i.e., the inner program) is now convex for a fixed $\widetilde{u}$.

Replacing $\widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u})$ with its equivalent Karush-Kuhn-Tucker (KKT) conditions leads to a single program with, first, a large number of nonconvex complementary constraints equal to the number of inequalities in $\widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u})$ and, second, variable bounds that are generally complicated and highly nonlinear functions of the input due to its propagation through the system dynamics. As this procedure is used by many of the aforementioned algorithms to solve BLPs, computing the solution can be very computationally expensive, which is compounded by a combinatorial growth in the number of inner programs with the number of models to be separated and a polynomial growth in the size of each inner program with the number of time points.

Although standard algorithms can be used to solve (11.8), we propose a simple alternative that is easy to implement regardless of model complexity. The idea is to use a deterministic local nonlinear solver to supply input values to the convex inner program, $\widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u})$, that can be efficiently solved using a variety of solvers, such as CPLEX [1] or SeDuMi [240]. Because $\widehat{\delta}_{\text{CR}}^{[i,j]}(\widetilde{u})$ provides a measure of distance between the set of outputs, the outer solver can easily compute a gradient of this constraint with respect to the input using finite differencing. Combined with the gradient/subgradients of the outer objective function, a feasible descent direction can be estimated to update $\widetilde{u}$. This process can be repeated until optimality is achieved. This approach is used for the example of a two-tank system in the next section.

11.5 Example: Two-tank Benchmark Problem

The previously described active FDI algorithm is applied to a variant of the two-tank problem that is a common problem in the fault diagnosis literature. An illustration of the two-tank system is provided in Figure 11-1.

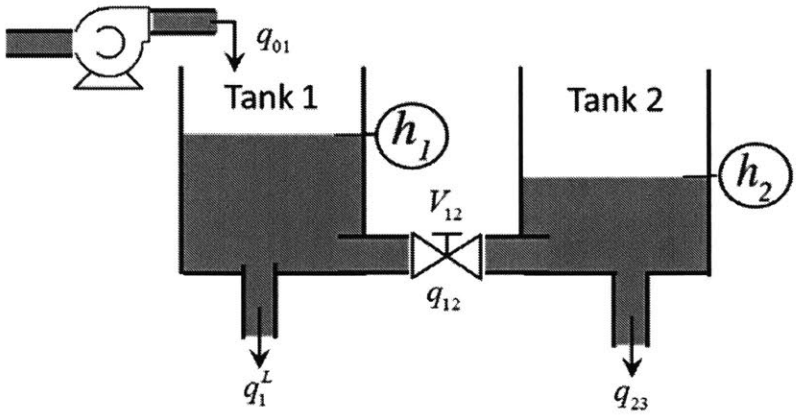


Figure 11-1: Sequential two-tank system.

System Description

Both tanks have the same cross-sectional area A and are connected in series by a valve V_{12} with inflow q_{01} and outflow q_{23}. The inflow to the first tank q_{01} represents the system input and can vary with time. The states (outputs) of the system are the true (measured) heights of the tanks, which are denoted by x_1 (y_1) and x_2 (y_2) for tanks 1 and 2, respectively. This example considers operating conditions in which the system satisfies $x_1 \geq x_2$ under all fault scenarios.

Three fault scenarios are considered. First ($f^{[1]}$), a leakage in tank 1 represented by the flow q_1^L. Second ($f^{[2]}$), the valve V_{12} becomes clogged and its throughput is reduced by 50%. Third ($f^{[3]}$), a large leak occurs in tank 2 that increases the outflow to 5 times its nominal value. These scenarios can all be represented with the same model structure. Therefore, we will first derive a general description of the system and then present the different sets of parameters in the nominal and faulty models.

Under the aforementioned assumptions, the nonlinear discrete-time model is given by the following set of difference equations

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + \tfrac{\Delta t}{A}\left(q_{01,k} - q^L_{1,k} - q_{12,k}\right) \\ x_{2,k+1} &= x_{2,k} + \tfrac{\Delta t}{A}\left(q_{12,k} - q_{23,k}\right) \end{aligned} \tag{11.9}$$

where Δt is the sampling time and the flowrates are

$$\begin{aligned} &q_{01,k} = u_k, \quad q^L_{1,k} = c^L_1\sqrt{x_{1,k}} \\ &q_{12,k} = c_{12}\sqrt{x_{1,k} - x_{2,k}} \\ &q_{23,k} = c_{23}\sqrt{x_{2,k}}. \end{aligned} \tag{11.10}$$

The parameters c^L_1, c_{12}, and c_{23} are valve coefficients. The measured heights are corrupted with measurement noise:

$$\begin{aligned} y_{1,k} &= x_{1,k} + v_{1,k} \\ y_{2,k} &= x_{2,k} + v_{2,k}. \end{aligned} \tag{11.11}$$

The non-polynomial parts of (11.9) can be reformulated by introducing additional variables and constraints [221]:

$$\begin{aligned} \left(dw_{12,k}\right)^2 &= x_{1,k} - x_{2,k} \\ \left(Sqh_{1,k}\right)^2 &= x_{1,k} \\ \left(Sqh_{2,k}\right)^2 &= x_{2,k}. \end{aligned} \tag{11.12}$$

Placing $dw_{12,k}$, $Sqh_{1,k}$, and $Sqh_{2,k}$ in (11.9) instead of the appropriate square root terms results in a polynomial model. Defining the parameter vector $p = [A, c^L_1, c_{12}, c_{23}]^\top$,

the fault models can be classified as

$$\begin{aligned} f^{[0]} &:= \left\{ (11.9)\text{–}(11.12),\ p^{[0]} = [A, 0, c_{12}, c_{23}]^\top \right\} \\ f^{[1]} &:= \left\{ (11.9)\text{–}(11.12),\ p^{[1]} = [A, c_1^L, c_{12}, c_{23}]^\top \right\} \\ f^{[2]} &:= \left\{ (11.9)\text{–}(11.12),\ p^{[2]} = [A, 0, 0.5c_{12}, c_{23}]^\top \right\} \\ f^{[3]} &:= \left\{ (11.9)\text{–}(11.12),\ p^{[3]} = [A, 0, c_{12}, 5c_{23}]^\top \right\}. \end{aligned} \tag{11.13}$$

Simulation Details

The bilevel optimization problem derived in this chapter (11.8) was solved to compute the optimal separating input $\widetilde{u}^*$ for the nominal and three fault models summarized in (11.13) using the method in Section 11.4. The convex relaxations in Section 11.3 were used. The implementation was done in Matlab using `fmincon` as the outer solver and CPLEX [1] (respectively, CVX [91] with SeDuMi [240]) as the inner solver for the linear (respectively, semidefinite) relaxations.

All simulations used the parameter values $\Delta t = 5$ s, $A = 1.54\text{e}{-2}$ m^2, $c_L^1 = c_{12} = c_{23} = 1.2\text{e}{-4}$ m$^{5/2}$/s, $R = I_{n_u \times n_u}$, and $n_t = 4$. The uncertain initial tank levels were chosen to be $x_{1,0} \in [0.95, 1.05]$ and $x_{2,0} \in [0.475, 0.525]$ with bounded measurement noise $v_{1,k}, v_{2,k} \in [-0.05, 0.05]$, $\forall k \in \mathcal{T}$ all in units of meters. Bounds on the remaining states and outputs were calculated using the model equations, the uncertain initial condition interval, and the bounded measurement noise using the Matlab Intlab toolbox [220]. The initial guess for $\widetilde{u}$ in the outer program was chosen to be 0.1 for all time points in all simulations. Different initial guess values resulted in the exact same optimal solution for this example. Note that our proposed method easily handles process noise and parametric uncertainty; however, they were excluded from this particular example for simplicity.

Main Results

Table 11.1 compares our method for different levels of convex relaxations. The linear (respectively, semidefinite) relaxation without (11.4) is denoted as LP (respectively, SDP). The letter "t" stands for "tight" and precedes the abbreviation when the addi-

tional constraints in (11.4) are included in the relaxation. The LP relaxation is the fastest method by far, but is also the most conservative (i.e., requires a larger input to separate the fault models). The LP relaxation found a separating input 40 to 50 times faster than the t-LP relaxation, but with a norm that is 35 to 50% larger.

Another interesting observation is that the SDP relaxation was more conservative than the t-LP case. This example highlights the importance of the additional constraints (11.4), which include the McCormick relaxations, in directly affecting the tightness of the relaxation. Furthermore, optimization time for the t-LP relaxation scaled much more favorably with the number of models in the simulation than for the SDP relaxation. Note that, in order to reduce the online computational burden, it is always possible to compute an approximate explicit solution to the input separation problem in the same way as described in [212].

Figure 11-2 shows Monte Carlo samples of the outputs for each of the four models when the optimal separating input $\widetilde{u}^*$, computed using the t-LP relaxation, is injected. In the lower right panel, we can clearly see that all output sets are completely separated at the chosen final time point $k = n_t = 4$, which indicates that any sequence of these measurements taken on the interval $\mathcal{T}$ are consistent with at most one model. Thus, a complete fault diagnosis of the system has been achieved.

Another point of interest is that only the blue and green output sets are very close together, meaning these models are the most difficult to separate out of the set of four models. Since there is only a single input to the system, we are limited in the ways that we can shift the position of the output sets. This implies that removing the red or yellow models will not change the optimal separating input. This agrees with the results in Table 11.1, which shows that the input norm is the same for the three and four model cases. The input changes in the two model case because the green set was removed allowing the blue and red sets to be brought closer together with a smaller norm input.

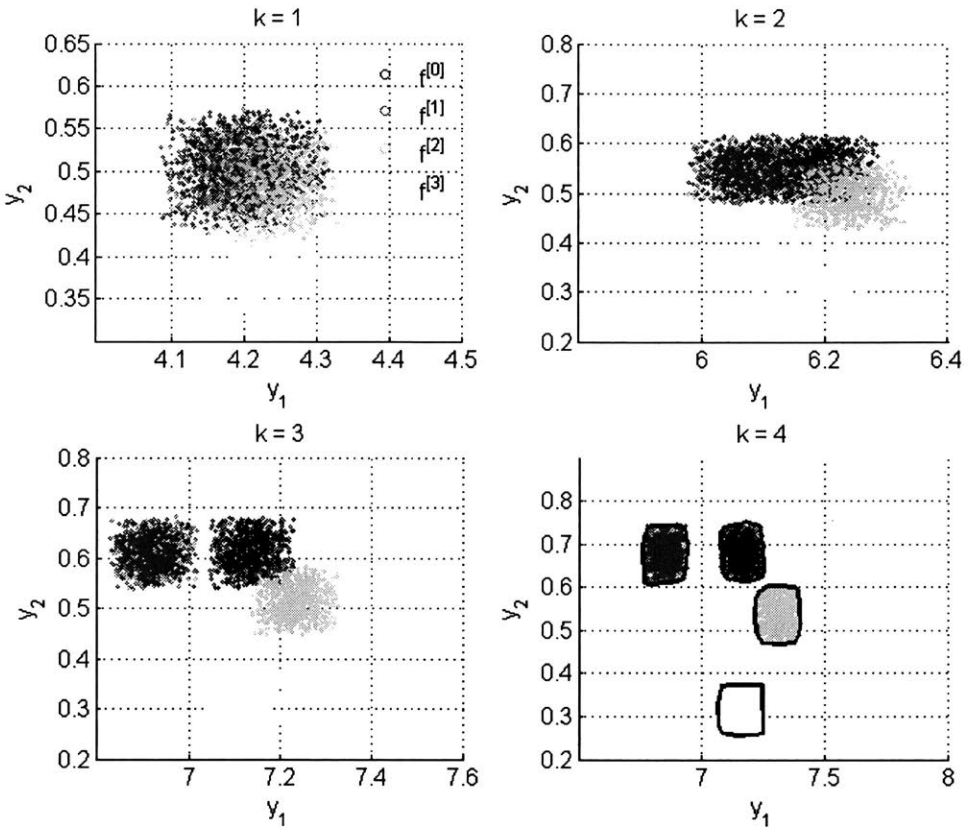


Figure 11-2: 1000 sampled outputs are shown by circles from the nominal and faulty models when the optimal separating input, calculated from the t-LP relaxation, is injected. $f^{[0]}$, $f^{[1]}$, $f^{[2]}$, and $f^{[3]}$ are represented by the blue, red, green, and yellow circles, respectively. The convex hulls (black lines) in the lower-right panel were drawn for clearer illustration.

Table 11.1: Comparison of the proposed method at different levels of convex relaxation. By including the additional constraints in (11.4), t-LP is a tighter relaxation than LP. Computations performed on a Desktop PC (Intel i7, 2.7GHz, 8 GB RAM) running Windows 7 (64-bit) using a single core.

Models, $f^{[i]}$	Relaxation	$\|\widetilde{u}^*\| \times 10^4$	CPU time [s]
$i = \{0, 1\}$	LP	39	8
	t-LP	26	468
	SDP	34	270
$i = \{0, 1, 2\}$	LP	166	16
	t-LP	123	882
	SDP	147	834
$i = \{0, 1, 2, 3\}$	LP	166	25
	t-LP	123	1014
	SDP	147	1710

11.6 Conclusions

A deterministic method is proposed for computing a guaranteed separating input for fault isolation of nonlinear polynomial and rational uncertain systems based on convex relaxations and bilevel optimization. The derived bilevel program has a convex inner program and could be solved to global optimality using standard methods, e.g., branch and bound. An alternative solution method is proposed that uses a nonlinear outer solver to supply inputs to the convex inner program and iteratively step towards a minimum. Although global optimality is not guaranteed *a priori*, the proposed method is much more computationally efficient and showed promising results when applied to an example problem. Furthermore, the general formulation is flexible with respect to the solution method and choice of objective, constraints, and number of possible fault models.

Further work in analyzing the degree of conservatism added to the solution from the convex relaxation methods is recommended. Of particular interest would be an adaptive approach that heuristically adds constraints that are most likely to tighten the relaxation while warm starting the algorithm with the previously computed solution.

Part VI

Conclusions and Suggestions for Future Work

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 12

Conclusions and Future Outlook

12.1 Summary of Contributions

Model predictive control (MPC) is the most advanced control technology widely practiced in industry today. Its strengths are that it is able to handle constraints, large-scale interactive dynamics, and competing performance objectives in a unified manner. However, the MPC framework does not adequately address other practical process control considerations including uncertainty, sensor/actuator failures, and faults. Explicitly accounting for model uncertainty and disturbances in MPC has been an active area of research for the past few decades. This thesis explores *stochastic MPC* as a method for handling probabilistic uncertainty descriptions. A batch crystallization example is presented in Chapter 2 to motivate the need for distributional uncertainty quantification and propagation. The polynomial chaos framework, which is overviewed in Chapter 3, is utilized for efficient propagation of rigorously quantified uncertainty distributions through the dynamic model equations.

State-of-the-art optimization tools for MPC are first reviewed in Chapter 4. Differences between various formulations of the optimal control problem to be solved at each time instant are also discussed. In particular, we elaborate on how the quadratic dynamic matrix control (QDMC) algorithm (developed at Shell Oil in the early 1970s) relates to modern MPC formulations, while highlighting QDMC's computational advantages in large-scale systems. A demonstration of the effectiveness of the QDMC

algorithm for plant-wide control of an end-to-end continuous pharmaceutical manufacturing pilot plant immediately follows.

Methods for incorporating both parameter uncertainty and disturbances using the stochastic MPC framework are then addressed. The focus of Chapter 6 is plant-wide control of uncertain large-scale systems in which we develop an algorithm whose online cost is independent of the state dimension while being able to shape the distribution of the key outputs of the system. This approach showed promising results for a process with over 7000 states. Next, a framework for rejecting disturbances (modeled as random events) with unknown probability distributions is explored. Joint chance constraints (which are inherently non-convex) were handled using convex relaxations combined with an optimal *risk allocation* to reduce conservatism. Concepts from these two approaches are then combined to handle parameter uncertainty and stochastic disturbances in a unified manner. Stability is explored in the unconstrained case using theory for Markov processes. Then we prove that chance constraints may not be satisfied by the closed-loop system even when they are satisfied during each iteration of a receding horizon controller (such as MPC) when applied to systems with uncertain time-invariant parameters. This motivates the exploration of Bayesian learning methods for improved feedback within stochastic MPC. We propose efficiently approximating the solution to this Bayesian estimation problem using polynomial chaos methods. This approach uses online process measurements to reduce the amount of uncertainty in the parameters, and it showed promising results on a process consisting of a series of stirred tank reactors.

The next part of this thesis focuses on the development of failure/fault-tolerant process control methods. Two key additions to the MPC supervisory layer are developed. First, an internal model control (IMC) control structure is introduced that enables the independent design of multi-objective controllers with optimal failure tolerance, meaning that the control structure remains optimal when one or more controllers is taken offline due to sensor or actuator failures. This approach is best-suited for the design of regulatory control layers as they directly communicate with process equipment. We finish by developing one of the first approaches for active

fault diagnosis of nonlinear uncertain systems, which can be used for early diagnosis of non-obvious faults before these faults can lead to catastrophic failure of the process.

12.2 Suggestions for Future Work

The MPC framework was developed to handle many practical process control considerations in a flexible and unified manner. Extensions to account for general parameter uncertainty and disturbances (described by probability distributions) were explored in this thesis. Additionally, methods for handling equipment failures and faults were explored. Work in the following areas are needed for these advanced process control methods to be widely accepted and applied in industry.

Modeling for Control Purposes The most expensive and time-consuming step in the design of any model-based control system is the development of a process model. Modeling a process as an uncertain system requires both a nominal model and an uncertainty description. In this thesis, a probabilistic description of the uncertainty is assumed which further involves the determination of a prior distribution and the likelihood of the uncertainty given data as discussed in Chapter 3. Too narrow of an uncertainty distribution may lead to aggressive control actions that lead to overshoot and possibly instability while too broad of an uncertainty distribution leads to sluggish performance. The iterative framework described in Figure 1-2 is a systematic way to combine physics-based process models with data to obtain a nominal model with an uncertainty description. Applications of this method to real processes will suggest how it can be best utilized and improved upon. A clear understanding of the tradeoff between model accuracy and control quality is needed to determine when to stop iterating around this loop to attempt to improve the model quality.

Chemical processes typically have many inputs and outputs, have high-dimensional distributed dynamics, and are strongly interacting. Techniques must be developed for reducing the dimensionality and complexity of these models so that they are still able to capture the essential behavior of the plant. Since many chemical processes

are inherently nonlinear, these techniques must be able to handle a wide-range of dynamic behavior and should require minimal modeling effort as it is expensive. Efficient uncertainty quantification techniques, such as power series and polynomial chaos methods, should be exploited to reduce computational burden.

Nonlinear Optimization Methods The MPC methods explored in this thesis focus on linear models as they usually yield good results in the neighborhood of a specific operating point. This local operating point is typically decided by a higher layer in the control hierarchy (such as the real-time optimizer) that optimizes profit using steady-state nonlinear process models. Linear models can be identified from data or obtained by linearization of a first-principles model. The incorporation of uncertainty into these models also helps mitigate some of the errors due to model identification/linearization. However, demand for higher product quality, tighter specifications, and tougher environmental regulations necessitate good closed-loop performance over a wider range of operating conditions including startup and shutdown. This fact, combined with the inherent nonlinearity in most real chemical and biological processes, motivates the development of so-called nonlinear MPC (NMPC). NMPC refers to MPC schemes involving a nonlinear objective function and/or nonlinear constraints that typically arise due to the use of nonlinear models.

A key advantage of linear MPC is that it can be directly formulated as a convex optimization problem, which can efficiently be solved to global optimality (since any local minimum detected must also be a global minimum). However, NMPC methods are generally nonconvex optimization problems that are very expensive to solve to global optimality. Development of practical algorithms for NMPC is needed so that they can be successfully applied to real problems. Advances in nonlinear optimization methods are also needed to provide fast solutions with guaranteed convergence.

One promising route is polynomial optimization as many nonlinear systems can be approximated as polynomial systems by expanding all nonlinear functions using a Taylor series that is truncated to a finite number of terms. Taylor's theorem can then be used to rigirously bound the approximation error to provide guaranteed accuracy

(see [97] for a perspective on polynomial optimization within the context of MPC).

Unmodeled Dynamics Models of chemical processes are never perfect (commonly referred to as *plant/model mismatch*). The two most common sources of plant/model mismatch are *parameter uncertainty* and *unmodeled dynamics*. In this general setting, the model may be of the form $x_{k+1} = f^{\text{model}}(x_k, u_k)$ while the true plant dynamics evolve as $z_{k+1} = f^{\text{plant}}(z_k, u_k)$ where the structure and parameter values of f^{model} differs from that of f^{plant} and the number of states are different in the model and plant. Not much can be done to cope with this broad form of uncertainty unless further assumptions are made. An example is given in [241], which assumes that the bounds on the plant/model mismatch can be adequately characterized from data.

All of the methods introduced in this thesis assume that the structure of the model is correct such that only the parameters are unknown. This effectively lumps the effect of parameter and structural uncertainty when estimating the distribution of the parameters. If possible, methods for quantifying structural uncertainty and parameter uncertainty simultaneously should be developed. For certain processes, it is known that the model structure is incorrect (for example, not every reaction is known or can be modeled in many bioreactors) and this may result in significant errors when identifying the model. As such, future research should explore ways to handle structural uncertainty within the proposed control framework. The main difficulties are that this error is not an inherently probabilistic quantity (it is simply unknown) and it is likely to vary with time.

Time-varying Parameters Certain chemical processes involve unknown parameters that vary with time. Some examples are catalyst degradation in a reactor, denaturation of enzymes in a bioreactor, and heat exchanger fouling. When these parameters vary slowly with respect to the time window of interest, then they can assumed to be approximately time-invariant meaning polynomial chaos propagation tools can be applied. If this is not the case, other methods must be explored for handling uncertainty that varies with time. When the parameters are modeled as a

stochastic process, the correlation of the parameters in time must be captured with a joint distribution function $f(\theta_k, \theta_{k-1}, \theta_{k-2}, \cdots)$. This joint distribution greatly simplifies when the parameters are assumed to be independent and identically distributed, however, this may not be a realistic assumption since it is unlikely that the parameter can take on any possible value at every time step. Future work is needed to find ways to estimate this joint distribution efficiently from data and to develop techniques for propagating this joint distribution through the dynamic system model. One route may be to apply Bayesian learning (see Chapter 9) with a forgetting factor for the prior distribution chosen based on the rate at which the parameter changes.

Fault-tolerant Control The methods described in this thesis tackle robust control, tolerance to equipment failure, and active diagnosis of critical system faults, which can be combined to produce a fault-tolerant control method. Although the methods were developed with simplicity of integration in mind, some open questions remain regarding how best to intertwine these various methods. Since control and fault diagnosis typically have opposing objectives, it remains an open question how best to integrate these tools. The current standard is to wait to apply active fault diagnosis (aka input design) methods until a process monitoring method (usually purely based on data and statistics) detects faulty behavior. Future research might explore embedding fault diagnosis within the MPC framework (using constraints for example) so as to better tradeoff performance with fault diagnosability. In addition, *closed-loop* input design methods should be explored as a way to reduce conservatism of the proposed method of computing a single input sequence offline.

Bibliography

[1] *ILOG CPLEX 11.0 User's Manual.* ILOG SA, Gentilly, France, 2007.

[2] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*, pages 345–369. Springer, 2009.

[3] I. Andjelkovic, K. Sweetingham, and S. L. Campbell. Active fault detection in nonlinear systems using auxiliary signals. In *Proceedings of the American Control Conference*, pages 2142–2147, Seattle, WA, USA, June 2008.

[4] D. Arbuckle and A. A. G. Requicha. Active self-assembly. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 896–901, New Orleans, LA, USA, 2004.

[5] M. Athans. The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. *IEEE Transactions on Automatic Control*, 16(6):529–552, 1971.

[6] D. Axehill. Controlling the level of sparsity in MPC. *Systems and Control Letters*, 76:1–7, 2015.

[7] M. R. A. Bakar, Z. K. Nagy, and C. D. Rielly. Seeded batch cooling crystallization with temperature cycling for the control of size uniformity and polymorphic purity of sulfathiazole crystals. *Organic Process Research & Development*, 13(6):1343–1356, 2009.

[8] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith. μ-analysis and synthesis toolbox. *MUSYN Inc. and The MathWorks, Natick MA*, 1993.

[9] J. F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*, volume 30. Kluwer Academic Publisher, Boston, London, Dordrecht, 1998.

[10] R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal. Quadratic programming algorithms for large-scale model predictive control. *Journal of Process Control*, 12(7):775–795, 2002.

[11] R. A. Bartlett, A. Wächter, and L. T. Biegler. Active set vs. interior point strategies for model predictive control. In *Proceedings of the American Control Conference*, pages 4229–4233, Chicago, IL, USA, June 2000.

[12] H. Bazargan. *An Efficient Polynomial Chaos-based Proxy Model for History Matching and Uncertainty Quantification of Complex Geological Structures.* PhD thesis, Heriot–Watt University, 2014.

[13] J. V. Beck and K. J. Arnold. *Parameter Estimation in Engineering and Science.* John Wiley & Sons, New York, 1977.

[14] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in Identification and Control*, pages 207–226. Springer, 1999.

[15] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[16] A. Ben-Tal, S. Boyd, and A. Nemirovski. Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Journal of Mathematical Programming*, 107:63–89, 2006.

[17] F. Benaskar, A. Ben-Abdelmoumen, N. G. Patil, E. V. Rebrov, J. Meuldijk, L. A. Hulshof, V. Hessel, U. Krtschill, and J. C. Schouten. Cost analysis for a continuously operated fine chemicals production plant at 10 kg/day using a combination of microprocessing and microwave heating. *Journal of Flow Chemistry*, 1:74–89, 2011.

[18] B. Benyahia, R. Lakerveld, and P. I. Barton. A plant-wide dynamic model of a continuous pharmaceutical process. *Industrial Engineering & Chemistry Research*, 51:15393–15421, 2012.

[19] B. W. Bequette. *Process Control: Modeling, Design, and Simulation.* Prentice Hall Professional, New Jersey, 2003.

[20] D. Bernardini and A. Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. In *Proceedings of the IEEE Conference on Decision and Control held jointy with the Chinese Control Conference*, pages 6333–6338, Shanghai, China, December 2009.

[21] D. P. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. *European Journal of Control*, 11(4):310–334, 2005.

[22] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability.* Athena Scientific Belmont, MA, 2002.

[23] D. Bertsimas and D. B. Brown. Constrained stochastic LQC: A tractable approach. *IEEE Transactions on Automatic Control*, 52:1826–1841, 2007.

[24] D. Bertsimas and M. Sim. Tractable approximations to robust conic optimization problems. *Mathematical Programming*, 107(1-2):5–36, 2006.

[25] S. W. Bishnoi, C. J. Rozell, C. S. Levin, M. K. Gheith, B. R. Johnson, D. H. Johnson, and N. J. Halas. All-optical nanoscale pH meter. *Nano Letters*, 6(8):1687–1692, 2006.

[26] L. Blackmore and M. Ono. Convex chance constrained predictive control without sampling. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 7–21, Chicago, IL, USA August 2009.

[27] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26(3):502–517, 2010.

[28] E. Blanchard, A. Sandu, and C. Sandu. Parameter estimation method using an extended Kalman filter. In *Proceedings of the Joint North America, Asia-Pacific ISTVS Conference, and Annual Meeting of Japanese Society for Terramechanics*, pages 23–26, 2007.

[29] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[30] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control.* Springer, Heidelberg, Berlin, 2006.

[31] E.-K. Boukas. *Control of Singular Systems with Random Abrupt Changes.* Springer-Verlag, Berlin, 2008.

[32] R. D. Braatz. *Robust Loopshaping for Process Control.* PhD thesis, California Institute of Technology, Pasadena, California, 1993.

[33] R. D. Braatz. Internal model control. In *The Control Handbook*, pages 215–224. CRC Press, Florida, 1996.

[34] R. D. Braatz, R. C. Alkire, E. Seebauer, E. Rusli, R. Gunawan, T. O. Drews, X. Li, and Y. He. Perspectives on the design and control of multiscale systems. *Journal of Process Control*, 16(3):193–204, 2006.

[35] R. D. Braatz, R. C. Alkire, E. G. Seebauer, T. O. Drews, E. Rusli, M. Karulkar, F. Xue, Y. Qin, M. Y. L. Jung, and R. Gunawan. A multiscale systems approach to microelectronic processes. *Computers & Chemical Engineering*, 30(10):1643–1656, 2006.

[36] C. Brosilow and N. Markale. Model predictive cascade control and its implications for classical and imc cascade control. In *AIChE Annual Meeting*, Miami, FL, USA, September 1992.

[37] G. C. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications*, 130(1):1–22, 2006.

[38] G. C. Calafiore and L. Fagiano. Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1):219–224, 2013.

[39] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[40] S. L. Campbell and R. Nikoukhah. *Auxilary Signal Design for Failure Detection.* Princeton University Press, New Jersey, 2004.

[41] M. Cannon, B. Kouvaritakis, S. Raković, and Q. Cheng. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, 56:194–200, 2011.

[42] M. Cannon, B. Kouvaritakis, and X. Wu. Probabilistic constrained MPC for systems with multiplicative and additive stochastic uncertainty. In *Proceedings of the IFAC World Congress*, pages 15297–15302, Seoul, Korea, July 2008.

[43] D. Chatterjee and J. Lygeros. On stability and performance of stochastic predictive control techniques. *IEEE Transactions on Automatic Control*, 60(2):509–514, 2015.

[44] D. Chatterjee, F. Ramponi, P. Hokayem, and J. Lygeros. On mean square boundedness of stochastic linear systems with bounded controls. *Systems and Control Letters*, 61(2):375–380, 2012.

[45] Q. Chen, S. C. Bae, and S. Granick. Directed self-assembly of a colloidal kagome lattice. *Nature*, 469(7330):381–384, 2011.

[46] Q. Chen, J. K. Whitmer, S. Jiang, S. C. Bae, E. Luijten, and S. Granick. Supracolloidal reaction kinetics of Janus spheres. *Science*, 331(6014):199–202, 2011.

[47] L. H. Chiang, E. L. Russell, and R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems.* Springer, London, 2001.

[48] S. H. Chung, D. L. Ma, and R. D. Braatz. Optimal seeding in batch crystallization. *The Canadian Journal of Chemical Engineering*, 77:590–596, 1999.

[49] M. J. Cima and L. G. Cima. Tissue regeneration matrices by solid free form fabrication techniques. US Patent 5,518,680, May 21 1996.

[50] W. Coffey, Y. P. Kalmykov, and J. T. Waldron. *The Langevin Equation: With Applications to Stochastic Problems in Physics, Chemistry, and Electrical Engineering.* World Scientific, Singapore, second edition, 2004.

[51] L. Cognet, D. A. Tsyboulski, J.-D. R. Rocha, C. D. Doyle, J. M. Tour, and R. B. Weisman. Stepwise quenching of exciton fluorescence in carbon nanotubes by single-molecule reactions. *Science*, 316(5830):1465–1468, 2007.

[52] P. D. Couchman, M. Cannon, and B. Kouvaritakis. Stochastic MPC with inequality stability constraints. *Automatica*, 42:2169–2174, 2006.

[53] R. F. Curtain and H. Zwart. *An Introduction to Infinite-Dimensional Linear Systems Theory.* Springer-Verlag, New York, 1995.

[54] C. R. Cutler and B. L. Ramaker. Dynamic matrix control - A computer control algorithm. In *AIChE Annual Meeting*, Houston, TX, USA, April, 1979.

[55] P. F. Damasceno, M. Engel, and S. C. Glotzer. Predictive self-assembly of polyhedra into complex structures. *Science*, 337(6093):453–457, 2012.

[56] G. De Nicolao. System identification: Problems and perspectives. In *12th Workshop on Qualitative Reasoning*, pages 379–386, 1997.

[57] J. G. Van de Vusse. Plug-flow type reactor versus tank reactor. *Chemical Engineering Science*, 19:994–997, 1964.

[58] A. Dehghani, A. Lanzon, and B. D. O. Anderson. A two-degree-of-freedom H_∞ control design method for robust model matching. *International Journal of Robust and Nonlinear Control*, 16:467–483, 2006.

[59] S. X. Ding. *Model-based Fault Diagnosis Techniques – Design Schemes, Algorithms and Tools*. Springer, Berlin, Heidelberg, 2008.

[60] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 668–674, Maui, HI, USA, December 2012.

[61] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23:756–757, 1978.

[62] M. Du and P. Mhaskar. Active fault isolation of nonlinear process systems. *AIChE Journal*, 59(7):2435–2453, 2013.

[63] E. Edlund, O. Lindgren, and M. N. Jacobi. Predicting self-assembled patterns on spheres with multicomponent coatings. *Soft Matter*, 10(17):2955–2960, 2014.

[64] J. A. Egea, D. Henriques, T. Cokelaer, A. F. Villaverde, A. MacNamara, D.-P. Danciu, J. R. Banga, and J. Saez-Rodriguez. MEIGO: An open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinformatics*, 15:1–9, 2014.

[65] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:317–339, 2012.

[66] L. Fagion and M. Khammash. Nonlinear stochastic model predictive control via regularized polynomial chaos expansions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 142–147, Maui, HI, USA, December 2012.

[67] M. Farina, L. Giulioni, L. Magni, and R. Scattolini. An approach to output-feedback MPC of stochastic linear discrete-time systems. *Automatica*, 55:140–149, 2015.

[68] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[69] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[70] K. A. Fichthorn and W. H. Weinberg. Theoretical foundations of dynamical Monte Carlo simulations. *The Journal of Chemical Physics*, 95(2):1090–1096, 1991.

[71] R. Findeisen and F. Allgöwer. Nonlinear model predictive control for index-one DAE systems. In *Nonlinear Model Predictive Control*, pages 145–161. Springer, 2000.

[72] J. R. Fisher. *Stability Analysis and Control of Stochastic Dynamic Systems Using Polynomial Chaos.* PhD thesis, Texas A&M University, 2008.

[73] J. R. Fisher and R. Bhattacharya. Linear quadratic regulation of systems with stochastic parameter uncertainties. *Automatica*, 45(12):2831–2841, 2009.

[74] R. Fletcher. *Practical methods of optimization.* John Wiley and Sons, 2013.

[75] G. Frison and J. B. Jorgensen. A fast condensing method for solution of linear-quadratic control problems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7715–7720, Florence, Italy, December 2013.

[76] C. E. Garcia and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.

[77] S. Gayadeen and W. Heath. An internal model control approach to mid-ranging control. In *Proceedings of the IFAC International Symposium on Advanced Control of Chemical Processes*, pages 542–547, Istanbul, Turkey, July 2009.

[78] C. W. Gear. Differential-algebraic equation index transformation. *SIAM Journal on Scientific Computing*, 9:39–47, 1988.

[79] J. J. Gertler. *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, New York, 1998.

[80] T. Geyer, F. D. Torrisi, and M. Morari. Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7):1728–1740, 2008.

[81] R. Ghanem and P. Spanos. *Stochastic Finite Elements - A Spectral Approach.* Springer-Verlag, New York, 1991.

[82] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[83] L. Giovanini. Cooperative-feedback control. *ISA Transactions*, 46:289–302, 2007.

[84] S. C. Glotzer and J. A. Anderson. Nanoparticle assembly: Made to order. *Nature Materials*, 9(11):885–887, 2010.

[85] S. C. Glotzer and M. J. Solomon. Anisotropy of building blocks and their assembly into complex structures. *Nature Materials*, 6(8):557–562, 2007.

[86] S. C. Glotzer, M. J. Solomon, and N. A. Kotov. Self-assembly: From nanoscale to microscale colloids. *AIChE Journal*, 50(12):2978–2985, 2004.

[87] L. Goh, K. Chen, V. Bhamidi, G. He, N. C. S. Kee, P. J. A. Kenis, C. F. Zukoski III, and R. D. Braatz. A stochastic model for nucleation kinetics determination in droplet-based microfluidic systems. *Crystal Growth & Design*, 10(6):2515–2521, 2010.

[88] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, volume 140, pages 107–113, April 1993.

[89] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42:523–533, 2006.

[90] S. Granick, S. Jiang, and Q. Chen. Janus particles. *Physics Today*, 62:68–69, 2009.

[91] M. Grant, S. Boyd, and Y. Ye. CVX: Matlab Software for Disciplined Convex Programming, 2008. http://cvxr.com/cvx.

[92] M. J. Grimble. Two-degrees of freedom feedback and feedforward optimal control of multivariable stochastic systems. *Automatica*, 24:809–817, 1988.

[93] M. Grzelczak, J. Vermant, E. M. Furst, and L. M. Liz-Marzán. Directed self-assembly of nanoparticles. *ACS Nano*, 4(7):3591–3605, 2010.

[94] R. Gunawan, M. Y. L. Jung, E. G. Seebauer, and R. D. Braatz. Maximum a posteriori estimation of transient enhanced diffusion energetics. *AIChE journal*, 49:2114–2123, 2003.

[95] M. F. Hagan. *Modeling viral capsid assembly*, volume 155. John Wiley & Sons, Inc., Hoboken, New Jersey, 2014.

[96] J. Haleblian and W. McCrone. Pharmaceutical applications of polymorphism. *Journal of Pharmaceutical Sciences*, 58(8):911–929, 1969.

[97] E. Harinath, L. C. Foguth, J. A. Paulson, and R. D. Braatz. Nonlinear model predictive control using polynomial optimization methods. In *Proceedings of the American Control Conference*, pages 1–6, Boston, MA, USA, July 2016.

[98] S. Haus, S. Jabbari, T. Millat, H. Janssen, R. J. Fischer, H. Bahl, J. R. King, and O. Wolkenhauer. A systems biology approach to investigate the effect of pH-induced gene regulation on solvent production by Clostridium acetobutylicum in continuous culture. *BMC Systems Biology*, 5(10), 2011.

[99] M. A. Henson, B. A. Ogunnaike, and J. S. Schwaber. Habituating control strategies for process control. *AIChE Journal*, 41:604–617, 1995.

[100] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proceedings of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 Zurich, Switzerland, July 2013.

[101] M. W. Hermanto, M.-S. Chiu, X.-Y. Woo, and R. D. Braatz. Robust optimal control of polymorphic transformation in batch crystallization. *AIChE Journal*, 53(10):2643–2650, 2007.

[102] M. W. Hermanto, N. C. Kee, R. B. H. Tan, M.-S. Chiu, and R. D. Braatz. Robust bayesian estimation of kinetics for the polymorphic transformation of L-glutamic acid crystals. *AIChE journal*, 54(12):3248–3259, 2008.

[103] D. H. Van Hessem and O. H. Bosgra. A full solution to the constrained stochastic closed-loop MPC problem via state and innovations feedback and its receding horizon implementation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 929–934, Maui, HI, USA, December 2003.

[104] P. Hokayem, D. Chatterjee, and J. Lygeros. On stochastic receding horizon control with bounded control inputs. In *Proceedings of the IEEE Conference on Decision and Control held jointy with the Chinese Control Conference*, pages 6359–6364, Shanghai, China, December 2009.

[105] P. Hokayem, E. Cinquemani, D. Chatterjee, F. Ramponi, and J. Lygeros. Stochastic receding horizon control with output feedback and bounded controls. *Automatica*, 48:77–88, 2012.

[106] L. Hong, A. Cacciuto, E. Luijten, and S. Granick. Clusters of amphiphilic colloidal spheres. *Langmuir*, 24(3):621–625, 2008.

[107] I. G. Horn, J. R. Arulandu, C. J. Gombas, J. G. VanAntwerp, and R. D. Braatz. Improved filter design in internal model control. *Industrial & Engineering Chemistry Research*, 35:3437–3441, 1996.

[108] S. Hovland and J. T. Gravdahl. Complexity reduction in explicit MPC through model reduction. In *Proceedings of the IFAC World Congress on Automatic Control*, pages 7711–7716, Seoul, Korea, July 2008.

[109] S. H. Hsu, G. V. Reklaitis, and V. Venkatasubramanian. Modeling and control of roller compaction for pharmaceutical manufacturing. Part II: Control system design. *Journal of Pharmaceutical Innovation*, 5:24–36, 2010.

[110] T. Huschto and S. Sager. Stochastic optimal control in the perspective of the Wiener chaos. In *Proceedings of the European Control Conference*, pages 3059–3064, Orlando, FL, December 2013.

[111] O. Idan, A. Lam, J. Kamcev, J. Gonzales, A. Agarwal, and H. Hess. Nanoscale transport enables active self-assembly of millimeter-scale wires. *Nano Letters*, 12(1):240–245, 2011.

[112] R. Isermann. *Fault Diagnosis Systems: An Introduction from Detection to Fault Tolerance*. Springer-Verlag, Berlin, 2006.

[113] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari. Embedded online optimization for model predictive control at megahertz rates. *IEEE Transactions on Automatic Control*, 59(12):3238–3251, 2014.

[114] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides. A sparse and condensed QP formulation for predictive control of LTI systems. *Automatica*, 48(5):999–1002, 2012.

[115] M. Jiang, X. Zhu, M. C. Molaro, M. L. Rasche, H. Zhang, K. Chadwick, D. M. Raimondo, K.-K. K. Kim, L. Zhou, Z. Zhu, M. H. Wong, D. O'Grady, D. Hebrault, J. Tedesco, and R. D. Braatz. Modification of crystal shape through deep temperature cycling. *Industrial & Engineering Chemistry Research*, 53(13):5325–5336, 2014.

[116] T. Johansen, W. Jackson, R. Schreiber, and P. Tøndel. Hardware synthesis of explicit model predictive controllers. *IEEE Transactions on Control Systems Technology*, 15(1):191–197, 2007.

[117] M. R. Jones and C. A. Mirkin. Materials science: Self-assembly gets new direction. *Nature*, 491(7422):42–43, 2012.

[118] J. J. Juárez and M. A. Bevan. Feedback controlled colloidal self-assembly. *Advanced Functional Materials*, 22(18):3833–3839, 2012.

[119] D. Kagan, P. Calvo-Marzal, S. Balasubramanian, S. Sattayasamitsathit, K. M. Manesh, G.-U. Flechsig, and J. Wang. Chemical sensing based on catalytic nanomotors: Motion-based detection of trace silver. *Journal of the American Chemical Society*, 131(34):12082–12083, 2009.

[120] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[121] D. G. Kendall. Stochastic processes and population growth-symposium on stochastic processes. *Journal of the Royal Statistical Society: Series B*, 11(2):230–264, 1949.

[122] E. C. Kerrigan and J. M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Proceedings of the UKACC International Conference (Control 2000)*, 2000.

[123] I. G. Kevrekidis, C. W. Gear, and G. Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7):1346–1355, 2004.

[124] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

[125] K.-K. K. Kim and R. D. Braatz. Generalized polynomial chaos expansion approaches to approximate stochastic receding horizon control with applications to probabilistic collision checking and avoidance. In *Proceedings of the IEEE International Conference on Control Applications*, pages 350–355, 2012.

[126] K.-K. K. Kim, D. E. Shen, Z. K. Nagy, and R. D. Braatz. Wiener's polynomial chaos for the analysis and control of nonlinear dynamical systems with probabilistic uncertainties. *IEEE Control Systems Magazine*, 33:58–67, 2013.

[127] E. Klavins. Programmable self-assembly. *IEEE Control Systems Magazine*, 27(4):43–56, 2007.

[128] H.-A. Klok and S. Lecommandoux. Supramolecular materials via block copolymer self-assembly. *Advanced Materials*, 13(16):1217–1229, 2001.

[129] M. V. Konnik. *Online Constrained Receding Horizon Control for Astronomical Adaptive Optics*. PhD thesis, School of Electrical Engineering and Computer Science, the University of Newcastle, Australia, 2013.

[130] B. Kouvaritakis and M. Cannon. Developments in robust and stochastic predictive control in the presence of uncertainty. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 1(2):021003–021003–9, 2015.

[131] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

[132] P. R. Krishnaswamy, G. P. Rangaiah, R. K. Jha, and P. B. Deshpande. When to use cascade control? *Industrial & Engineering Chemistry Research*, 29:2163–2166, 1990.

[133] S. K. Kufer, E. M. Puchner, H. Gumpp, T. Liedl, and H. E. Gaub. Single-molecule cut-and-paste surface assembly. *Science*, 319(5863):594–596, 2008.

[134] M. Kvasnica, J. Hledík, I. Rauová, and M. Fikar. Complexity reduction of explicit model predictive control via separation. *Automatica*, 49(6):1776–1781, 2013.

[135] M. Kvasnica, J. Löfberg, M. Herceg, L. Cirka, and M. Fikar. Low-complexity polynomial approximation of explicit MPC via linear programming. In *Proceedings of the American Control Conference*, pages 4713–4718, Baltimore, MD, USA, June 2010.

[136] R. Lakerveld, B. Benyahia, R. D. Braatz, and P. I. Barton. Model-based design of a plant-wide control strategy for a continuous pharmaceutical pilot plant. *AIChE Journal*, 59:3671–3685, 2013.

[137] R. Lakerveld, G. Stephanopoulos, and P. I. Barton. A master-equation approach to simulate kinetic traps during directed self-assembly. *The Journal of Chemical Physics*, 136(18):184109, 2012.

[138] W. E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 596–604, Honolulu, HI, USA, December 1990.

[139] J. H. Lee, M. Morari, and C. E. Garcia. State-space interpretation of model predictive control. *Automatica*, 30(4):707–717, 1994.

[140] H. Leuenberger. New trends in the production of pharmaceutical granules: Batch versus continuous processing. *Pharmaceutical Research*, 52:289–296, 2001.

[141] Q. Li and J. A. Lewis. Nanoparticle inks for directed assembly of three-dimensional periodic structures. *Advanced Materials*, 15(19):1639–1643, 2003.

[142] S. Li, K. Y. Lim, and D. G. Fisher. A state space formulation for model predictive control. *AIChE Journal*, 35(2):241–249, 1989.

[143] K. G. Libbrecht. The physics of snow crystals. *Reports on Progress in Physics*, 68(4):855, 2005.

[144] D. J. N. Limebeer, E. M. Kasenally, and J. D. Perkins. On the design of robust two degree of freedom controllers. *Automatica*, 29:157–168, 1993.

[145] R. A. Lionberger, S. L. Lee, L. Lee, A. Raw, and L. X. Yu. Quality by design: Concepts for ANDAs. *The AAPS Journal*, 10:268–275, 2008.

[146] T. Liu, W. Zhang, and F. Gao. Analytical two-degrees-of-freedom (2-dof) decoupling control scheme for multi-input multi-output (mimo) processes with time delays. *Industrial & Engineering Chemistry Research*, 46:6546–6557, 2007.

[147] X. Liu, Y. Zhang, D. K. Goswami, J. S. Okasinski, K. Salaita, P. Sun, M. J. Bedzyk, and C. A. Mirkin. The controlled evolution of a polymer single crystal. *Science*, 307(5716):1763–1766, 2005.

[148] L. Ljung. *System Identification: Theory for the User.* Prentice Hall PRC, New Jersey, 1999.

[149] P. Lundström, J. H. Lee, M. Morari, and S. Skogestad. Limitations of dynamic matrix control. *Computers and Chemical Engineering*, 19:409–421, 1995.

[150] W. L. Luyben. Parallel cascade control. *Industrial & Engineering Chemistry Fundamentals*, 12:463–467, 1973.

[151] D. L. Ma and R. D. Braatz. Worst-case analysis of finite-time control policies. *IEEE Transactions on Control Systems Technology*, 9:766–774, 2001.

[152] Y. Ma, S. Vichik, and F. Borrelli. Fast stochastic MPC with optimal risk allocation applied to building control systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7559–7564, Maui, HI, USA, December 2012.

[153] R. Madankan, P. Singla, T. Singh, and P. D. Scott. Polynomial-chaos-based Bayesian approach for state and parameter estimations. *Journal of Guidance, Control, and Dynamics*, 36(4):1058–1074, 2013.

[154] G. Mapili, Y. Lu, S. Chen, and K. Roy. Laser-layered microfabrication of spatially patterned functionalized tissue-engineering scaffolds. *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, 75(2):414–424, 2005.

[155] A. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications.* Academic Press, New York, 1979.

[156] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, 2007.

[157] S. Mascia, P. L. Heider, H. Zhang, R. Lakerveld, B. Benyahia, P. I. Barton, R. D. Braatz, C. L. Cooney, J. M. B. Evans, T. F. Jamison, K. F. Jensen, A. S. Myerson, and B. L. Trout. End-to-end continuous manufacturing of pharmaceuticals: Integrated synthesis, purification, and final dosage formation. *Angewandte Chemie International Edition*, 52:12359–12363, 2013.

[158] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine*, 31(3):52–65, 2011.

[159] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[160] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[161] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[162] E. S. Meadows. *Stability and Continuity of Nonlinear Model Predictive Control.* PhD thesis, The Unversity of Texas at Austin, 1994.

[163] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[164] A. Mesbah. Stochastic model predictive control: An overview and perspective for future research. *IEEE Control Systems Magazine*, In Press.

[165] A. Mesbah, A. N. Ford Versypt, X. Zhu, and R. D. Braatz. Nonlinear model-based control of thin-film drying for continuous pharmaceutical manufacturing. *Industrial & Engineering Chemistry Research*, 53(18):7447–7460, 2013.

[166] A. Mesbah, M. Kishida, and R. D. Braatz. Design of multi-objective failure-tolerant control systems for infinite-dimensional systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3006–3013, Florence, Italy, December 2013.

[167] A. Mesbah, Z. K. Nagy, A. E. M. Huesman, H. J. M. Kramer, and P. M. J. Van den Hof. Nonlinear model-based control of a semi-industrial batch crystallizer using a population balance modeling framework. *IEEE Transactions on Control Systems Technology*, 20:1188–1201, 2012.

[168] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz. Active fault diagnosis for nonlinear systems with probabilistic uncertainties. *Proceedings of the IFAC World Congress*, pages 7079–7084, Cape Town, South Africa, August 2014.

[169] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *Proceedings of the American Control Conference*, pages 2413–2419, Portland, Oregon, USA, June 2014.

[170] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stabiliy.* Cambridge University Press, Cambridge, 2009.

[171] S. M. Miller and J. B. Rawlings. Model identification and control strategies for batch cooling crystallizers. *AIChE Journal*, 40:1312–1327, 1994.

[172] A. Mitsos, P. Lemonidis, and P. I. Barton. Global solution of bilevel programs with a nonconvex inner program. *Journal of Global Optimization*, 42:475–513, 2008.

[173] J. Moon, A. C. Caballero, L. Hozer, Y.-M. Chiang, and M. J. Cima. Fabrication of functionally graded reaction infiltrated SiC–Si composite by three-dimensional printing 3DP^{TM} process. *Materials Science and Engineering: A*, 298(1):110–119, 2001.

[174] M. Morari and J. H. Lee. Model predictive control: Past, present and future. *Computers & Chemical Engineering*, 23:667–682, 1999.

[175] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice-Hall, New Jersey, 1989.

[176] T. Mühlpfordt, J. A. Paulson, R. D. Braatz, and R. Findeisen. Output feedback model predictive control with probabilistic uncertainties for linear systems. In *Proceedings of the American Control Conference*, pages 2035–2040, Boston, MA, USA, July 2016.

[177] K. R. Muske and J. B. Rawlings. Linear model predictive control of unstable processes. *Journal of Process Control*, 3(2):85–96, 1993.

[178] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[179] Z. K. Nagy and R. D. Braatz. Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. *IEEE Transactions on Control Systems Technology*, 11:694–704, 2003.

[180] Z. K. Nagy and R. D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control*, 14:411–422, 2004.

[181] Z. K. Nagy and R. D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17:229–240, 2007.

[182] Z. K. Nagy and R. D. Braatz. Advances and new directions in crystallization control. *Annual Review of Chemical and Biomolecular Engineering*, 3:55–75, 2012.

[183] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41:35–52, 2009.

[184] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[185] Y. Nesterov. A method of solving a convex programming problem with convergence rate O(1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[186] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science and Business Media, 2004.

[187] R. Nikoukhah. Guaranteed active failure detection and isolation for linear dynamical systems. *Automatica*, 34(11):1345–1358, 1998.

[188] R. Nikoukhah and S. L. Campbell. Auxiliary signal design for active failure detection in uncertain linear systems with a priori information. *Automatica*, 42(2):219–228, 2006.

[189] R. Nikoukhah, S. L. Campbell, and F. Delebecque. Detection signal design for failure detection: A robust approach. *International Journal of Adaptive Control and Signal Processing*, 14:701–724, 2000.

[190] B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control.* Oxford University Press, New York, 1994.

[191] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications.* Springer-Verlag Berlin Heidelberg, New York, 2003.

[192] F. Oldewurtel, C. N. Jones, and M. Morari. A tractable approximation of chance constrained stochastic MPC, based on affine disturbance feedback. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4731–4736, Cancun, Mexico, December 2008.

[193] F. Oldewurtel, A. Parisio, C. N. Jones, , M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy efficient building climate control using stochastic model predictive control and weather predictions. In *Proceedings of the American Control Conference*, pages 2100–5105, Baltimore, MD, USA, June 2010.

[194] C.-A. Palma, M. Cecchini, and P. Samorì. Predicting self-assembly: From empirism to determinism. *Chemical Society Reviews*, 41(10):3713–3730, 2012.

[195] S. Pankavich, Z. Shreif, Y. Miao, and P. Ortoleva. Self-assembly of nanocomponents into composite structures: Derivation and simulation of Langevin equations. *The Journal of Chemical Physics*, 130(19):194115, 2009.

[196] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

[197] R. J. Patton and J. Chen. Observer-based fault detection and isolation: Robustness and applications. *Control Engineering Practice*, 5(5):671–682, 1997.

[198] J. A. Paulson, E. Harinath, L. C. Foguth, and R. D. Braatz. Nonlinear model predictive control of systems with probabilistic time-invariant uncertainties. In *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control*, pages 937–943, Seville, Spain, September 2015.

[199] J. A. Paulson, A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz. Fast stochastic model predictive control of high-dimensional systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2802–2809, Los Angeles, CA, USA, December 2014.

[200] J. A. Paulson, A. Mesbah, X. Zhu, M. C. Molaro, and R. D. Braatz. Control of self-assembly in micro- and nano-scale systems. *Journal of Process Control*, 27:38–49, 2015.

[201] J. A. Paulson, D. M. Raimondo, R. Findeisen, R. D. Braatz, and S. Streif. Guaranteed active fault diagnosis for uncertain nonlinear systems. In *Proceedings of the European Control Conference*, pages 926–931, Strasbourg, France, June 2014.

[202] J. A. Paulson, S. Streif, and A. Mesbah. Stability for receding-horizon stochastic model predictive control. In *Proceedings of the American Control Conference*, pages 937–943, Chicago, IL, USA, July 2015.

[203] P. Poechlauer, J. Manley, R. Broxterman, B. Gregertsen, and M. Ridemark. Continuous processing in the manufacture of active pharmaceutical ingredients and finished dosage forms: An industry perspective. *Organic Process Research & Development*, 16:1586–1590, 2013.

[204] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, New York, 1998.

[205] L. Popiel, T. Matsko, and C. Brosilow. Coordinated control. In M. Morari and T. J. McAvoy, editors, *Chemical Process Control III*, pages 295–319, 1986.

[206] M. Pottmann, M. A. Henson, B. A. Ogunnaike, and J. S. Schwaber. A parallel control strategy abstracted from the baroreceptor reflex. *Chemical Engineering Science*, 51:931–945, 1996.

[207] N. Poulsen and H. Niemann. Active fault diagnosis based on stochastic tests. *International Journal of Applied Mathematics and Computer Science*, 18(4):487–496, 2008.

[208] J. A. Primbs and C. H. Sung. Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Transactions on Automatic Control*, 54(2):221–230, 2009.

[209] R. Pulch. Polynomial chaos for the computation of failure probabilities in periodic problems. In J. Roos and L. R. J. Costa, editors, *Scientific Computing in Electrical Engineering*, pages 275–284. Springer, Berlin, 2010.

[210] S. J. Qin. An overview of subspace identification. *Computers & Chemical Engineering*, 30:1502–1513, 2006.

[211] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[212] D. M. Raimondo, R. D. Braatz, and J. K. Scott. Active fault diagnosis using moving horizon input design. In *Proceedings of the European Control Conference*, pages 3131–3136, Zurich, Switzerland, July 2013.

[213] R. Ramachandran, J. Arjunan, A. Chaudhury, and M. G. Ierapetritou. Model-based control-loop performance of a continuous direct compaction process. *Journal of Pharmaceutical Innovation*, 6:249–263, 2011.

[214] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation–A moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

[215] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99(3):723–757, 1998.

[216] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design.* Nob Hill Publishing, 2009.

[217] S. Richter, C. N. Jones, and M. Morari. Real-time input-constrained MPC using fast gradient methods. In *Proceedings of the IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference*, pages 7387–7393, Shanghai, China, December 2009.

[218] D. M. Roberge, B. Zimmermann, F. Rainone, M. Gottsponer, M. Eyholzer, and N. Kockmann. Microreactor technology and continuous processes in the fine chemical and pharmaceutical industry: Is the revolution underway? *Organic Process Research & Development*, 12:905–910, 2008.

[219] M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23:470–472, 1952.

[220] S. M. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[221] A. Savchenko, P. Rumschinski, and R. Findeisen. Fault diagnosis for polynomial hybrid systems. In *Proceedings of the IFAC World Congress*, pages 2755–2760, Milano, Italy, August 2011.

[222] A. Savchenko, P. Rumschinski, S. Streif, and R. Findeisen. Complete diagnosability of abrupt faults using set-based sensitivities. In *Proceedings of the IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 860–865, Mexico City, Mexico, August 2012.

[223] S. D. Schaber, D. I. Gerogiorgis, R. Ramachandran, J. M. B. Evans, and P. I. Barton. Economic analysis of integrated continuous and batch pharmaceutical manufacturing: A case study. *Industrial Engineering & Chemistry Research*, 50:10083–10092, 2011.

[224] A. P. Schoen, N. Cordella, S. Mehraeen, M. A. Arunagirinathan, A. J. Spakowitz, and S. C. Heilshorn. Dynamic remodelling of disordered protein aggregates is an alternative pathway to achieve robust self-assembly of nanostructures. *Soft Matter*, 9(38):9137–9145, 2013.

[225] A. Schwarm and M. Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45:1743–1752, 1999.

[226] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43:1163–1169, 1998.

[227] J. K. Scott, R. Findeisen, R. D. Braatz, and D. M. Raimondo. Design of active inputs for set based fault diagnosis. In *Proceedings of the American Control Conference*, pages 3567–3572, Washington, DC, USA, June 2013.

[228] J. K. Scott, G. R. Marseglia, L. Magni, R. D. Braatz, and D. M. Raimondo. A hybrid stochastic-deterministic input design method for active fault diagnosis. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5656 – 5661, Firenze, Italy, December 2013.

[229] D. Semino and A. Brambilla. An efficient structure for parallel cascade control. *Industrial & Engineering Chemistry Research*, 35:1845–1852, 1996.

[230] M. Simandl, I. Puncochár, and J. Královec. Rolling horizon for active fault detection. In *Proceedings of the IEEE Conference on Decision and Control and the European Control Conference*, pages 3789–3794, Seville, Spain, December 2005.

[231] E. Simon, P. R-Ayerbe, C. Stoica, D. Dumur, and V. Wertz. LMIs-based coordinate descent method for solving BMIs in control design. In *Proceedings of the IFAC World Congress on Automatic Control*, pages 10180–10186, Milano, Italy, August 2011.

[232] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* Wiley, New York, 1996.

[233] E. O. P. Solis, P. I. Barton, and G. Stephanopoulos. Controlled formation of nanostructures with desired geometries. 1. Robust static structures. *Industrial & Engineering Chemistry Research*, 49(17):7728–7745, 2010.

[234] E. O. P. Solis, P. I. Barton, and G. Stephanopoulos. Controlled formation of nanostructures with desired geometries. 2. Robust dynamic paths. *Industrial & Engineering Chemistry Research*, 49(17):7746–7757, 2010.

[235] S. Srivastava, A. Santos, K. Critchley, K.-S. Kim, P. Podsiadlo, K. Sun, J. Lee, C. Xu, G. D. Lilly, S. C. Glotzer, and N. A. Kotov. Light-controlled self-assembly of semiconductor nanoparticles into twisted ribbons. *Science*, 327(5971):1355–1359, 2010.

[236] S. Streif, D. Hast, R. D. Braatz, and R. Findeisen. Certifying robustness of separating inputs and outputs in active fault diagnosis for uncertain nonlinear systems. In *Proceedings of the IFAC International Symposium on Dynamics and Control of Process Systems*, pages 837–842, Mumbai, India, December 2013.

[237] S. Streif, M. Karl, and R. Findeisen. Outlier analysis in set-based estimation for nonlinear systems using convex relaxations. In *Proceeings of the European Control Conference*, pages 2921–2926, Zurich, Switzerland, July 2013.

[238] S. Streif, M. Karl, and A. Mesbah. Stochastic nonlinear model predictive control with efficient sample approximation of chance constraints. *arXiv preprint arXiv:1410.4535*, 2014.

[239] S. Streif, F. Petzke, A. Mesbah, R. Findeisen, and R. D. Braatz. Optimal experiment design for probabilistic model discrimination using polynomial chaos. In *Proceedings of the IFAC World Congress*, pages 4103–4109, Cape Town, South Africa, August 2014.

[240] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[241] S. Subramanian, S. Lucia, and S. Engell. Handling structural plant-model mismatch via multi-stage nonlinear model predictive control. In *Proceeings of the European Control Conference*, pages 1602–1607, Linz, Austria, July 2015.

[242] P. Suresh and P. Basu. Improving pharmaceutical product development and manufacturing: Impact on cost of drug development and cost of goods sold of pharmaceuticals. *Journal of Pharmaceutical Innovation*, 3:175–187, 2008.

[243] J. W. Swan, J. L. Bauer, Y. Liu, and E. M. Furst. Directed colloidal self-assembly in toggled magnetic fields. *Soft Matter*, 10(8):1102–1109, 2014.

[244] B. Takács, J. Holaza, M. Kvasnica, and S. Di Cairano. Nearly-optimal simple explicit MPC regulators with recursive feasibility guarantees. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7089–7094. IEEE, Florence, Italy, December 2013.

[245] S. Talreja, P. J. A. Kenis, and C. F. Zukoski. A kinetic model to simulate protein crystal growth in an evaporation-based crystallization platform. *Langmuir*, 23(8):4516–4522, 2007.

[246] S. Talreja, S. L. Perry, S. Guha, V. Bhamidi, C. F. Zukoski, and P. J. A. Kenis. Determination of the phase diagram for soluble and membrane proteins. *The Journal of Physical Chemistry B*, 114(13):4432–4441, 2010.

[247] X. Tang, Y. Xue, and M. A. Grover. Colloidal self-assembly with model predictive control. In *Proceedings of the American Control Conference*, pages 4228–4233, Washington, DC, USA, June 2013.

[248] M. A. Tatang, W. Pan, R. G. Prinn, and G. J. McRae. An efficient method for parametric uncertainty analysis of numerical geophysical models. *Journal of Geophysical Research*, 102(D18):21925–21932, 1997.

[249] J. Tolsma and P. I. Barton. DAEPACK: An open modeling environment for legacy models. *Industrial & Engineering Chemistry Research*, 39:1826–1839, 2000.

[250] Z. W. Ulissi, M. S. Strano, and R. D. Braatz. Control of nano and microchemical systems. *Computers & Chemical Engineering*, 51:149–156, 2013.

[251] F. Ullmann. FiOrdOs: A matlab toolbox for c-code generation for first order methods. Master's thesis, ETH Zurich, 2011.

[252] A. W. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

[253] P. van Overschee and B. L. De Moor. *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Kluwer Academic Publishers, Norwell, Massachusetts, 1996.

[254] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed, linear model predictive control. Part I: State feedback. Technical Report 2006-03, Texas-Wisconsin Modeling and Control Consortium, 2006.

[255] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37:1515–1528, 2000.

[256] R. Vilanova, I. Serra, C. Pedret, and R. Moreno. Reference processing in two-degree-of-freedom control: Separation, independence or optimality. In *Proceedings of the American Control Conference*, pages 5680–5685, Minneapolis, MN, USA, June 2006.

[257] M. P. Vitus and C. J. Tomlin. On feedback design and risk allocation in chance constrained control. In *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, pages 734–739, Orlando, FL, USA, December 2011.

[258] D. E. Wakeham, C.-Y. Chen, B. Greene, P. K. Hwang, and F. M. Brodsky. Clathrin self-assembly involves coordinated weak interactions favorable for cellular regulation. *The EMBO Journal*, 22(19):4980–4990, 2003.

[259] F. Y. Wang, X. Y. Ge, N. Balliu, and I. T. Cameron. Optimal control and operation of drum granulation processes. *Chemical Engineering Science*, 61:257–267, 2006.

[260] L. Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science and Business Media, 2009.

[261] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[262] Y. Wang, Y. Wang, D. R. Breed, V. N. Manoharan, L. Feng, A. D. Hollingsworth, M. Weck, and D. J. Pine. Colloids with valence and specific directional bonding. *Nature*, 491(7422):51–55, 2012.

[263] S. Whitelam. Control of pathways and yields of protein crystallization through the interplay of nonspecific and specific attractions. *Physical Review Letters*, 105(8):088102, 2010.

[264] G. M. Whitesides and M. Boncheva. Beyond molecules: Self-assembly of mesoscopic and macroscopic components. *Proceedings of the National Academy of Sciences*, 99(8):4769–4774, 2002.

[265] G. M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[266] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60:897–936, 1938.

[267] S. J. Wright. Interior point methods for optimal control of discrete time systems. *Journal of Optimization Theory and Applications*, 77(1):161–187, 1993.

[268] S. J. Wright. Applying new optimization algorithms to model predictive control. In *AIChE Symposium Series*, volume 93, pages 147–155, 1997.

[269] S. J. Wright. *Primal-dual interior-point methods*. SIAM Publications, 1997.

[270] H. Wu, M.r A. Khan, and A. S. Hussain. Process control perspective for process analytical technology: Integration of chemical engineering practice into semiconductor and pharmaceutical industries. *Chemical Engineering Communications*, 194:760–779, 2007.

[271] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

[272] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24:619–644, 2002.

[273] Y. Xue and M. A. Grover. Optimal design for active self-assembly system. In *Proceedings of the American Control Conference*, pages 3269–3274, San Francisco, CA, USA, June 2011.

[274] D. Youla, J. Bongiorno, and H. Jabr. Modern Wiener-Hopf design of optimal controllers - Part II: The multivariable case. *IEEE Transactions on Automatic Control*, AC-21:319–338, 1976.

[275] C.-C. Yu. Design of parallel cascade control for disturbance rejection. *AIChE Journal*, 34:1833–1838, 1988.

[276] L. X. Yu. Pharmaceutical quality by design: Product and process development, understanding, and control. *Pharmaceutical Research*, 25:781–791, 2008.

[277] M. N. Zeilinger, C. N. Jones, D. M. Raimondo, and M. Morari. Real-time MPC-stability through robust MPC design. In *Proceedings of the IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference*, pages 3980–3986, Shanghai, China, December 2009.

[278] X. Zhang, K. Margellos, P. Goulart, and J. Lygeros. Stochastic model predictive control using a combination of randomized and robust optimization. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7740–7745, Florence, Italy, December 2013.

[279] X. J. Zhang. *Auxiliary Signal Design in Fault Detection and Diagnosis.* Lecture Notes in Control and Information Sciences. Springer, Heidelberg, Berlin, 1989.

[280] Y. Zhang and N. V. Sahinidis. Uncertainty quantification in CO_2 sequestration using surrogate models from polynomial chaos expansion. *Industrial & Engineering Chemistry Research*, 52:3121–3132, 2012.

[281] D. H. Zhou and P. M. Frank. Fault diagnostics and fault tolerant control. *IEEE Transactions on Aerospace and Electronic Systems*, 34:420–427, 1998.

[282] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, New Jersey, 1996.

[283] K. Zhou and Z. Ren. A new controller architecture for high performance, robust, and fault-tolerant control. *IEEE Transactions on Automatic Control*, 46:1613–1618, 2001.

[284] R. Zwanzig. Nonlinear generalized Langevin equations. *Journal of Statistical Physics*, 9(3):215–220, 1973.