Probabilistic and Learning Approaches through Concurrent Parameter Estimation and Adaptive Control for In-Space Robotic Assembly

by

Antonio Terán Espinoza

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Signature redacted

Author ....
Department of Aeronautics and Astronautics
May 25, 2017

Signature redacted

Certified by...... ....
Alvar Saenz-Otero
Director, Space Systems Laboratory
Thesis Supervisor

Signature redacted

Accepted by ....
Youssef M. Marzouk
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Probabilistic and Learning Approaches through Concurrent Parameter Estimation and Adaptive Control for In-Space Robotic Assembly

by

Antonio Terán Espinoza

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Autonomous and multi-agent space operations within the context of in-space robotic servicing, assembly, and debris removal have received particular attention from both research and industry communities. The presence of uncertainties and unknown system parameters amongst these missions is prevalent, as they primarily deal with unknown or uncooperative target objects, e.g., asteroids or unresponsive, unsupervised tumbling spacecraft. To lower the inherent risk associated with these types of operations, possessing an accurate knowledge of the aforementioned characteristics is essential. In order to achieve this, approaches that employ a unified framework between parameter estimation and learning methodologies through a Composite Adaptation (CA) structure are presented. Furthermore, to evaluate the likelihood of mission success or objective completion, a probabilistic approach upon the system's operations is introduced; by employing probability distributions to model the control system's response and pairing these with the analysis of objectives' requirements and agents' characteristics, the calculation of on-board feasibility and performance assessments is presented. A formulation for the estimator and the controllers is developed, and results for the adaptive approach are demonstrated through hardware implementation using MIT's Synchronized Position Hold Engage Reorient Experimental Satellites (SPHERES) ground testing facilities. On-orbit test session data is analyzed, and further improvements upon the initial learning approach are verified through simulations.

Thesis Supervisor: Alvar Saenz-Otero
Title: Director, Space Systems Laboratory

Acknowledgments

First of all, I would like to express my gratitude to Dr. Alvar Saenz-Otero and Professor David Miller for the opportunity of allowing me to join the MIT Space Systems Laboratory team, and for their mentorship throughout these two years (and hopefully a couple more).

I am also eternally grateful to the rest of the SPHERES team for embracing me as their friend and for the endless advices and support that they have given me, specially Chris, David, and Tim. Special thanks to Dr. Danilo Roascio; words truly fall short whenever I try to explain how many times you have saved me from catastrophic situations. If it were not for you, I would still be trying to install the SPHERES sim.

To my friend, office mate, and the best 2nd Lt. out there, Zachary Funke, with whom it was an honor to have spent countless hours discussing everything from seapunk to quantum mechanics, thank you for all the good times. And to my office mate William David Sanchez, the coolest home-schooled kid out there, thank you for keeping my feet on the ground, for all those reality checks, for the invaluable bodybuilding and nutrition advices, and most importantly, for your unlimited patience; I promise you will not leave MIT without knowing how to ride a bike while standing up.

Lastly, even though it goes without saying, I would like to officially thank my family for always being by my side and unquestionably supporting my every endeavor. And saving the very best for the very last, an infinite number of thanks to my partner in crime, confidant, and ever-loving better half Olivia Claire Wolcott Kern for everything; it has been a wonderful journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, as the objectives of future space missions become more ambitious, the area of space robotics is gradually starting to play a more prevalent design and operational role [1]. More specifically, autonomous and multi-agent space operations within the context of in-space robotic servicing, assembly, and debris removal have received particular attention from both research and industry communities, yielding the advances within the field to be a paramount component of the technology in charge of accomplishing such tasks [2, 3].

A constant theme amongst these types of missions is the prevalence of uncertainties and unknown system parameters [4, 5]; this can be seen by the fact of having to deal with unknown or uncooperative target objects — e.g., asteroids or unresponsive, unsupervised tumbling spacecraft — or the propagated parameter uncertainties arising from docking and aggregative maneuvers — strictly increasing variance of model's parameters after an assembly step. In order to guarantee a certain degree of mission success and to be able to lower the inherent risk associated with these operations, possessing an accurate knowledge of these aforementioned characteristics is essential [6, 7].

Depending on the specified mission objective, different approaches for dealing with uncertainties can be taken. In the case in which an accurate knowledge of the parameters describing the system is of particular interest, then system identification (SysId) techniques can be employed to fully characterize the specifications of the model [8, 9]; an example scenario might include two cooperative satellites, one of which performs SysId maneuvers to corroborate (or in the case of a broken fuel gauge, to fully determine) its current mass and inform the servicer the amount of fuel it needs.

On the other hand, if these properties are of no particular use to the spacecraft's autonomy cycle and the only objective is to accurately and effectively control the system's state, then on-board adaptation methods for learning under a strictly need-to-know basis are adequate [10]; this means that once the system reaches the desired tracking state, e.g., negligible tracking errors, the need for performing the adaptation is no longer warranted, entailing that the attainment of the true parameter values for the estimated quantities in not strictly necessary (e.g., if the mass parameter was not fully characterized, but the control performance is more than acceptable, that incomplete mass estimate will be kept and used until further adaptation is warranted).

A common approach for performing system identification through parameter estimation actions is to employ an output prediction error obtained by comparing the sensed system's current output to an analytical prediction of the value at the current time step [11]. Contrastingly, adaptive approaches employ the information granted by the system controller's tracking error [12], which is given by the difference between a desired reference trajectory and the current state of the system.

The purpose of this work is to further investigate a unifying approach between the two frameworks through a composite adaptation structure in order to assess the stability and performance of extracting information from both sources — the tracking error and the output prediction error — and concurrently performing adaptation and estimation techniques while employing probabilistic models to perform on-board assessments on the system's performance.

In order to develop such algorithms, not only must the theoretical framework and formulation be elaborated and proven, but its reliability through rigorous trials and testing needs to be demonstrated. Since the true spacecraft operating environment is not easily accessible (outer space!), the design and development cycle results will mostly depend on the authenticity level of the simulation and testbeds employed. Fortunately for the case of autonomy and control algorithms involving spacecraft, the MIT Space Systems Laboratory SPHERES testbed provides a genuine 6DOF microgravity environment for development and maturation practices [13].

1.2 SPHERES Facility

The MIT SSL's Synchronized Position Hold Engage Reorient Experimental Satellites (SPHERES) are a self-contained testbed for control, estimation, autonomy and artificial intelligence algorithms available aboard the International Space Station (ISS) for a full 6DOF true microgravity environment, paired with a ground testing segment that allows for 3DOF testing using an air-bearing system [14].

Each of these SPHERES satellites possesses an expansion port that enables the attachment of external peripherals such as stereo-vision cameras, docking ports, additional avionics stacks, and robotic arms. In order to further research the field of in-space robotic assembly, the Halo assembly — a peripheral device that acts as an augmented expansion port for SPHERES, enabling the simultaneous use of up to six peripheral attachments — was designed, built, and sent to station under DARPA's InSPIRE-II program [15]. A time lapse of the build process for the Halo flight units is shown in Fig. 1-1; the actual flight hardware sent to the ISS can be seen in Fig. 1-2.

The addition of this new Halo hardware, paired with the available SPHERES suite on board the ISS, allows for the recreation of distinct modular parts of an in-space robotic assembly final ensemble; as shown in Fig. 1-3, the configuration space spanned by all of the elements of the SPHERES framework and the unique microgravity environment it offers, enables not only the design and implementation of in-space robotic assembly maneuvers, but also an incremental and iterative algorithm and hardware developing process to further mature this technology.

An example of an achievable scenario using the SPHERES platform would include an inspector satellite outfitted with a Halo, stereo-vision camera, and docking ports

Figure 1-1: Time lapse of the Halo Flight Hardware build, featuring 2nd Lt. Zachary Funke, Roedolph Opperman, and Dr. Danilo "***the wizard***" Roascio. (Video speed: $\times 10^6$)

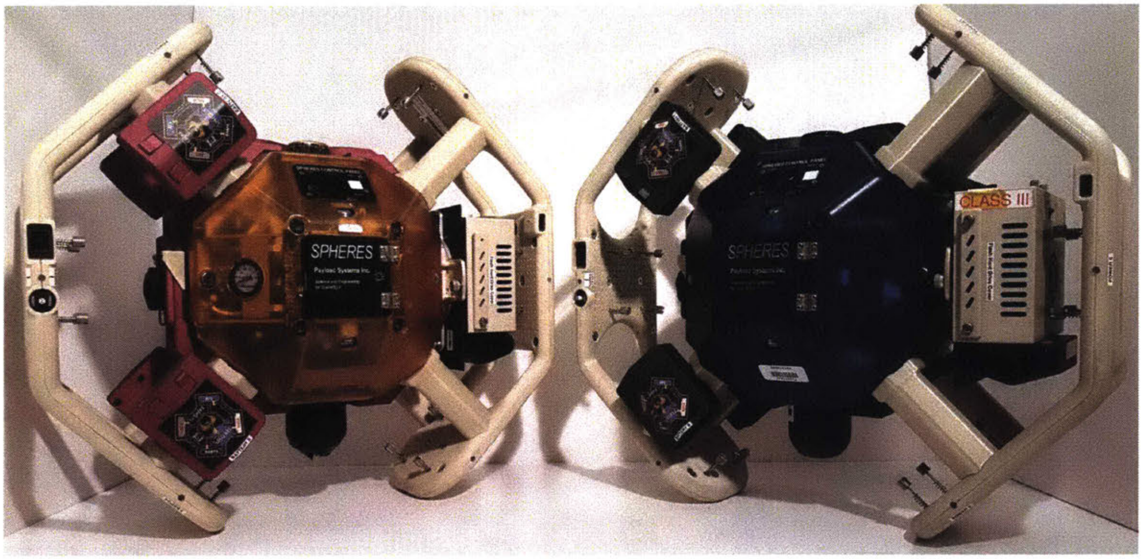


Figure 1-2: The two flight-certified Halos that were sent to the International Space Station on April 8, 2016 with SpaceX CRS8.

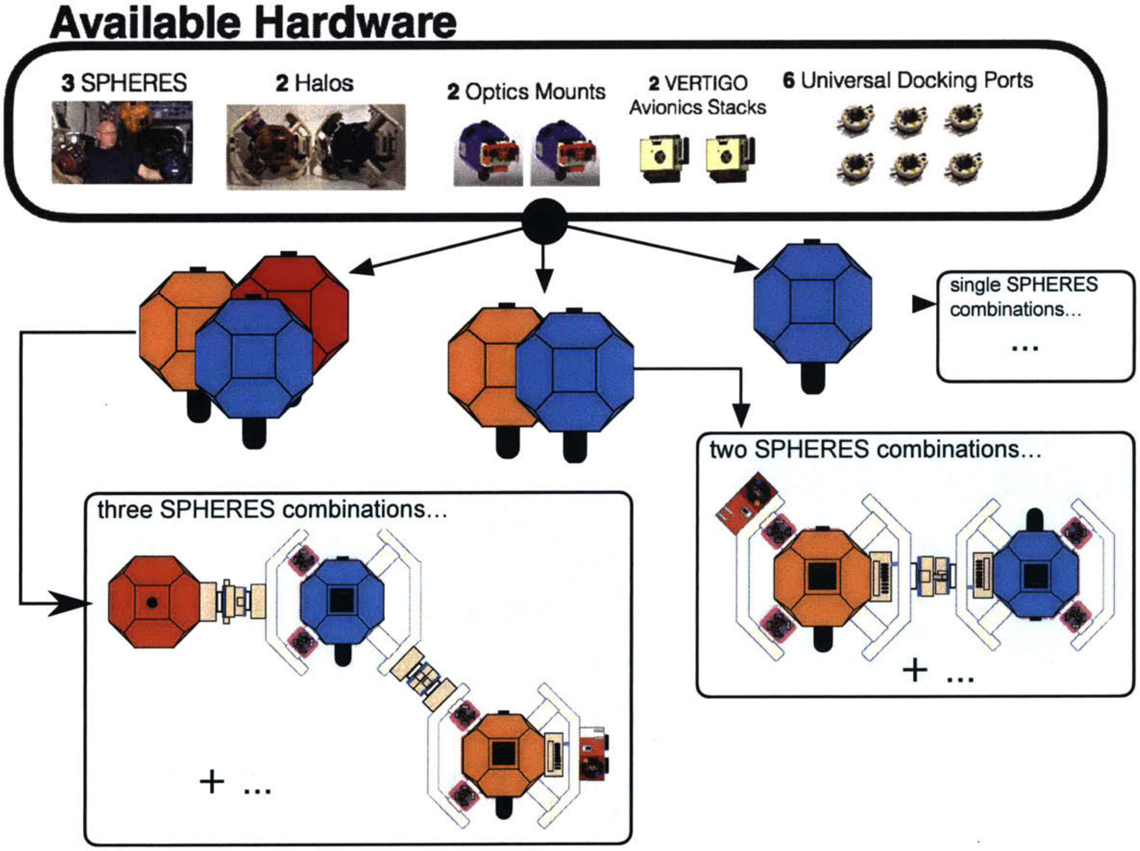

Available Hardware
3 SPHERES
2 Halos
2 Optics Mounts
2 VERTIGO Avionics Stacks
6 Universal Docking Ports
single SPHERES combinations...
...
two SPHERES combinations...
+ ...
three SPHERES combinations...
+ ...

Figure 1-3: SPHERES Suite: hardware attachments currently available on-board the International Space Station. It is easy to see the many possible combinations available through the use of multiple peripherals.

to use vision-based navigation to autonomously and rigidly dock to a second satellite equipped with its own Halo and sets of docking ports, shown in Fig. 1-4. This assembly would then cooperatively aggregate its control resources, hold its position (or reconfigure), and wait for a third satellite to dock to a port. A translation and slew maneuver can then be carried out, followed by a reconfiguration maneuver in which the inspector satellite undocks, circumnavigates the full ensemble, assesses it, and docks to a different port of the expansion device.

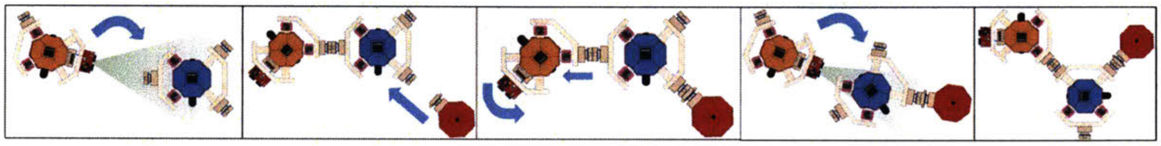


Figure 1-4: Sample sequence of actions for recreating an in-space robotic assembly scenario with the SPHERES hardware.

Furthermore, the aforementioned uncertainties governing these types of operation regimes can be easily identified; since it is intractable to perform system identification tests for all the possible hardware combinations available, and due to the fact that the physical properties of the system are intrinsically tied to the control system, a learning approach to determine these during real-time operations is warranted.

1.3 Research Objective

The problem of in-space robotic servicing and assembly is a multidisciplinary and multifaceted endeavor; in order for it to be solved, issues that arise in many fields need to be tackled, such as communication protocols, thermal problems, mechanical coupling interfaces, control systems, among very many others. The work being done here is concerned with the control, estimation and autonomy aspect of these types of missions.

The objective of this thesis is summarized as follows:

- **To** contribute towards achieving reliable in-space robotic assembly control systems

- **By** leveraging adaptive control and estimation theory to enable on-board real time learning methods and employing probabilistic approaches to reduce the mission's complexity and avoid unwarranted unfeasibility claims.

- **Using** both the controller's tracking error and the estimator's output prediction error as a joint information source from which to perform parameter adaptation corrections

- **While** preserving global asymptotic stability and ensuring tracking error convergence.

1.4 Literature Review

Although much has been done for on-orbit servicing and in-space robotic assembly missions, to ascertain reliable autonomous space operations, further development of research areas such as intelligent controls, communication technologies, kinematic and dynamic model uncertainties, on-orbit model parameters identification, motion estimation and prediction of target objects, amongst many others, is warranted [16].

1.4.1 Spacecraft System Identification Methods

In terms of on-orbit parameter identification techniques, the most common framework employed for computing estimates using the so-called innovation term, or output-prediction error, is the Kalman Framework [17]. Simplified approaches that does not utilize the full Kalman formulation have also been applied, such as the recursive least squares formulation and tracking differentiator employed by Xu [18]; in his work, a comparison between the simplified approach and an Extended Kalman Filter (EKF) shows that the latter provides more robustness to the estimation scheme, but adds overhead and complexity to the algorithm by including extra tuning parameters such as discretization step size, noise matrices and variance values, initial conditions, and time synchronization. Furthermore, Kim [19] pairs a regressor matrix implementation of the linear least squares algorithm with an additional Savitzky-Golay filtering step to smooth out noise and differentiation disturbances, and is able to estimate the constant inertia parameters of a satellite through numerical simulations.

Similarly, through an Observer/Kalman filter identification method, Li [20] presents a technique for estimating the system's unknown parameters through impulse actuations and induced excitations while analyzing the displacement and mode order of large flexible appendages on satellites; results of identification tests on a single spacecraft were shown through simulation, but the presented methods are not transferable to or appropriate for small, reconfigurable spacecraft. Additionally, with fuel being a scant resource for space systems, techniques that estimate at the expense of fuel consumption are not always available.

The capabilities of the Kalman framework have recently been exploited to the fact of being able to achieve simultaneous identification and calibration of sensor, as Kiani [21] shows by introducing a Marginal Modified Unscented Kalman Filter (UMMUKF) for system identification of a spacecraft's inertia tensor and the calibration of its magnetometer's parameters. Through modifications to the sigma point sampling scheme, an increase in accuracy while reducing the computation costs with respect to the vanilla UKF was demonstrated through Monte Carlo simulations. As with the previously mentioned works, no hardware validation was acquired.

Although the aforementioned framework clearly provides useful techniques, Chandler [22], while similarly elucidating the importance of on-line system identification methods for time-varying and reconfigurable systems, touches upon the endemic divergence problems presented by the utilization of a standalone Kalman framework. He therefore introduces a static approach that includes the formulation of a static regressor, but still heavily relies on an accurate and detailed model of the system, as well as in accurate parameter priors.

Another aspect worth mentioning is that the space robotics field has been mostly impervious to the dramatic impact that the nonparametric estimation world has had upon ground robotics [23, 24]; while the state-of-the-art learning and adaptive techniques employed by the robotics community rely nowadays in Bayesian nonparametrics [25], probabilistic inference through graphical models [26], information theory [27], etc. [28], such techniques have not been widely or commonly explored within the spacecraft systems, as far as the author concerns. This leaves much room for further research into the on-orbit parameter identification methods, specially for in-space robotic assembly.

1.4.2 Adaptive Control for Spacecraft Operations

Within the control theory realm, adaptive control methods have been largely studied; excellent references such as Ioannou's [29] textbook on the principles of adaptive control theory, as well as the variants that Tao [30] introduces in its analyses and designs of adaptive control systems cover a broad spectrum on the characteristics of the approach. Nevertheless, these techniques have been shown to be quite intricate, relying on complex matrix theory factorizations [31, 32, 33] for system decoupling effects that are not directly translatable into simple real-time control algorithms to be discretely implemented into a spacecraft's control subsystem.

Discrete-time formulations of adaptive controllers have been recently formulated by Li [34], in which output sensor uncertainties are taken into account, and generalized for cases in which the system's dynamical model is unknown. Similarly, Maiti [35] show a multivariable formulation of a similar problem, in which validation results through simulated environments are presented. These types of approaches deal exclusively with linearized dynamical models, and formulate the problem in the Laplace domain to determine the convergence properties and guarantees of the algorithms.

A different take on adaptive controllers formulation is introduced by Slotine [10], in which an Invariant Set-like solution for non-autonomous systems is generated to formulate control and adaptation laws, while analyzing the proposed solution's stability and guarantees through continuous-time domain formulations. Some of these ideas are expanded by Yucelen [36] in order to accommodate stochastic operations, but still preserving a linear system.

The exploration of the aforementioned techniques withing the spacecraft operations world is scant given that the current space missions can be satisfactorily achieved by means of more traditional control systems; nonetheless, the majority of efforts lie in the adaptive control of space manipulators. By analyzing unknown tumbling targets, Nguyen-Huynh [37, 38] proposes a scheme based on the momentum conservation of the system that allows for the new ensemble's inertia parameters to be adaptively modified. Simulation examples are presented, but no converge proofs or stability guarantees are offered.

In the realm of satellite control, only a few avenues have been explored. Ulrich [39] formulates a passivity-based approach for a Simple Adaptive Control (SAC) technique, in which no model parameters are adapted, and the time varying gains are directly fed into the control law. ClohessyWiltshire relative dynamics are taken into account, and coarse test scenarios involving flyby maneuvers are demonstrated; experimental results using MIT's SPHERES 3DOF testbed were obtained for a single simple step response test, in which the position response was analyzed without taking into account any attitude information or actuation. Although guarantees are given by the passivity assumption, no stability analysis is offered for the stochastic case.

The same ClohessyWiltshire dynamics problem is tackled by Singla [40], in which an output feedback strategy is used to provide stability proofs and convergence properties, under the assumptions of a known upper bound in the system's properties and a bounded measurement signal provided by the sensors, modeled under Gaussian noise assumptions. Coarse tracking maneuvers for relative motions are analyzed through simulations.

Contrasting with the two former approaches, Zhang [41] addresses the nonlinear equations of motion for rendezvous maneuvers between two spacecraft using an adaptive Lypaunov-based controller. Even though only theoretical predictions are posed as a function of the obtained numerical results, insights on optimal gain selection strategies for the controller's tuning factors are presented.

1.5 Thesis Roadmap

This thesis consists of five chapters. Chapter 1 offer a motivation for in-space robotic assembly, a literature review of previous work on learning techniques and adaptive control for reconfigurable spacecraft, and formulates the research objective of this thesis. Chapter 2 formalized the mathematical framework of the problem, introduces the current control methodologies employed by the SPHERES testbed, and presents the derivation and formulation of the proposed adaptive controller. Chapter 3 analyzes on-orbit SPHERES test session data, and elucidates the problem of dealing with large system uncertainties; a comparison between traditional control methodologies and adaptive approaches is shown using hardware implementation data. Chapter 4 presents a probabilistic approach which builds upon the data obtained through the testing shown in chapter 3, exhibits the limitations presented by a standalone adaptive controller while operating in stochastic environments, and introduces an algorithm that concurrently and jointly uses an adaptive controller and a parameter identifying estimator to overcome the problems obtained during the stochastic MRAC operation. Chapter 5 summarizes the findings of this work, and presents future work areas and suggestions.

Chapter 2

Theoretical Framework

To formalize the problematics of space robotic assembly, the mathematical framework is laid through a state space formulation of the problem. The different sources of external disturbances arising from operating within a stochastic environment are introduced, and the uncertainties within the system's parameters denoted.

A quick recap on the form of the current control techniques employed within the SPHERES framework for position and attitude regulation is performed. Subsequently, an adaptive control technique called Model Reference Adaptive Controller (MRAC) is introduced, and the derivation of a position and attitude adaptive controller carried out.

2.1 Problem Formulation

Given the fact that the problem to be dealt with is concerned with the dynamics of rigid bodies, both the position and orientation, as well as its respective rates of change, are of interest. For convention, simplicity, and practicality, a state space formulation of the scenario is carried out. Of particular importance are the individual states — the set of which is called a state vector — representing the overall condition of the system, which in this case are chosen as:

$$\mathbf{x}(t) = \left[\underbrace{x(t)\ \ y(t)\ \ z(t)}_{\text{position}}\ \ \overbrace{\dot{x}(t)\ \ \dot{y}(t)\ \ \dot{z}(t)}^{\text{linear velocities}}\ \ \underbrace{\phi(t)\ \ \theta(t)\ \ \psi(t)}_{\text{orientation}}\ \ \overbrace{\dot{\phi}(t)\ \ \dot{\theta}(t)\ \ \dot{\psi}(t)}^{\text{angular velocities}} \right]^T, \quad (2.1)$$

with

$$\begin{aligned}
\mathbf{x}_p(t) &= [\, x(t)\ \ y(t)\ \ z(t)\,]^T \in \mathbb{R}^{1\times 3} \ : \ \text{position of spacecraft's center of geometry} \\
\dot{\mathbf{x}}_p(t) &= [\, \dot{x}(t)\ \ \dot{y}(t)\ \ \dot{z}(t)\,]^T \in \mathbb{R}^{1\times 3} \ : \ \text{linear velocity of spacecraft's center of geometry} \\
\boldsymbol{\theta}_o(t) &= [\, \phi(t)\ \ \theta(t)\ \ \psi(t)\,]^T \in \mathbb{R}^{1\times 3} \ : \ \text{orientation of spacecraft} \\
\dot{\boldsymbol{\theta}}_o(t) &= \left[\, \dot{\phi}(t)\ \ \dot{\theta}(t)\ \ \dot{\psi}(t)\, \right]^T \in \mathbb{R}^{1\times 3} \ : \ \text{angular velocities of spacecraft,}
\end{aligned}$$

where ϕ(t) represents the satellite's roll angle at time t, θ(t) its pitch angle, and ψ(t) its yaw angle. In addition to these elements, internal system disturbances such as a process model noise arising from actuator uncertainties and measurement noises driven by the sensors' sensitivity is taken into account.

For a free floating spacecraft situated at a location with an absence of a gravity well, an abstracted system governed by the differential equation

$$\begin{gathered}
\mathbf{F}(t) = m \cdot \mathbf{a}(t) \\
[\, F_x(t)\ \ F_y(t)\ \ F_z(t)\,]^T = m \cdot [\, \ddot{x}(t)\ \ \ddot{y}(t)\ \ \ddot{z}(t)\,]^T,
\end{gathered} \quad (2.2)$$

is obtained, where $m \in \mathbb{R}^+$ represents the unknown ensemble's mass, and $\mathbf{a}(t) = \frac{d}{dt}\left(\dot{\mathbf{x}}_p(t)\right) \in \mathbb{R}^{1\times 3}$ the time derivative of the spacecraft's linear velocity.

By choosing the first half of the aforementioned state vector and denoting it as $\mathbf{X}(t) = [\, \mathbf{x}_p(t)\ \ \dot{\mathbf{x}}_p(t)\,]^T$, the system dynamics can be expressed as

$$\begin{aligned}
\dot{\mathbf{X}}(t) &= [\, \dot{\mathbf{x}}_p(t)\ \ \mathbf{a}(t)\,]^T \\
&= [\, \dot{x}(t)\ \ \dot{y}(t)\ \ \dot{z}(t)\ \ \ddot{x}(t)\ \ \ddot{y}(t)\ \ \ddot{z}(t)\,]^T \\
&= \left[\, \dot{x}(t)\ \ \dot{y}(t)\ \ \dot{z}(t)\ \ \frac{m}{F_x(t)}\ \ \frac{m}{F_y(t)}\ \ \frac{m}{F_z(t)}\, \right]^T,
\end{aligned} \quad (2.3)$$

which can then be further expressed as a linear time-invariant (LTI) system of the form

$$\begin{aligned} \dot{\mathbf{X}}(t) &= A \cdot \mathbf{X}(t) + B \cdot \mathbf{u}(t), \\ \mathbf{u}(t) \in \mathbb{R}^{1 \times \bar{m}} &: \text{ system's control inputs}, \\ A \in \mathbb{R}^{\bar{n} \times \bar{n}} &: \text{ system matrix}, \\ B \in \mathbb{R}^{\bar{m} \times \bar{n}} &: \text{ output matrix}, \end{aligned}$$

where $\bar{n} \in \mathbb{N}$ is the cardinality of the state vector, and $\bar{m} \in \mathbb{N}$ is the number of available inputs to the system. This formulation is employed to ease the manipulation of the multiple input multiple output systems presented in the adaptive control section.

Expressing 2.2 using this form yields the following LTI system:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{:=A} \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix}}_{:=B} \underbrace{\begin{bmatrix} F_x(t) \\ F_y(t) \\ F_z(t) \end{bmatrix}}_{:=\mathbf{u}(t)} . \tag{2.4}$$

In terms of the output of the system, the satellite is taken to be operating under the assumption that a positioning system is being made available (e.g., by surrounding larger spacecraft like GPS), and that the craft possesses gyroscopes and accelerometers to sense its angular rate and acceleration. Thus, an output equation of the form

$$\begin{aligned} \mathbf{Y}(t) &= C \cdot \mathbf{X}(t) + D \cdot \mathbf{u}(t), \quad \text{with} \\ C &= \mathbf{I}^{\bar{n} \times \bar{n}}, \\ D &= \mathbf{0}^{\bar{n} \times \bar{m}} \end{aligned} \tag{2.5}$$

is obtained. This implies both a strictly proper ($D = \mathbf{0}$) and fully observable ($\mathbf{I}^{\bar{n}\times\bar{n}}$ spans $\mathbb{R}^{\bar{n}\times\bar{n}}$) system. The latter can be proved by determining the rank of the system's observability matrix

$$\mathcal{M}_o = \left[\, C^T \,\vert\, (CA)^T \,\vert\, \cdots \,\vert\, (CA^{\bar{n}-2})^T \,\vert\, (CA^{\bar{n}-1})^T \,\right] \tag{2.6}$$

by substituting in the values of the matrices, and checking whether the entire state space is spanned (no unobservable states) by obtaining a full rank matrix, i.e.,

$$\operatorname{rank}\mathcal{M}_o = \operatorname{rank}\left(\left[\, \mathbf{I}^T \,\vert\, (\mathbf{I}A)^T \,\vert\, \cdots \,\vert\, (\mathbf{I}A^5)^T\right]^T\right) = 6,$$

which is equal to the number of states. It can easily be seen that if the system were to lose any of its position measurements, the observability condition would be lost, given that $\operatorname{rank}(\mathcal{M}_o) \leq 5 < n$. Thus the assumption that no measurements are lost is also made, given that a lack of these would force the controller to completely rely solely on the process model estimate (no innovation), which could potentially be problematic for stability conditions in high uncertainty environments.

2.1.1 Process Noise Characteristics

Knowing that most undoubtedly any real-life system subject to conditions similar to the ones described above will behave in a non-deterministic manner, a way to account for this intrinsic stochasticity is needed. To do this, a zero-mean continuous-time additive Gaussian white noise $\mathbf{w}(t)$ model will be considered which, as explained in [17], can be expressed as

$$\begin{aligned} \mathbb{E}\left[\mathbf{w}(t)\mathbf{w}^T(\tau)\right] &= Q_c\delta(t-\tau), \text{ with} \\ Q_c = Q/T_s &: \text{ autocorrelation of continuous-time noise,} \\ Q &: \text{ equivalent discrete-time noise covariance,} \\ T_s &: \text{ sample period of this latter system,} \\ \delta(t-\tau) &: \text{ continuous-time impulse response, and} \\ Q_c\delta(t) &: \text{ covariance matrix of continuous-time noise,} \end{aligned} \tag{2.7}$$

yielding a stochastic system driven by noise and disturbances, expressed by the equation

$$\dot{\mathbf{X}}(t) = A \cdot \mathbf{X}(t) + B \cdot \mathbf{u}(t) + B_w \cdot \mathbf{w}(t).$$

2.1.2 Measurement Noise Characteristics

To sense their state and the environment around them, spacecraft are equipped with an array of sensors spanning from star trackers to inertial measurement units (IMUs). Since full state sensing abilities are being considered, to simulate the data acquisition, a measurement model with linear properties and additive white Gaussian noise is utilized, with the form

$$\mathbf{y}_k = \mathbf{y}(k) = H \cdot \mathbf{X}(k) + \mathbf{v}_k, \tag{2.8}$$

where the linear matrix $H \in \mathbb{R}^{\bar{n}\times\bar{n}}$ is the observation operator, $\bar{n}$ is the state dimension, and $\mathbf{v}_k$ is a zero-mean white Gaussian noise vector

$$\mathbf{v}_k = \begin{bmatrix} v_{x_k} & v_{y_k} & v_{z_k} & v_{\dot{x}_k} & v_{\dot{y}_k} & v_{\dot{z}_k} \end{bmatrix}^T,$$

with discrete-time covariance matrix

$$\mathbf{R} = \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{v_z}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_{\dot{x}}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_{\dot{y}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_{\dot{z}}}^2 \end{bmatrix},$$

where $\sigma_{v_\alpha}^2$ denotes the variance of the uncertainty distribution over the state $\alpha \in \{\mathbf{X}(t)\}$. The measurement noise model can be succinctly described as

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}). \tag{2.9}$$

Constant additive white Gaussian noise might not be the ideal noise model for spacecraft sensors, given the extreme conditions to which they are subject; their environment can radically change from orbit to orbit, going from over 100^{o}C to under -30^{o}C in a matter of minutes [42]. This effect can impact a spacecraft's sensors [9, 43], and a more thorough evaluation of the problem might greatly benefit from including this conditions into the formulation, as well as delving into noise adaptation theory, activities which will be added to the future work list.

2.2 Current Control Methodologies for SPHERES

The control techniques used for full 3D pose control of the SPHERES satellites rely on linear and nonlinear PD and PID feedback strategies. These controllers are completely agnostic to the system's underlying model and properties, and exclusively utilize the tracking error dynamics to compute a control input driven by the tuned value of the controllers' weights [44].

2.2.1 Position Controller

Being one of the most utilized feedback strategies throughout industry [45], a Proportional-Derivative-Integral control law is utilized to regulate the SPHERES position states. The control input fed to the plant is calculated using the expression

$$\begin{aligned}\mathbf{u}_p(t) &= k_p \cdot \mathbf{e}_p(t) + k_i \cdot \int_0^t \mathbf{e}_p(\tau)\, d\tau + k_d \cdot \frac{d}{dt}\left(\mathbf{e}_p(t)\right) \\ &= k_p \cdot \left(\mathbf{e}_p(t) + \frac{1}{T_i} \cdot \int_0^t \mathbf{e}_p(\tau)\, d\tau + T_d \cdot \frac{d}{dt}\left(\mathbf{e}_p(t)\right)\right),\end{aligned} \tag{2.10}$$

with

$$\mathbf{e}_p(t) \equiv \mathbf{x}_p(t) - \mathbf{x}_d(t),$$

where $\mathbf{x}_d(t)$ represents the desired reference state.

By assuming decoupled translation movement, the controller is tuned by taking as reference a second order system and defining a desired time response; this time response is described by a natural frequency ω_n and a damping ratio ζ, both of which are user defined parameters. Tuning PID controllers is not particularly intuitive and is commonly performed through frequency domain analysis.

2.2.2 Attitude Controller

The orientation of the satellite within the SPHERES state vector is represented through a quaternion formulation, which allows the system to overcome issues such as representation ambiguities and gimbal lock situations [46]. Quaternions are a nonlinear attitude representation, therefore yielding a nonlinear control feedback strategy when calculating the errors using said representation.

By utilizing the angular rates as well as the attitude quaternion error, which is computed by establishing a desired orientation and expressing it through a quaternion, the Proportional-Derivative control law [47] described by equation

$$\mathbf{u}_o(t) = \mathbf{K} \cdot \mathbf{q}_e(t) - \mathbf{C} \cdot \dot{\boldsymbol{\theta}}_o(t) \tag{2.11}$$

is obtained, with $\mathbf{q}_e(t)$ denoting the quaternion error. In the 6DOF case, $\mathbf{K} \in \mathcal{R}^{3\times4}$ and $\mathbf{C} \in \mathcal{R}^{3\times3}$ represent the proportional and derivative gains, respectively. Asymptotic global stability is guaranteed if and only if one of the following conditions are met [48]:

$$\mathbf{C} = \text{diag}(c_1, c_2, c_3), c_i \in \mathcal{R}^+, \quad \text{and} \quad \left[\mathbf{K} = k\mathbf{I} \ \text{ or } \ \mathbf{K} = \frac{k}{1_4^3}\mathbf{I} \ \text{ or } \ \mathbf{K} = k\text{sgn}(q_4)\mathbf{I}, k \in \mathcal{R}^+ \right].$$

2.3 Adaptive Controller Formulation

It has been shown so far how susceptible traditional controllers for spacecraft are to model uncertainties and unknown parameters [7]. In order to account for these problems, the derivation of adaptive techniques is carried out throughout this section.

2.3.1 Direct and Indirect Adaptation

Within the adaptive control realm there exist two main types of paradigms: direct adaptive controllers, and indirect adaptive controllers. The difference between the two is quite simple, but grants them two very different types of characteristics with its respective advantages and disadvantages.

Whenever the controller's parameters are a function of the plant's parameters, the on-line parameter adaptation techniques are first concerned with obtaining estimates for the latter values (e.g., mass, inertia, etc) and subsequently using these guesses to compute the corresponding control law [29]. This approach is known as indirect or explicit adaptive control and its block diagram representation is shown in Fig. 2-1.

Conversely, whenever the plant's model is parameterized in such a way that the estimated parameters are the controller's parameters directly, the approach is called direct or implicit adaptive control, given that the plant's model remains implicit [29]; the block diagram corresponding to this approach is shown in Fig. 2-2. Examples of

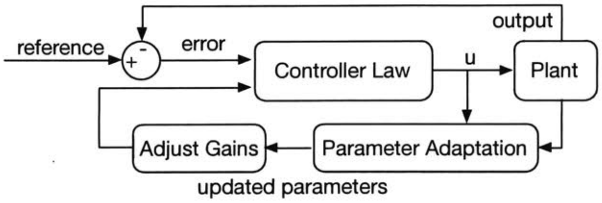


Figure 2-1: Indirect adaptive control block diagram.

indirect adaptive controllers are self-tuning PID controllers, or controllers in which the parameters to be calculated are the coefficients of the polynomials found within a system's transfer function [49].

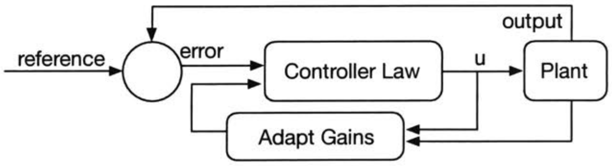


Figure 2-2: Direct adaptive control block diagram.

For the in-space robotic assembly problem, it is of much interest to be able to iteratively and recursively learn and adapt to the system's parameters on-line. Determining in real time physical properties such as mass, center of mass, products and moments of inertia, amongst others, would not only allow for a more accurate model of the system, but also to draw inferences about the status of the spacecraft at a certain time. It is the fact of having a well-studied model describing the dynamical interactions within the satellite's environment, paired with these aforementioned reasons, that makes an indirect adaptive control approach to be favored over the simpler factorized formulation offered by implicit adaptive control techniques.

2.3.2 Model Reference Adaptive Control

A type of indirect adaptive control formulation is the Model Reference Adaptive Control (MRAC) technique. What an MRAC facilities is the ability to leverage one's understanding of the underlying physical and dynamical phenomena that describe the input/output behavior of a system [30]. Using this knowledge, a reference model can be created and set as a target reference for the to-be-controlled plant to follow;

through these manipulations, a control law is formed as a function of the plant's parameters, some of which may be well known and characterized, while others could be very uncertain, time-varying, or even completely unknown. The goal that MRAC seeks to accomplish is to pair the controller law with an adaptation law that allows the system to learn and adapt the current parameter estimates, yielding a nonlinear time-varying controller that guarantees stability and parameter convergence to its ground truth values, asymptotically [10].

While many formulations for MRAC exist — the majority of them operating within the Laplace domain — an intuitive approach for spacecraft control problems consisting of equation factorizations, stability analyses, and time-domain control and adaptation law derivations will be presented using a walked-through example.

2.3.3 Position Controller

Using the problem formulation introduced in Section 2.1, and the dynamics equation shown in Eq. (2.2), a position controller is derived. In order to encompass an entirely general case, one in which the force exerted upon the system is a linear combination of the position, velocity, and acceleration of the plant, an expression of the form

$$\mathbf{F}(t) = m \cdot \ddot{\mathbf{x}}_p(t) + \alpha \cdot \dot{\mathbf{x}}_p(t) + \beta \cdot \mathbf{x}_p(t) \tag{2.12}$$

is formulated, where $\mathbf{x}_p(t) \in \mathrm{R}^3$ represents the position states of the satellite. It is important to note that the coefficients $\alpha, \beta \in \mathrm{R}$ of the differential equation — or model — are completely unknown and form part of the parameters to be learned online.

A transformation of the second order system to a first order system is to be performed. This yields a more amenable and analogous problem that when solved, the more complicated second order problem will also be effectively solved [50]. To do this, a sliding mode approach is utilized, and a time-varying surface $\mathbf{s}_p$ comprised of a weighted sum of the system's position and velocity tracking errors is defined, obtaining

$$\begin{aligned}\mathbf{s}_p(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t} + \lambda\right)^{\bar{n}-1} \cdot \tilde{\mathbf{x}}_p(t) \\ &= \frac{\mathrm{d}\tilde{\mathbf{x}}_p(t)}{\mathrm{d}t} + \lambda \cdot \tilde{\mathbf{x}}_p(t),\end{aligned} \tag{2.13}$$

where $n = 2$ represents the order of the governing differential equation, $\lambda \in \mathrm{R}^+$ is a user-defined strictly positive constant that forms part of the design parameters (weighting factor between the position and velocities), and $\tilde{\mathbf{x}}_p(t)$ being the position tracking error defined as

$$\tilde{\mathbf{x}}_p(t) = \mathbf{x}_p(t) - \mathbf{x}_d(t), \quad \text{with}$$

$$\mathbf{x}_d(t) = [\, x_d(t) \;\; y_d(t) \;\; z_d(t) \,]^T \in \mathrm{R}^3 : \quad \text{desired reference trajectory or state.}$$

The complete expression is shown here, after plugging in and substituting for the newly defined variables, obtaining

$$\mathbf{s}_p(t) = \left(\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} - \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{z}_d(t) \end{bmatrix} \right) + \lambda \cdot \left(\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} - \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} \right).$$

Another characteristic that the variable change yields is the fact that the control input will be present within the expression after taking the first derivative of the surface. This is convenient for the upcoming analysis, in which the functions' derivatives will be analyzed for convergence properties and guarantees. Thus, although a tad less intuitively, Eq. (2.13) can be rewritten as

$$\begin{aligned} \mathbf{s}_p(t) &= \dot{\tilde{\mathbf{x}}}_p(t) + \lambda \cdot \tilde{\mathbf{x}}_p(t) \\ &= (\dot{\mathbf{x}}_p(t) - \dot{\mathbf{x}}_d(t)) + \lambda \cdot \tilde{\mathbf{x}}_p(t) \\ &= \dot{\mathbf{x}}_p(t) - \underbrace{(\dot{\mathbf{x}}_d(t) - \lambda \cdot \tilde{\mathbf{x}}_p(t))}_{:= \dot{\mathbf{x}}_r(t)} \\ &= \dot{\mathbf{x}}_p(t) - \dot{\mathbf{x}}_r(t). \end{aligned} \tag{2.14}$$

This part of the controller seeks to adaptively determine and learn the model parameters m, α, and β through a Model Reference Adaptive Controller (MRAC). Given their time-varying nature, analyses must be performed to assess the asymptotic properties of the functions involved in the controlled translational dynamics.

2.3.3.1 Stability Analysis

Since the formulation of the problem involves a non-autonomous system (after time-discretization and with time-varying parameters such as the mass), traditional Lyapunov tools and theorems — like the Invariant Set Theorem — cannot be used to reach any conclusions [10]. Instead, Barbalat's lemma will be employed to evaluate Lyapunov-like functions and analyze the convergence and stability of the system (for the full derivation, please refer to Appendix A):

Remark. *From Barbalat's Lemma, an analogous to a Lyapunov analysis invariant set theorem can be expressed. Therefore, if a scalar function $V(\mathbf{x}, t)$ satisfies the conditions*

- *$V(\mathbf{x}, t)$ is lower bounded*
- *$\dot{V}(\mathbf{x}, t)$ is negative semi-definite*
- *$\dot{V}(\mathbf{x}, t)$ is uniformly continuous in time*

then $\dot{V}(\mathbf{x}, t) \to 0$ as $t \to \infty$.

By choosing a Lyapunov-like function of the form

$$\begin{aligned} \bar{V}_p(\mathbf{s}_p, t) &= \frac{1}{2}\, \mathbf{s}_p^T(t) \cdot m \cdot \mathbf{s}_p(t) + \frac{1}{2}\, \tilde{\mathbf{a}}_p^T(t) \cdot \mathbf{P}_p^{-1} \cdot \tilde{\mathbf{a}}_p^T(t), \quad \text{where} \\ \inf \bar{V}_p(\mathbf{s}_p, t) &= 0, \quad \text{and} \\ \mathbf{P}_p \succ 0 \in \mathrm{R}^{3\times 3} &: \quad \text{user-defined system parameter matrix} \end{aligned} \tag{2.15}$$

the first condition of Barbalat's lemma is satisfied, given that it is multivariate quadratic function in terms of the position, velocity, and parameter estimates squared errors, lower bounded by zero.

By defining the unknown parameters' vector as

$$\hat{\mathbf{a}}_p(t) = \begin{bmatrix} \hat{m}(t) & \hat{\alpha}(t) & \hat{\beta}(t) \end{bmatrix}^T, \tag{2.16}$$

and taking the derivative of the candidate function while performing the adequate matrix factorizations

$$\begin{aligned} \dot{\bar{V}}_p(\mathbf{s}_p, t) &= \frac{1}{2} \cdot \left(2\mathbf{s}_p^T(t) \cdot m \cdot \dot{\mathbf{s}}_p(t)\right) + \frac{1}{2} \cdot \left(2\dot{\hat{\mathbf{a}}}_p^T(t) \cdot \mathbf{P}_p^{-1} \cdot \tilde{\mathbf{a}}_p(t)\right) \\ &= -k\, \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t) + \left(\mathbf{s}_p^T(t) \cdot \mathbf{Y}_p(t) + \dot{\hat{\mathbf{a}}}_p^T(t) \cdot \mathbf{P}_p^{-1}\right) \cdot \tilde{\mathbf{a}}_p(t), \end{aligned} \tag{2.17}$$

a term with the rate of change of the parameter estimates $\dot{\hat{\mathbf{a}}}_p(t)$ is obtained; this term is called the adaptation of the unknown parameters' vector, given that it expresses how the unknown coefficients of the differential equation are changing with respect to time.

By defining an adaptation law of the form

$$\dot{\hat{\mathbf{a}}}_p(t) := -\mathbf{P}_p \cdot \mathbf{Y}_p(t)^T \cdot \mathbf{s}_p(t), \tag{2.18}$$

and pairing it with a control law of the form

$$\begin{aligned} \mathbf{F}(t) \equiv \mathbf{u}(t) := \mathbf{Y}_p(t) \cdot \mathbf{a}_p - k \cdot \mathbf{s}_p(t), \quad &\text{where} \\ k \in \mathrm{R}^+ : \text{strictly positive user-defined parameter} & \end{aligned} \tag{2.19}$$

a negative semi-definite function for the time derivative of the Lyapunov-like function is obtained,

$$\dot{V}_p(\mathbf{s}_p, t) = -k\, \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t) \leq 0. \tag{2.20}$$

This indicates that, as the lemma and corollaries state, as $t \to \infty$ the asymptotic entailments

$$\dot{V}_p(\mathbf{s}_p, t) \to 0 \Longrightarrow \begin{Bmatrix} \tilde{\mathbf{a}}_{\mathbf{p}}(\mathbf{t}) \to 0 \\ \mathbf{s}_p(t) \to 0 \end{Bmatrix} \Longrightarrow \begin{Bmatrix} \tilde{\mathbf{x}}_{\mathbf{p}}(\mathbf{t}) \to 0 \\ \dot{\tilde{\mathbf{x}}}_p(t) \to 0 \end{Bmatrix}$$

hold, and the controller tends towards a stable perfect tracking state, provided the references are bounded and stable (BIBO) ■.

2.3.4 Attitude Controller

Given that no linearization steps were performed throughout the formulation for the position controller, by following the same formulation an adaptive attitude control law can be formulated by using the differential equation

$$\boldsymbol{\tau}(t) = \mathbf{J} \cdot \ddot{\boldsymbol{\theta}}_o(t), \tag{2.21}$$

and the system matrix and vector factorizations of the form

$$\mathbf{Y}_o(t) = \begin{bmatrix} \ddot{\theta}_{x_r}(t) & \ddot{\theta}_{y_r}(t) & \ddot{\theta}_{z_r}(t) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddot{\theta}_{x_r}(t) & \ddot{\theta}_{y_r}(t) & \ddot{\theta}_{z_r}(t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddot{\theta}_{x_r}(t) & \ddot{\theta}_{y_r}(t) & \ddot{\theta}_{z_r}(t) \end{bmatrix}, \text{ and} \tag{2.22}$$

$$\hat{\mathbf{a}}_o(t) = \begin{bmatrix} \hat{J}_{xx} & \hat{J}_{xy} & \hat{J}_{xz} & \hat{J}_{yx} & \hat{J}_{yy} & \hat{J}_{yz} & \hat{J}_{zx} & \hat{J}_{zy} & \hat{J}_{zz} \end{bmatrix}, \tag{2.23}$$

with $\dot{\boldsymbol{\theta}}_r(t) = \dot{\boldsymbol{\theta}}_d(t) - \lambda_o \boldsymbol{\theta}_o(t)$. The system's factorizations are then to be paired with

$$\dot{\hat{\mathbf{a}}}_o(t) := -\mathbf{P}_o \cdot \mathbf{Y}_o(t)^T \cdot \mathbf{s}_o(t) \quad \text{(adaptation law), and} \tag{2.24}$$

$$\boldsymbol{\tau}(t) \equiv \mathbf{u}_o(t) := \mathbf{Y}_o(t) \cdot \hat{\mathbf{a}}_o - k \cdot \mathbf{s}_o(t). \tag{2.25}$$

The model reference adaptive attitude controller is then combination of the adaptation law (2.24) and the control law (2.25), which are a function of the unknown parameters of the system, and the performed matrix factorizations.

2.3.4.1 Clohessy-Wiltshire Dynamics Formulation

Given that the analysis shown was performed as a function of a general differential equation expressing the dynamics of a system and an arbitrary state vector, the formulation can easily be extended to handle more complex systems or mathematical models; one such system of equations is the Clohessy-Wiltshire dynamics used to express the relative motion between two spacecraft in close proximity. The derivation of an adaptive controller using these differential equations is relevant, since the objective of this work is to contribute towards achieving reliable in-space robotic assembly.

By taking the equations

$$\begin{aligned} \ddot{x}(t) &= 3n^2 \cdot x(t) + 2n \cdot y(t) \\ \ddot{y}(t) &= -2n \cdot \dot{x}(t) \\ \ddot{z}(t) &= -n^2 \cdot z(t), \end{aligned} \tag{2.26}$$

where $n = \sqrt{\mu / a^3}$ a controlled formulation in matrix form yields

$$\frac{\mathbf{F}(t)}{m} = \ddot{\mathbf{x}}_p(t) + \boldsymbol{\Phi} \cdot \dot{\mathbf{x}}_p(t) + \boldsymbol{\Theta} \cdot \mathbf{x}_p(t), \tag{2.27}$$

with coefficient matrices

$$\mathbf{\Phi} = \begin{bmatrix} 0 & -2n & 0 \\ 2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{ and } \qquad \mathbf{\Theta} = \begin{bmatrix} -3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & n^2 \end{bmatrix}.$$

Performing now a sliding mode transformation into a first order system with a surface variable $\mathbf{s}_p^{cw}(t)$, a Lyapunov-like function $V_p^{cw}(\mathbf{s}_p^{cw}, t)$ can be defined to determine the factorization utilized within the control and adaptation laws.

By taking the time derivative of the function

$$\begin{aligned} \dot{V}_p^{cw}(\mathbf{s}_p^{cw}, t) &= \frac{1}{2} \cdot \left(2\mathbf{s}_p^{cwT}(t) \cdot m \cdot \dot{\mathbf{s}}_p^{cw}(t)\right) \\ &= \mathbf{s}_p^{cwT}(t) m \left(\ddot{\mathbf{x}}_p(t) - \ddot{\mathbf{x}}_r(t)\right) \\ &= \mathbf{s}_p^{cwT}(t) m \left(\frac{1}{m}\mathbf{F}(t) - \ddot{\mathbf{x}}_r(t) - \mathbf{\Phi} \cdot \dot{\mathbf{x}}_p(t) - \mathbf{\Theta} \cdot \mathbf{x}_p(t)\right) \\ &= \mathbf{s}_p^{cwT}(t) \left(\mathbf{F}(t) - \overbrace{m}^{\text{unknown}} \cdot \underbrace{\ddot{\mathbf{x}}_r(t)}_{\text{known}} - \overbrace{m \cdot \mathbf{\Phi}}^{unknown} \cdot \underbrace{\dot{\mathbf{x}}_p(t)}_{\text{known}} - \overbrace{m \cdot \mathbf{\Theta}}^{unknown} \cdot \underbrace{\mathbf{x}_p(t)}_{\text{known}}\right), \end{aligned} \tag{2.28}$$

and factorizing out the result into

$$\dot{V}_p^{cw}(\mathbf{s}_p^{cw}, t) = \mathbf{s}_p^{cwT}(t) \cdot \left(\mathbf{F}(t) - \mathbf{Y}_p^{cw}(t) \cdot \mathbf{a}_p^{cw}\right),$$

the control and adaptation law known matrix and unknown regressor vector are obtained,

$$\mathbf{Y}_p^{cw} = \begin{bmatrix} \ddot{x}_r^{cw}(t) & \dot{y}(t) & x(t) \\ \ddot{y}_r^{cw}(t) & -\dot{x}(t) & 0 \\ \ddot{z}_r^{cw}(t) & 0 & -\frac{1}{3}z(t) \end{bmatrix}, \tag{2.29}$$

and

$$\hat{\mathbf{a}}_p^{cw}(t) = \left[\ \hat{m}(t) \quad -2\hat{m}\hat{n} \quad -3\hat{m}\hat{n}^2\right]^T. \tag{2.30}$$

The control law for a system governed by Clohessy-Wiltshire dynamics would then take the form

$$\begin{aligned} \mathbf{F}(t) \equiv \mathbf{u}^{cw}(t) := \mathbf{Y}_p^{cw}(t) \cdot \hat{\mathbf{a}}_p^{cw} - k \cdot \mathbf{s}_p^{cw}(t), \quad & \text{where} \\ k \in \mathrm{R}^+ : \text{strictly positive user-defined parameter} & \end{aligned} \tag{2.31}$$

complemented by the adaptation law

$$\dot{\hat{\mathbf{a}}}_p^{cw}(t) := -\mathbf{P}_p^{cw} \cdot \mathbf{Y}_p^{cw}(t)^T \cdot \mathbf{s}_p^{cs}(t). \tag{2.32}$$

It is worth noticing the power of the formulation; not only is one now able to perform inferences upon the satellites mass characteristics, but also over the spacecraft's trajectory details. The system identification that comes for free by employing this control technique (indirect adaptive control) allows for the learning of parameters such as the orbital radius, given that $n = \sqrt{\mu/a^3}$ (assuming a known μ).

If inference and adaptation wants to be performed over specific parameters, one just needs to modify the process model to include them in its formulation and perform the appropriate factorization for obtaining the desired vector of unknowns. The control law provides asymptotic stability guarantees, and true parameter value convergence given sufficient richness in the tracking signal.

2.4 Summary

The mathematical framework describing the dynamics of the system, as well as the sources of disturbances and uncertainties were introduced in the first section of the chapter. Subsequently, two control methodologies were presented: the traditional control methods used for SPHERES, which involve linear and nonlinear PD controllers, and a learning approach consisting of a Model Reference Adaptive Control strategy. Now that their formulation were posed, these approaches will be compared and contrasted to assess their suitability to the in-space robotic assembly problem.

Chapter 3

Overcoming System Uncertainties

One key aspect surrounding on-orbit servicing and assembly maneuvers is the level of uncertainty that comes into play; clearly, it is this stochastic component that poses a challenge for the traditional control systems designs and approach [7].

In order to elucidate on this aspect, analyses on the control authority and performance achieved during MIT SSL's InSPIRE-II Test Sessions is shown as further motivation.

Subsequently, a learning approach using a Model Reference Adaptive controller is analyzed in simulation, and glass table testing results comparing its performance against the baseline SPHERES controller using unknown hardware configurations is shown.

3.1 Prelude to Adaptive Control

In spite of the fact that the control theory field is a rapidly evolving area, the vast majority of control systems designed for spacecraft still utilize the so-called classical and modern control theory techniques [47]. These techniques include Laplace-domain driven control (e.g., the almighty Proportional-Integral-Derivative controllers), Pole-Placement methods for dealing with mostly second order systems (e.g., lead and lag controllers), and more importantly, guarantee-offering robust control formulations (e.g., $\mathcal{H}_\infty$). Additionally, whenever a certain performance metric needs to be handled, criteria-optimizing controllers — such as the Bang-Bang (min-time) or minimum fuel/energy controllers — are utilized, given that they offer a much more intuitive and useful methodology for designing control systems with cost functions, being able to leverage the techniques of the optimal control theory field [51].

Given an accurate dynamics model of a system, paired with an accurate knowledge of its parameters' values, these aforementioned control techniques, needless to say, excel in their performance and accuracy [52]. Within the aerospace realm, this is almost always the case, given the stringent, extensive, and thorough verification and validation processes a spacecraft needs to undergo before being put into orbit [53]. The main objective of these exhaustive testing methods is the elimination, or at least, minimization of the uncertainties within the system that hinder performance and increase the inherent risk of encountering a failure.

3.1.1 SPHERES Testbed Platform

A testament to these claims is the SPHERES framework, a testbed platform for control, estimation, and autonomy algorithms that operates on board the International Space Station (ISS) [54]. Given its spherical-like design and the strategic positioning of its 12 CO_2 thrusters that enable it to operate in full 6DOF microgravity environment, the controllability able to be exerted upon the standalone[1] satellite is remarkable. By having its geometric center, ensemble center of mass, and thruster pivoting points almost perfectly aligned, the degree of control accuracy allowed for formation flight algorithms [55, 56], soft docking algorithms [57], collision avoidance algorithms [58], amongst others, to be successfully demonstrated on-orbit while relying on traditional control techniques.

Nowadays, test sessions involving the standalone SPHERES keep demonstrating that, even after 10 years of continuous operation inside the ISS, successful results can still be obtained while relying in these aforementioned traditional control techniques;

[1] by standalone it is meant without any of the newer SPHERES attachments such as the VERTIGO Avionics Stack, the stereo-vision Optics Mount, Universal Docking Ports, or the Halo Expansion Device.

the Zero Robotics competition is ran semi-annually, yielding adequate performances from PID controllers, as well as the Tether experiment, which uses a pair of SPHERES to study the dynamics of tethered ensembles in microgravity.

This is, however, not the case for projects involving external attachments to the SPHERES satellite that completely modify the mass properties of the original system (some of which are shown in Fig. 3-1). One such project is the InSPIRE-II program, which needs to make use of Universal Docking Ports (UDP) and Halo expansion devices to achieve fully autonomous docking and reconfiguration maneuvers.

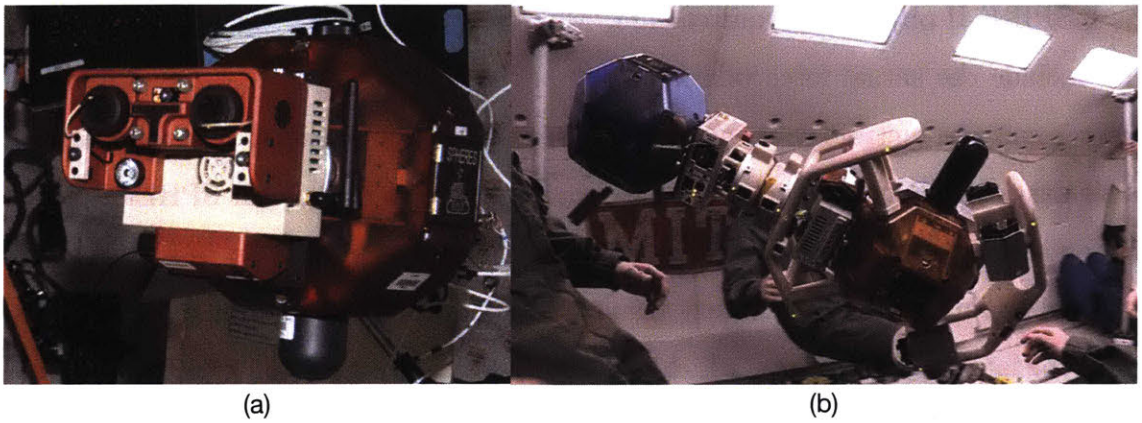


Figure 3-1: Some of the attachments available for the SPHERES framework: (a) a SPHERES equipped with the VERTIGO Avionics Stack (VAS) and Optics Mount; (b) Orange SPHERES equipped with the Halo expansion device, VAS, and a Universal Docking Port (UDP), while the Blue SPHERES contains only a VAS and a UDP.

3.1.2 InSPIRE-II Program

The InSPIRE-II program is a DARPA funded project whose objective is to mature the guidance, navigation, and control algorithms for in-space robotic assembly and servicing missions through a risk tolerant, and dynamically authentic environment. The addition of the UDP and the Halo peripherals were realized as part of the program in order to fulfill the needed hardware requirements. With this enhanced hardware suite, the requirements warranted by maneuvers such as autonomous docking and on-orbit robotic assembly could be met.

The complexity of the augmented hardware ensemble rendered ground system identification tests to be very complicated; a thorough analysis was not performed, and best estimates from CAD models were initially used. In order to improve upon these characteristics, RGA flights were performed to assess the system's overall functionality and to obtain a new prior on the new physical properties, such as mass and inertia quantities.

It is clear then that now, the challenge is not only having to deal with a modified system, whose true mass and inertia properties were not directly measured but inferred from CAD models and on-orbit system identification tests, but also to satisfy the demands for greater accuracy and precision from the control algorithm posed by the InSPIRE-II objectives requirements. Moreover, additional sources of uncertainty that include the actual force value for each of the thrusters, the ultrasound localization sensors and metrology estimation, environmental and external disturbances such as air drag and IR noise, plus the non-negligible 10 years of use that SPHERES have had on station, only worsen the situation in terms of complexity.

To elaborate on the preliminary results obtained from the program's first test sessions, the latest batch of tests, which include UDP Science 1.5 (TS92, 04.05.2017) and Halo Checkout (TS86, 02.03.2017), will be briefly discussed. While the Halo Checkout test session yielded satisfactory results in terms of objectives met (full functionality of the hardware), both test sessions demonstrated a sheer lack of controllability of the entire ensemble. Using telemetry data, individual degrees of controllability will be assessed for each of them separately and then compared.

3.1.3 Docking Port Science: TS92

The objective of Test Session 92, UDP Science 1.5, was to demonstrate autonomous docking algorithms on-orbit using the SPHERES satellite paired with a VAS and a UDP, configuration shown in Fig. 3-2. Given that controllability wanted to be ensured, the entire docking procedure was sliced into all of its simpler components: a reference tracking step and glideslope translation maneuver, and a reference tracking attitude correction maneuver. The ability to precisely perform this two movements is paramount to the success of the more elaborate algorithms, since they represent the building blocks upon which more complicated tests are formed.

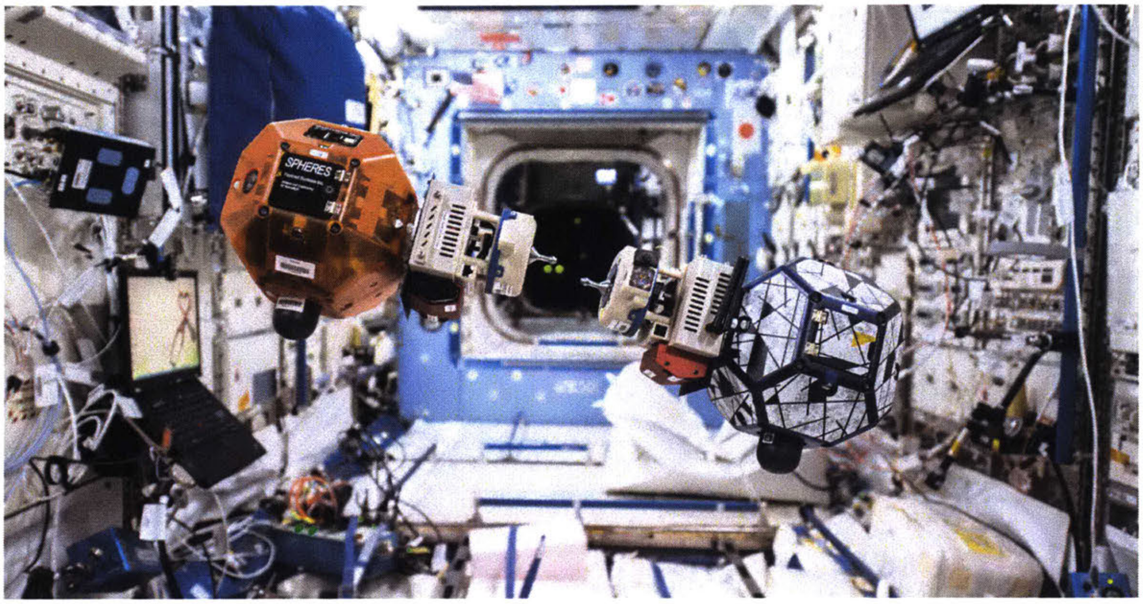


Figure 3-2: Blue and Orange SPHERES attempting to autonomously dock with their respective VERTIGO Avionics Stacks and Universal Docking Ports during a run of Test Session 78: SPHERES Docking Port Checkout.

Despite the complexity reduction of the tests to be performed by the SPHERES ensemble, satisfactory results were not able to be obtained; not even the simplest of maneuvers, such as a basic position- and attitude-hold or a one-axis translation, were able to be performed to such a degree as to be compared to the performance obtained from previous standalone SPHERES test sessions.

An example of this is shown in Fig. 3-3, in which a so-called simple dock[2] maneuver was to be performed. The position and attitude plots for Orange SPHERES satellite corresponding to that test are shown in Figure 3-4. It can be seen from the video and data plots that coarse action maneuvers like recovering from poor initial conditions are able to be decently performed, but once finer and more precise maneuvers, like position- and attitude-holding, come into play the satellites do not perform as expected.

This is especially noticeable in the Orange satellite's behavior, which is supposed to act as a static target throughout the test. Instead, its control performance is evi-

[2] the "simple dock" test commands the target satellite to hold position and attitude at a reference point in the global frame of reference, independent of the chaser satellite, who is commanded to approach the target satellite's inertial reference position through a series of glideslope maneuvers, effectively uncoupling all motion between both parties.

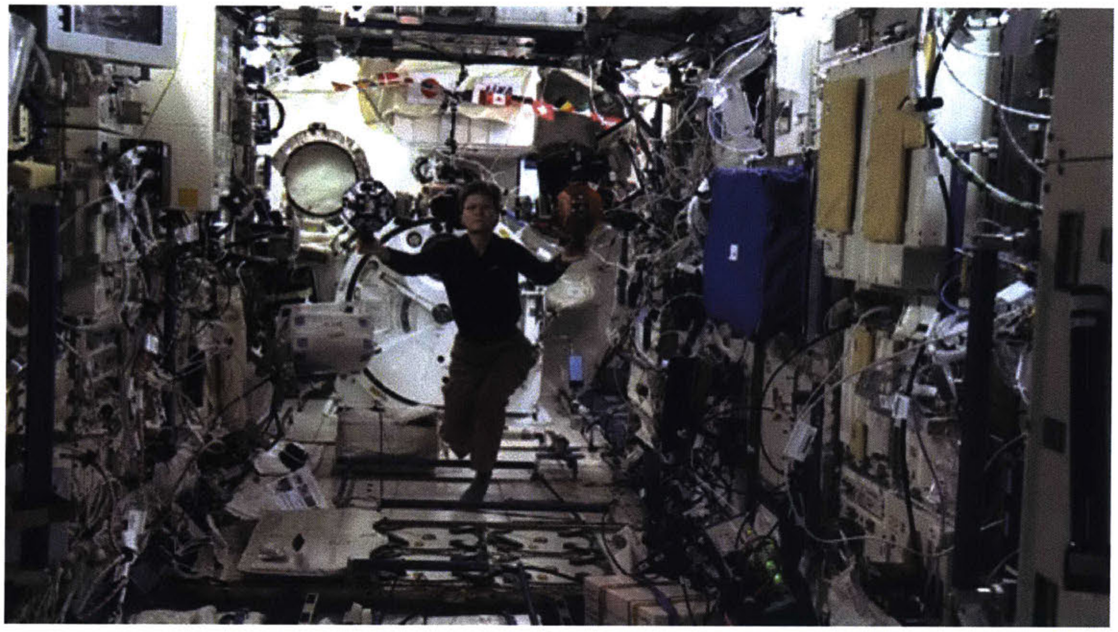
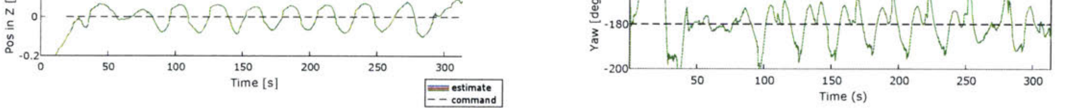


Figure 3-3: Video of a Simple Dock (Run #1) maneuver carried out by the SPHERES ensemble during Test Session 92: Docking Port Science 1.5. (Video speed: ×80)



(a) Position histories for Orange satellite.

(b) Attitude histories for Orange satellites.

Figure 3-4: Plots corresponding to the Orange SPHERES satellite during Simple Dock Run#1 shown in Fig. 3-3: position tracking (a) shows small deviations from the commanded references, while attitude performance (b) exhibits non-negligible disturbances ($t \approx 170\ s$).

dently lacking, and a limit-cycle type of behavior is obtained. The coupling between attitude and position control given the thruster's geometry and (unknown) center of mass location is well represented in Fig. 3-4. The observed pattern is interesting, since it does not seem to be neither diverging nor converging. The Blue satellite's performance is significantly better than its counterpart, and its attitude tracking is only disturbed by a backtracking action commanded by the glideslope algorithm (returning to a previous gate point due unsatisfied attitude and position constraints).

3.1.4 Halo Checkout: TS86

On the contrary, the main objective of Test Session 86: Halo Checkout was to first ensure the state of functionality of the hardware; a picture of the newly-arrived (to the ISS) Halo Expansion devices is shown in Fig. 3-5. Nonetheless, similar controllability tests as in TS92 for position and attitude were able to be performed, this time utilizing a SPHERES with its Halo Expansion Device, suited with three Universal Docking Port peripherals.

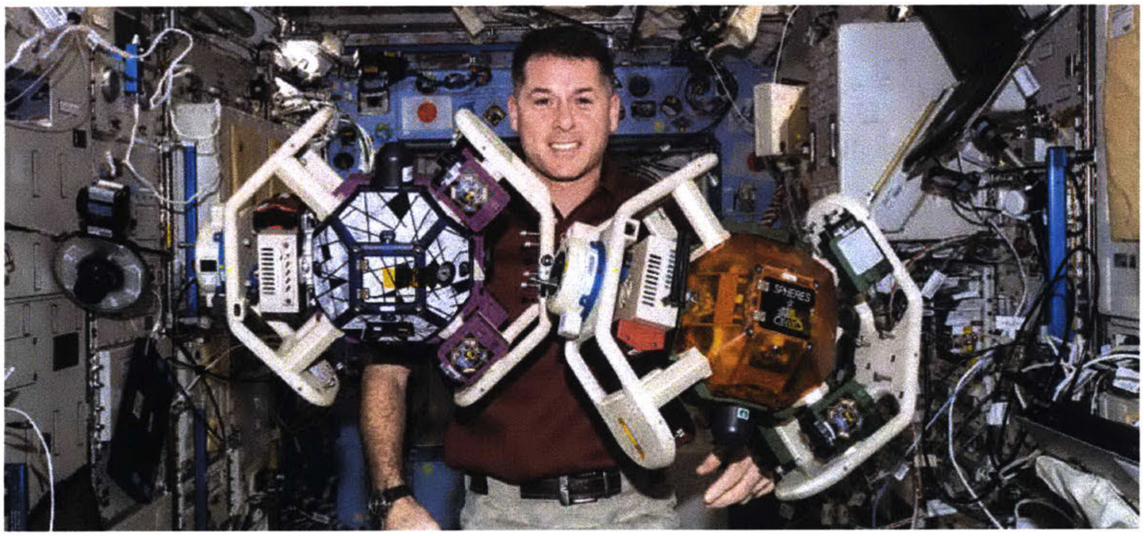


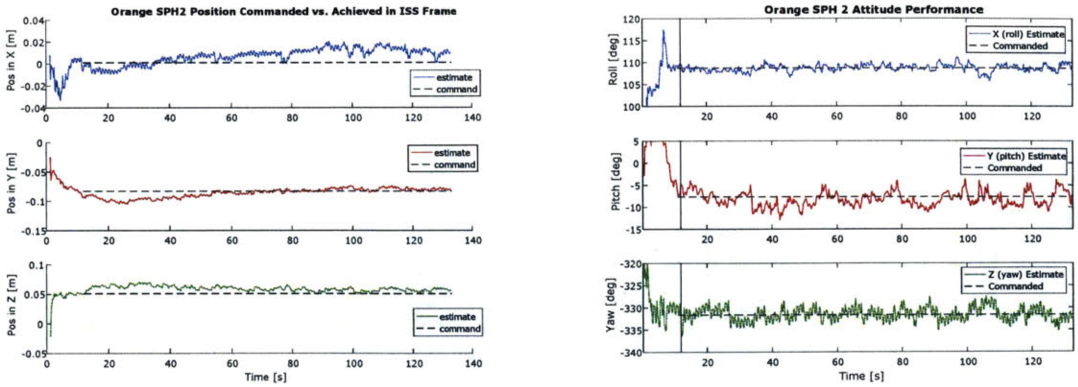
Figure 3-5: Expedition 50 Commander Shane Kimbrough of NASA floating around with the brand new SPHERES Halos on-board the International Space Station. Picture taken during TS86: Halo Checkout.

With the full ensemble being more massive, given the size of the Halo, the question of whether better control authority could be exerted upon this bigger assembly was sought to be answered. It is important to note that no on-orbit system identification tests had been run on Halo before, and best estimates of physical parameters obtained from CAD and the RGA flight were utilized.

The SPHERES behavior during position-hold tests was very stable, as is shown in Fig. 3-6. The corresponding plots for the Orange satellite are shown in Figure 3-7. Even though no specific analogous tests such as this one were run during TS92, the aforementioned "Simple Dock" test includes a simple position-hold maneuver for the target satellite, which is exactly comparable to this test; it was shown that the Orange satellite was not able to stabilize its position and attitude states, and that undamped oscillations were always present, whereas with the attached Halo configuration, no instabilities were able to be directly observed.

Figure 3-6: Video of a Position-Hold (Run #4) maneuver carried out by the SPHERES + Halo ensemble during Test Session 86: Halo Checkout. (Video speed: ×16)

Therefore, some of the position-hold maneuvers with the Halo Expansion assembly show much promise, given that centimeter accuracy for position and single digit degree accuracy for attitude was obtained, as can be easily observed from the corresponding



(a) Position histories for Orange satellite.

(b) Attitude histories for Orange satellites.

Figure 3-7: Plots corresponding to the Orange SPHERES satellite during Position-Hold#4 shown in Fig. 3-6: similar to Blue's behavior, both (a) position and (b) attitude tracking show very small deviations from the commanded references.

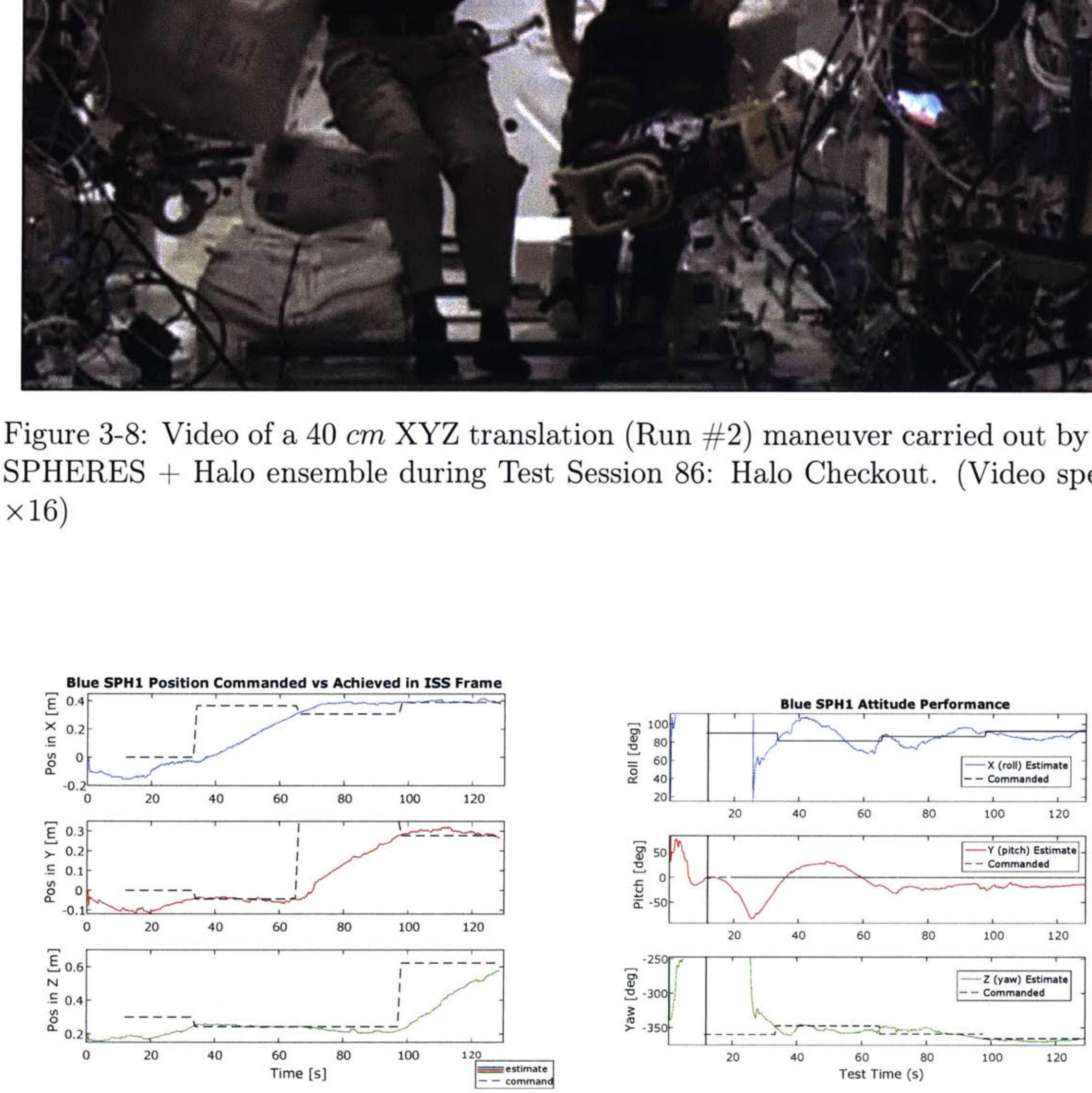
plots. However, the same performance is not entirely obtained when slightly more complicated tasks such as translation are carried out. A translation test run in which the satellites had to sequentially travel 40 *cm* in the positive x-direction, 40 *cm* in the positive y-direction, and 40 *cm* in the positive z-direction is shown in Fig. 3-8.

Unfortunately, as seen in the video, the Orange satellite runs out of CO_2 gas during the run and thus is not able to successfully complete the test. The data obtained for the Blue satellite is shown in Fig. 3-9. A more stable response, both in position and in attitude, as compared to results from TS92 are obtained, although not with centimeter accuracy as with the position-hold maneuvers previously shown. In Fig. 3-9(a) the translation histories for each axis composed of slightly overdamped responses are shown; intuitively, this behavior makes sense, as SPHERES with Halo form a more massive system that needs to be controlled by the same actuators. In Fig. 3-9(a), the tracking performance seems to be following the correct trend, but still large angle errors are seen throughout the test.

During these test sessions, the control strategies employed by the satellites relied on PID and PD controllers for the position states, and on quaternion feedback strategies[3] for attitude control (using a "proportional" gain for the nonlinear quaternion

[3] control law of the form $\mathbf{u}(t) = -\mathbf{K}\mathbf{q}_e(t) - \mathbf{C}\dot{\boldsymbol{\theta}}$, with $\mathbf{K} \in \mathcal{R}^{3\times 4}$ acting as a proportional gain multiplying the current orientation error of the satellite expressed in quaternion form, and $\mathbf{C} \in \mathcal{R}^{3\times 3}$

Figure 3-8: Video of a 40 *cm* XYZ translation (Run #2) maneuver carried out by the SPHERES + Halo ensemble during Test Session 86: Halo Checkout. (Video speed: ×16)



(a) Position histories for Blue satellite.

(b) Attitude histories for Blue satellites.

Figure 3-9: Plots corresponding to the Blue SPHERES satellite during the 2nd run of the 40 *cm* XYZ-Translation test shown in Fig. 3-6.

states and a "derivative" gain on the angular rates). Again, constant values for the system's physical parameters were considered, and no adaptive techniques were utilized to account for any uncertainty. Needless to say, the approach was unable to fulfill the accuracy and precision requirements warranted by the InSPIRE-II objectives.

Consequently, much thought has been given to determining the root of the problem, and causes such as pressure gauge irregularities, metrology and localization aspects, or infrared noise have been rendered to be less likely when compared to the trajectory generating and control algorithms, both of which heavily rely on precise knowledge of the mass characteristics of the system. This leaves as an ultimate culprit the fact that the system and its properties are not known accurately enough to exert sufficient control authority for achieving the program's objectives using the current algorithms.

Summing-up Even though a slightly better overall performance was shown while using the SPHERES paired with the Halo devices as compared to the UDP configuration, there is still a need for improving the position and attitude control performance if maneuvers such as autonomous docking want to be reliably performed. Conclusions and insights on further actions to be taken that could potentially alleviate the observed problem are proposed and discussed in the following section.

3.2 On-Board Learning through MRAC

So far, it has been shown how extremely effective traditional time-invariant controllers are whenever a plant's model, parameters, and variables are well known. After all, a perfectly tuned Linear Quadratic Regulator (LQR) or Proportional-Derivative-Controller (PID) will always outperform an adaptive or iterative learning approach, given that the latter tend to the former as its parameter value estimates start to converge to their true value.

Nonetheless, there are scenarios in which high nonlinearities are present through-

acting as a gain for the rate of change of the satellite's orientation, i.e., a derivative gain [59, 47].

out, and where a high-fidelity characterization of the system's properties is intractable/unfeasible to obtain [60]. Space missions involving on-orbit servicing and assembly objectives push these systems to operate outside the classical control methods' thriving regions, which is the space in which one is able to linearize a system, and fully describe it using a time-invariant model [7].

Simulation results are first be shown to demonstrate the learning behavior of the algorithm, followed by hardware implementation results from ground testing using the SPHERES satellites and the glass table facilities.

3.2.1 Simulation Tests and Results

Using the presented formulation for a position MRAC given a system with double integrator dynamics, an instantiation of such a controller was implemented in order to assess its performance with respect to the mass characterization of a satellite.

Assuming perfect full-state feedback information within the simulator, several important characteristics of the Model Reference Adaptive Controller can be observed. The simulation performed assumed **zero prior knowledge** of the satellite's mass property, initializing with a value of $\hat{m}_0 = 0\ kg$. The ground truth value of the spacecraft's mass was set to be of $m^* = 3\ kg$.

A sinusoidal signal in all three dimensions was utilized as the desired tracking reference state. This desired state is what then allows for the computation of the tracking error, which is just the difference between the current state and this reference. It is exactly this tracking error which acts as the information source for the inferences taken by the adaptation law; this is easily seen in Eq. 2.18, in which the rate of change of the estimates is posed as a function of the tracking error $\mathbf{s}_p(t)$ variable. Once a perfect tracking state is reached (i.e., zero tracking error obtained), the rate of change of the estimates in the unknown quantities vector $\hat{\mathbf{a}}_p$ would be null, reaching a constant value for the plant's parameters.

The phenomenon described can be very clearly seen in Figure 3-10. This is an extremely important characteristic of the MRAC that needs to be taken into consideration when the implementation wants to be carried out in real life. The adaptation happens on a so-called "need-to-know" basis, which means that even when the system hasn't been fully identified and characterized (as is the case for Fig. 3-10 at time $\approx 8\ s$), if there is no more need for the adaptation to happen (i.e., perfect tracking objective fulfilled), then the system just keeps its current parameter estimates, which are just "good enough" for the perfect tracking to be performed.

Conversely, if sufficient richness within the tracking error signal is available, full system identification and parameter characterization is possible, as is shown in Figure 3-11.

It is clear to see in Fig. 3-11(a) that a full characterization of the mass parameter is obtained, with the estimated value converging to the ground truth at time $t \approx 12\ s$. As compared with Fig. 3-10, in Fig. 3-11(b) a larger amount of information carried by the tracking error signal (larger errors) is shown; it is thus clear that the more drastic deviations from the reference objective alloted a broader "sufficient richness region", in which persistent excitation was supplied to the adaptation law and the convergence properties of the adaptive controller were able to be sufficed. The gap between the full characterization period region and the sufficient richness region allows one to see that while a lack of persistent excitation signifies an incomplete characterization procedure (e.g., see Fig. 3-10), and "excess" of information is of no particular relevance to the behavior fo the **deterministic** MRAC; this key point is shown is of great impact for the stochastic version of the algorithm and is addressed in the following sections.

Persistent Excitation For this aforementioned reason, a term coined *persistence of excitation* is used to describe the amount of inference information that a signal is delivering to the adaptive controller [61]. This allows for the introduction of a very important caveat, which is the trade-off between performing a full system identification upon the satellites parameters and performing as optimal of a maneuver as possible. It is clear then that for the convergence of the estimates to its true parameter values a minimum amount of information needs to be obtained, which in the MRAC case would be through the tracking error signal.

Hence, to ascertain a parameter estimate convergence, techniques such as pur-

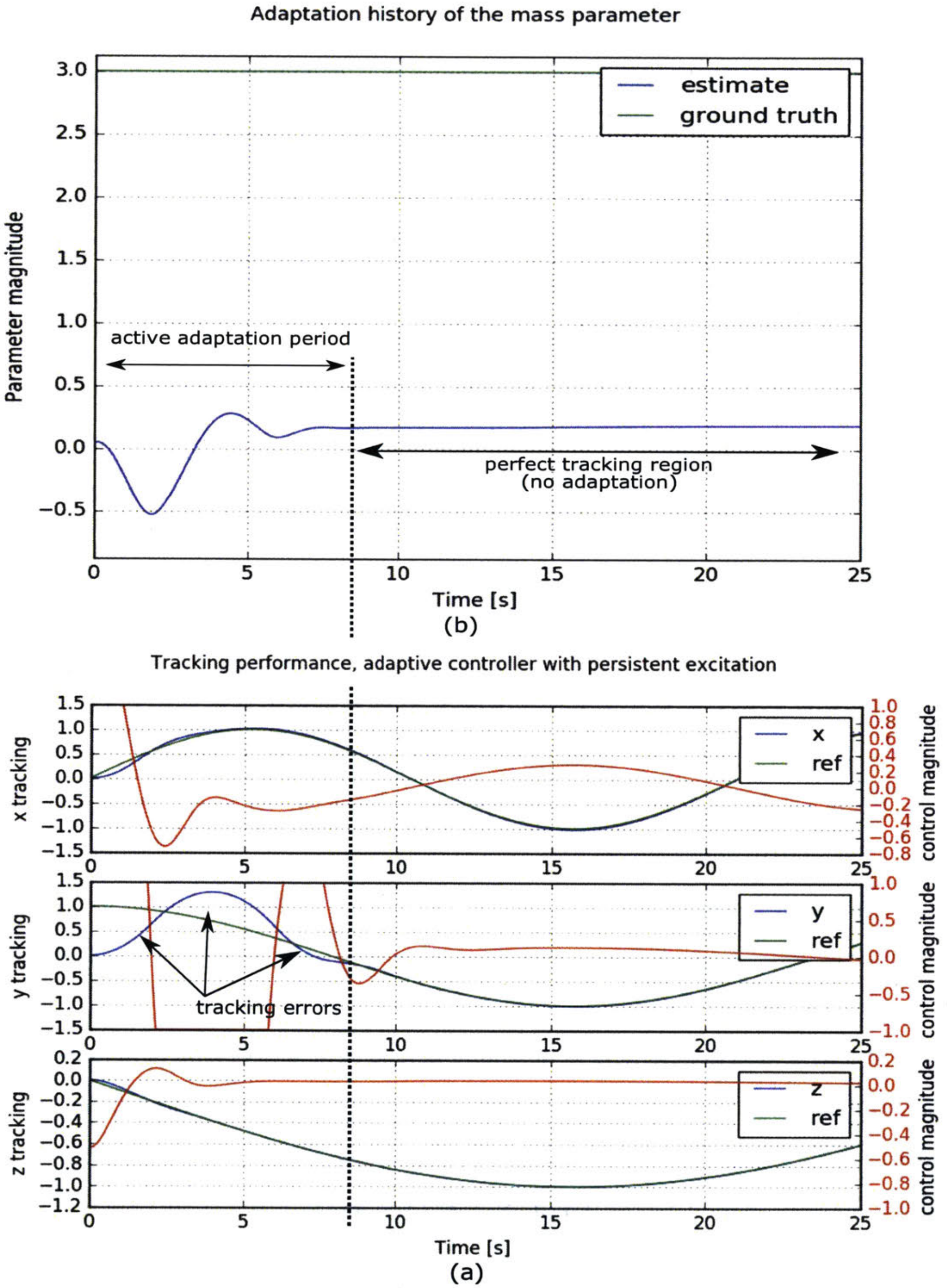
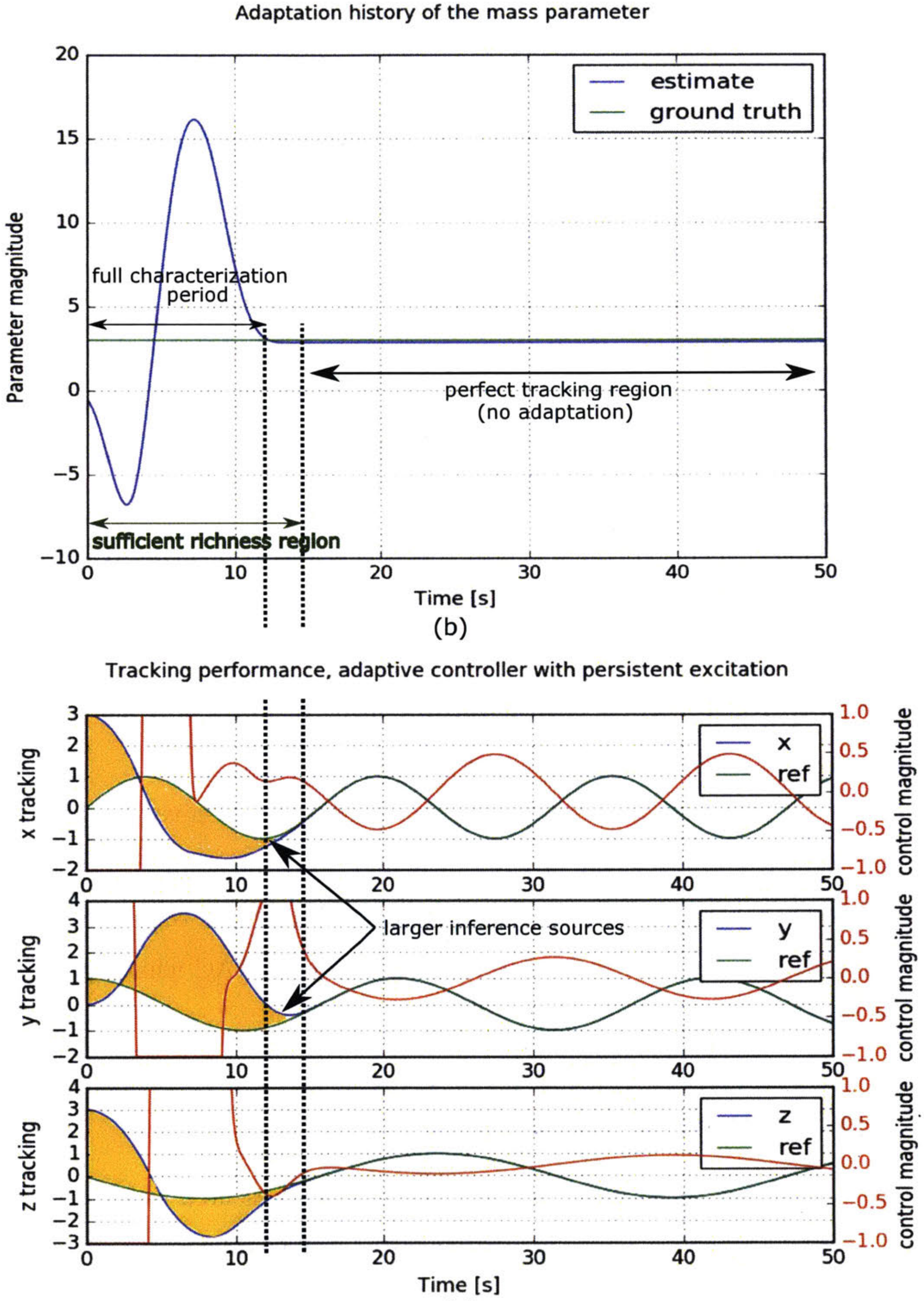


Figure 3-10: Deterministic MRAC run (perfect full-state feedback): (a) adapted parameter history for the mass estimate under the absence of persistent excitation, (b) state, reference, and control input histories; it can be clearly seen how the adaptation stops after the system reaches a perfect reference tracking state (at ≈ 8 s). Initial estimate $\hat{m}_0 = 0$ kg, true mass parameter $m = 3$ kg.



(b)

(a)

Figure 3-11: Deterministic MRAC run (perfect full-state feedback): (a) full characterization under the presence of persistent excitation, (b) state, reference, control input, and available inference information (yellow); estimated value fully converges to the ground truth parameter. Initial estimate $\hat{m}_0 = 0\ kg$, true mass parameter $m = 3\ kg$.

posefully introducing errors to the system, slight intentional deviations from reference states, or constant actuator usage for obtaining a signal with sufficient richness are commonly described in the literature [62]. However, for space systems, this poses an unacceptable practice, since fuel is a scant and precious resource that cannot necessarily be wasted in this fashion [63]. This limitation is further analyzed, and solutions are proposed throughout the next sections of this work.

Perfect Tracking Region As is shown towards the end of the test in both Fig. 3-10(b) and 3-11(b), once the system enters a perfect tracking region, the parameter estimates stabilize, remain constant (no learning is available when $\mathbf{s}_p(t) = 0$), and the state gracefully slides along the hypersurface defined by the weighted sum of the position and velocity tracking errors. This behavior can certainly only happen whenever perfect full-state feedback is available, which implies operation within a deterministic framework. Unfortunately, the real world is not as kind, and such a thing as a perfect sensor or perfect measurement readings does not apple when transitioning to plausible work environments and scenarios.

Needless to say, such a thing as $\mathbf{s}_p(t) \equiv 0$ would be infeasible given the all-pervasive effect of noise withing a real-life environment and the truncation errors inherent to numerical techniques (computers are not perfect). Throughout the next section, the effect caused by these aforementioned properties to the MRAC while operating in a stochastic world are shown and analyzed.

3.2.2 Glass Table Testing

The Model Reference Adaptive Controller formulation presented in Subsection 2.3.2 was arranged into an algorithm to be further tested using MIT Space System Laboratory's SPHERES ground testing facilities. Using a SPHERES satellite, an air carriage base, and a glass table, a 3DOF — x and y axis translation plus z axis rotation [yaw] — nearly frictionless environment can be recreated. The setup is figured in Fig. 3-12.

To assess the performance of the adaptive controller, a comparison against the SPHERES standard PD controller was realized. The test chosen for this experiment

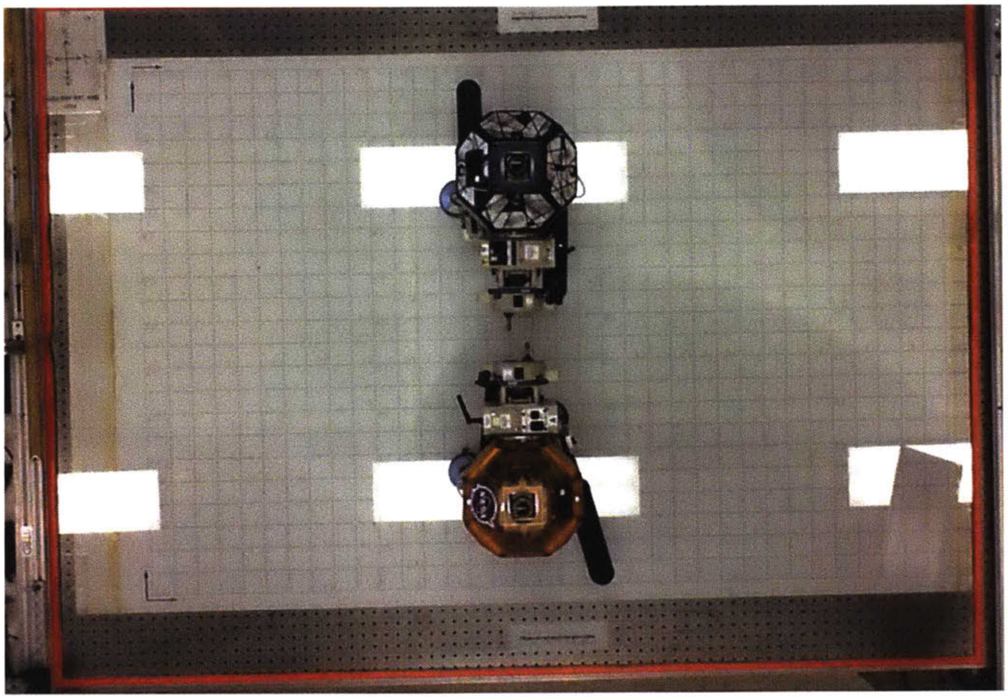


Figure 3-12: SSL's SPHERES ground testing facilities.

was a simple regulator maneuver, in which the satellite was to be initialized at an offset from the glass table's origin (x : 0.0 m, y : 0.0 m), and the controller had to then compute the necessary torques and forces to bring the system to the origin and position-hold at that point. A pictorial depiction of the test overview can be found on Fig. 3-13.

A formulation for both position and attitude adaptive control was utilized, and in order to rigorously test them, **initial estimates** for the mass parameter and for the moments and products of inertia were taken to be **equal to zero** ($\hat{m}_0 = 0\ kg$, $\hat{\mathbf{J}}_0 = \mathbf{0}^{3\times3}$). The baseline and standard SPHERES PD controller was granted the physical properties corresponding to ones determined during a Reduced-gravity flight experiment with the Halo device, with a Universal Docking Port attached to Halo-Port 6 (Configuration #1 for the realized tests).

Given the fact the the center of mass will not, almost surely, be aligned with the thruster's actuation center point, the position and attitude dynamics are certainly rendered to be coupled. Even if the satellite's initial position starts with an attitude error close to zero (UDP pointing towards the positive x-axis), the offset in position

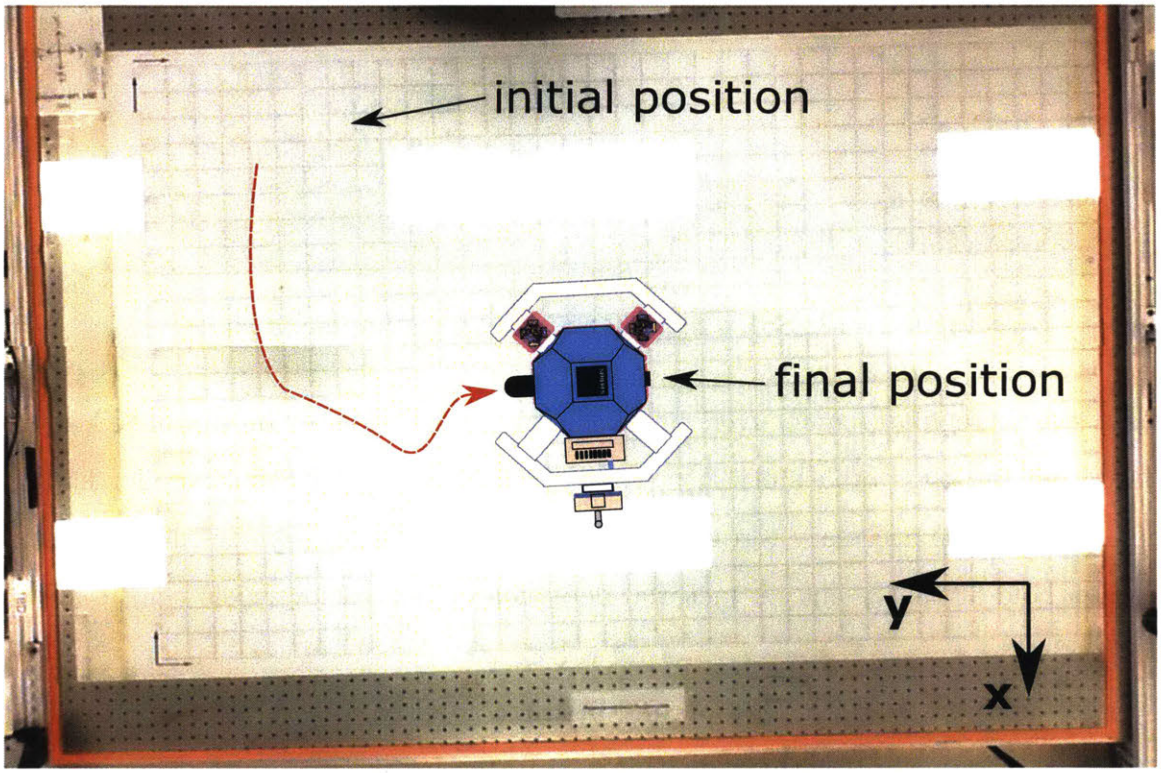


Figure 3-13: Regulator test overview for comparing the performance of the two controllers. The SPHERES/Halo assembly starts at an offset position from the origin, and needs to find its way towards the origin of the glass table.

will induce changes in the yaw angle, being able to test both controllers simultaneously.

3.2.2.1 Hardware Configurations

It is important to ensure that the adaptation could handle distinct magnitudes of uncertainty, and for this, three different hardware configurations were considered. All three of them are shown in Fig. 3-14 with their respective attachments.

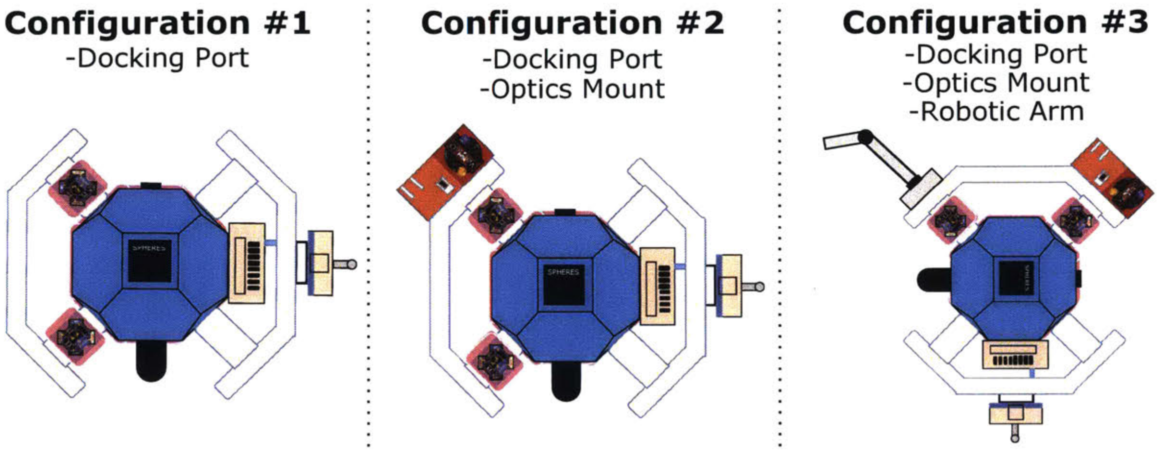


Figure 3-14: Hardware configurations used for MRAC/PD comparison.

All configurations have the SPHERES satellite, the VERTIGO Avionics Stack computer, and the Halo Expansion Device in common. Configuration #1 includes one Universal Docking Port as its only attachment, which is the physical configuration to which the baseline SPHERES PD controller will be tailored. Configuration #2 has, aside from the Docking Port from Config.#1, a stereo-vision camera optics mount attached to the Halo device, while Configuration #3 includes all of the former devices, in addition to a robotic arm. It is worth noticing that the size of the robotic arm, with respect to the rest of the assembly, is very comparable, a fact which greatly influences the inertia proportions of this latter configuration.

3.2.2.2 Algorithm and Implementation

Like the majority of the current control systems, SPHERES uses a discrete digital control approach for controlling its 12 actuators. Each control cycle is ran once per second, and consists in determining the thruster firing times as a function of the needed and calculated forces and torques. A high level overview of the SPHERES control loop is depicted in Fig. 3-15, in which the flowchart shows the position in which the adaptive control subroutine would be instantiated.

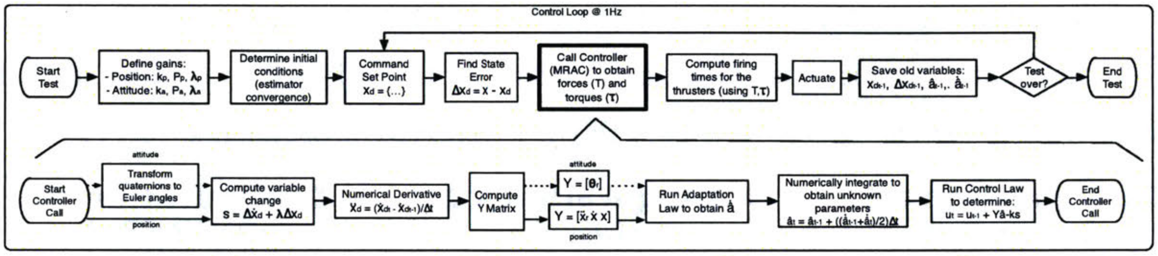

Control Loop @ 1Hz
Start Test
Define gains: - Position: k_p, P_p, λ_p - Attitude: k_a, P_a, λ_a
Determine initial conditions (estimator convergence)
Command Set Point $X_d = (...)$
Find State Error $\Delta X_d = X - X_d$
Call Controller (MRAC) to obtain forces (T) and torques (τ)
Compute firing times for the thrusters (using T,τ)
Actuate
Save old variables: X_{dt-1}, ΔX_{dt-1}, $\hat{a}_{t-1}$, $\dot{\hat{a}}_{t-1}$
Test over?
End Test
Start Controller Call
attitude
Transform quaternions to Euler angles
position
Compute variable change $s = \Delta\dot{X}_d + \lambda\Delta X_d$
Numerical Derivative $\dot{X}_d = (X_{dt} - X_{dt-1})/\Delta t$
Compute Y Matrix
attitude
$Y = [\theta_r]$
$Y = [\ddot{x}_r \dot{x}\, x]$
position
Run Adaptation Law to obtain $\dot{\hat{a}}$
Numerically integrate to obtain unknown parameters $\hat{a}_t = \hat{a}_{t-1} + ((\dot{\hat{a}}_{t-1}+\dot{\hat{a}}_t)/2)\Delta t$
Run Control Law to determine: $u_t = u_{t-1} + Y\hat{a}\text{-}ks$
End Controller Call

Figure 3-15: High level flowchart representing the overview of the SPHERES control cycle. Within the 1 *Hz* loop, the Model Reference Adaptive Controller subroutine is to be called to compute the warranted forces and torques given a desired state vector.

Thus, the presented continuous time formulation must be discretized and a sequential algorithm must be devised for its appropriate implementation. The steps followed by the programming code are shown in Algorithm 1, along with its helper functions, which are detailed in the following sections. SPHERES Test Projects run on an on-board DSP; all implementations were written using C programming language.

Algorithm 1 Discrete-time MRAC algorithm formulation.

```
procedure DT-MRAC
    $\mathbf{X}_k[\phi_k, \theta_k, \psi_k] \leftarrow$ quat2eul($\mathbf{X}_k$) # just for attitude
    $\tilde{\mathbf{X}}_k \leftarrow \mathbf{X}_k - \mathbf{X}_{d_k}$
    $\mathbf{s}_k \leftarrow$ computeSurface($\tilde{\mathbf{X}}_k, \lambda$)
    $\mathbf{Y}_k \leftarrow$ matrixFactY($\mathbf{s}_k$)
    $\dot{\hat{\mathbf{a}}}_k \leftarrow$ runAdaptationLaw($\mathbf{P}, \mathbf{Y}_k, \mathbf{s}_k$)
    $\hat{\mathbf{a}}_k \leftarrow$ getCurrentEstimates($\dot{\hat{\mathbf{a}}}_k, \dot{\hat{\mathbf{a}}}_{k-1}, \hat{\mathbf{a}}_{k-1}$)
    $\mathbf{u}_k \leftarrow$ getForcesAndTorques($\mathbf{s}_k, \mathbf{Y}_k, \hat{\mathbf{a}}_k, k$)
```

The following helper functions follow the mathematical equations derived throughout the MRAC formulation. It is assumed that the necessary variable handle is already being taken care of, and that the warranted parameters are available inside each call.

`computeSurface`$(\tilde{\mathbf{X}}_k, \lambda)$ The hypersurface calculation expressed by the variable change in Equation 2.13 is to be computed utilizing the constant gain λ and the state error vector, following the expression

$$\mathsf{s}[k] = \left(\begin{bmatrix} \dot{x}[k] \\ \dot{y}[k] \\ \dot{z}[k] \end{bmatrix} - \begin{bmatrix} \dot{x}_d[k] \\ \dot{y}_d[k] \\ \dot{z}_d[k] \end{bmatrix} \right) + \lambda \cdot \left(\begin{bmatrix} x[k] \\ y[k] \\ z[k] \end{bmatrix} - \begin{bmatrix} x_d[k] \\ y_d[k] \\ z_d[k] \end{bmatrix} \right).$$

`matrixFactY`(s_k) The form of the matrix of known quantities $\mathbf{Y}$ is ad-hoc and entirely dependent on the model employed for the MRAC's formulation. For a double integrator system, the factorization can be found in Equation A.4, while Equation 2.29 expresses the factorization warranted by the Clohessy-Witlshire dynamical system.

`runAdaptationLaw`$(\mathbf{P}, \mathbf{Y}_k, \mathsf{s}_k)$ After obtaining the matrix factorization, the rate of change of the parameter estimates is to be computed, following the expression

$$\dot{\hat{\mathsf{a}}}[k] = -\mathbf{P} \cdot \left(\begin{bmatrix} \ddot{x}_d[k] \\ \ddot{y}_d[k] \\ \ddot{z}_d[k] \end{bmatrix} - \lambda \left(\begin{bmatrix} \dot{x}[k] \\ \dot{y}[k] \\ \dot{z}[k] \end{bmatrix} - \begin{bmatrix} \dot{x}_d[k] \\ \dot{y}_d[k] \\ \dot{z}_d[k] \end{bmatrix} \right) \right)^T \cdot \left(\begin{bmatrix} x[k] \\ y[k] \\ z[k] \end{bmatrix} - \begin{bmatrix} x_d[k] \\ y_d[k] \\ z_d[k] \end{bmatrix} \right).$$

`getCurrentEstimates`$(\dot{\hat{\mathsf{a}}}_k, \dot{\hat{\mathsf{a}}}_{k-1}, \hat{\mathsf{a}}_{k-1})$ For the calculation of the current time step's parameter estimates, a numerical integration step needs to be carried out, using the values of previous time steps as well as the formerly computed rate of change. Making use of the trapezoidal rule of integration, one obtains

$$\hat{\mathsf{a}}[k] = \hat{\mathsf{a}}[k-1] + \frac{\dot{\hat{\mathsf{a}}}[k] + \dot{\hat{\mathsf{a}}}[k-1]}{2} \cdot \Delta t. \tag{3.1}$$

`getForcesAndTorques`$(\mathsf{s}_k, \mathbf{Y}_k, \hat{\mathsf{a}}_k, k)$ To finalize the Discrete-Time MRAC call, the corresponding forces and torques need to be computed. By following the derived control law shown in Equation A.6, the discretized version can be computed as

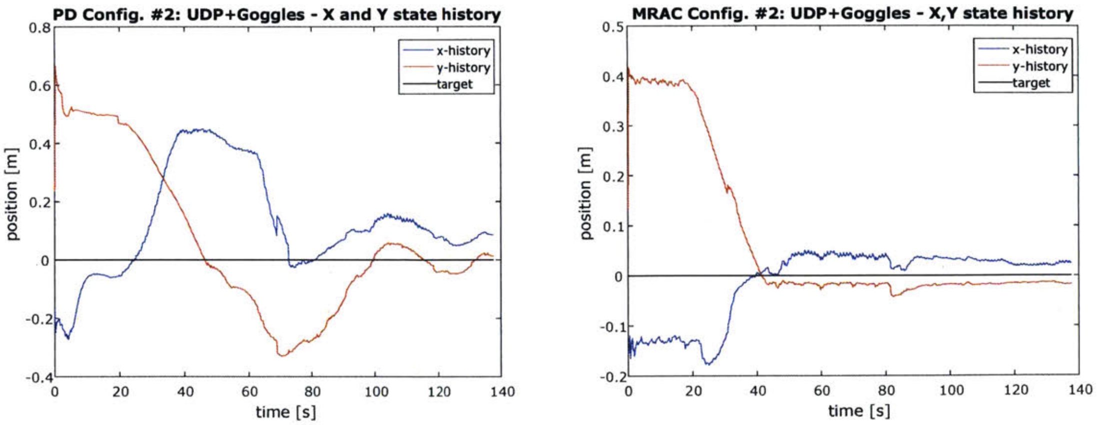
$$\mathbf{u}[k] = \left\{ \hat{m}[k] \begin{bmatrix} \ddot{x}_r[k] \\ \ddot{y}_r[k] \\ \ddot{z}_r[k] \end{bmatrix} + \hat{\alpha}[k] \begin{bmatrix} \dot{x}[k] \\ \dot{y}[k] \\ \dot{z}[k] \end{bmatrix} + \hat{\beta}[k] \begin{bmatrix} x[k] \\ y[k] \\ z[k] \end{bmatrix} \right\}$$
$$- k \cdot \left\{ \left(\begin{bmatrix} \dot{x}[k] \\ \dot{y}[k] \\ \dot{z}[k] \end{bmatrix} - \begin{bmatrix} \dot{x}_d[k] \\ \dot{y}_d[k] \\ \dot{z}_d[k] \end{bmatrix} \right) + \lambda \cdot \left(\begin{bmatrix} x[k] \\ y[k] \\ z[k] \end{bmatrix} - \begin{bmatrix} x_d[k] \\ y_d[k] \\ z_d[k] \end{bmatrix} \right) \right\}.$$

3.2.2.3 Results

After finalizing the algorithm's implementation, multiple runs for each controller, for each configuration, were performed in order to collect data sets, video recordings, and assess their performance difference. Graphical comparisons will be showed through phase plots and state histories, and performance metrics in terms of error characteristics will be determined using the compounded tests for each of the controllers (the full data set plots, videos, and comparison is shown in Appendix B).

A sample test run for both the baseline and the Model Reference Adaptive Controller using configuration#2 is shown in Fig. 3-16. A similar behavioral trend from both controllers is observed throughout the glass table testing session; large position and attitude overshoots exhibited by the baseline controller, and a slower, more precise control actuation from the adaptive controller's part that incurs lower tracking errors.

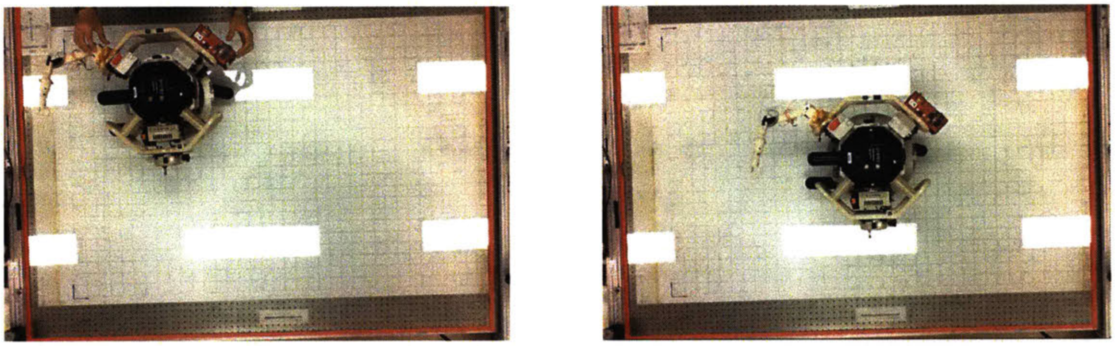
The most drastic difference between the two controllers' performance is shown while using configuration#3, which includes a docking port, the stereo-vision optics mounts, and a robotic arm. Arguably the hardest of the three, given the non-negligible change in the inertia properties induced by the addition of the robotic arm, the test clearly shows how unstable can gross uncertainties in model parameters render a traditional PD type of algorithm; even though the baseline controller is able to stop the spinning motion and counteract it, it was not able to exhibit any converging



(a) Config.#2 baseline xy-states. (b) Config.#2 MRAC xy-states.

Figure 3-16: X and Y state histories for Configuration #2 under (a) SPHERES PD controller and (b) MRAC controller.

stability properties given the alloted test time. This behavior is shown in Fig. 3-17(a).



(a) Baseline (b) MRAC

Figure 3-17: Baseline PD controller and MRAC test with hardware configuration #3 — Halo Expansion device with a docking port, the VERTIGO stereo-cameras, and a robotic arm. (Video speed: ×10)

Quite conversely, the story for the adaptive controller throughout configuration #3's trial run turned out to be entirely different, as seen in Fig. 3-17(b). Translation is again achieved with almost no attitude error, and position holding is initially accurately kept at a steady point. After some instants, it seems as if the weight and inertia of the robotic arm start inducing a spin in the yawing axis; the satellite tries to compensate for the error but it cannot completely, and then starts to slowly diverge from the set point. Despite that, the performance difference in comparison to the baseline PD controller is quite significant, yielding its results to be satisfactory. More importantly, notwithstanding the fact that **zero prior knowledge** about the physical properties of the system was given, a stable system was observed.

The complete plots detailing the x- and y-state histories, the attitude control performance (yaw angle), and the individual phase plots corresponding to each of the tests, for each configuration, can be found in Appendix B.

By merging all phase plots into a single graph, the performance difference between both controllers can be quickly and pictorially seen. Shown in Fig. 3-18, a comparison of all hardware configuration runs for both controllers is presented. The dashed lines represent the baseline PD controller trajectories, while the solid ones depict the MRAC's trajectories.

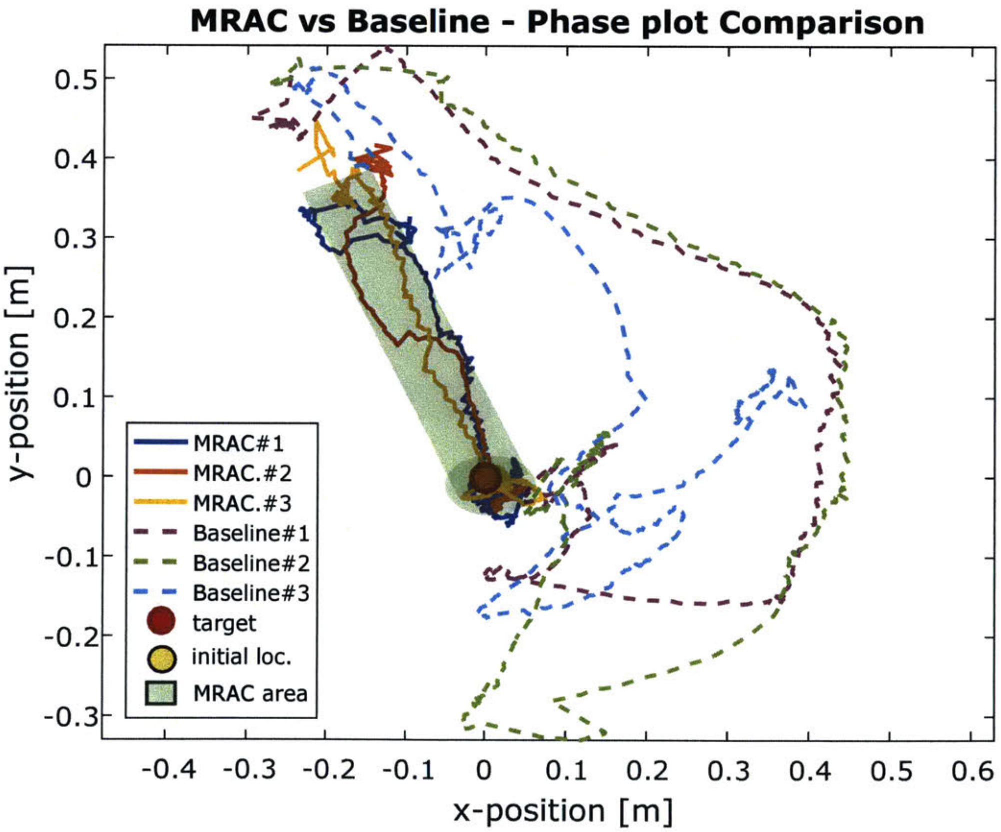


Figure 3-18: Phase plot comparison between all configuration runs for both controllers. The green shaded regions represents the area in which the MRAC was able to contain the majority of the trajectories.

A noticeable trend followed by each controller can be seen; the baseline controllers tend to travel initially towards the positive x-axis, always overshooting its target and needing to subsequently correct the error by curving around, and position-holding close to the origin (except for the baseline's arm run). In contrast to this, the MRAC tends to command to the satellite forces and torques that take it in a much straighter direction, tending not to overshoot neither in the x- nor the y-axis direction. In fact, a shaded region encompassing the MRAC's trajectories can be overlaid on top of the phase plot, more or less indicating the common operation region taken by the controller, for any hardware configuration.

A similar comparison for the attitude control performance in the yaw angle is presented in Figure 3-19. Again, the solid lines represent the z-axis rotation history exhibited by the tests using the Model Reference Adaptive Controller, while the dashed lines denote the baseline PD controller.

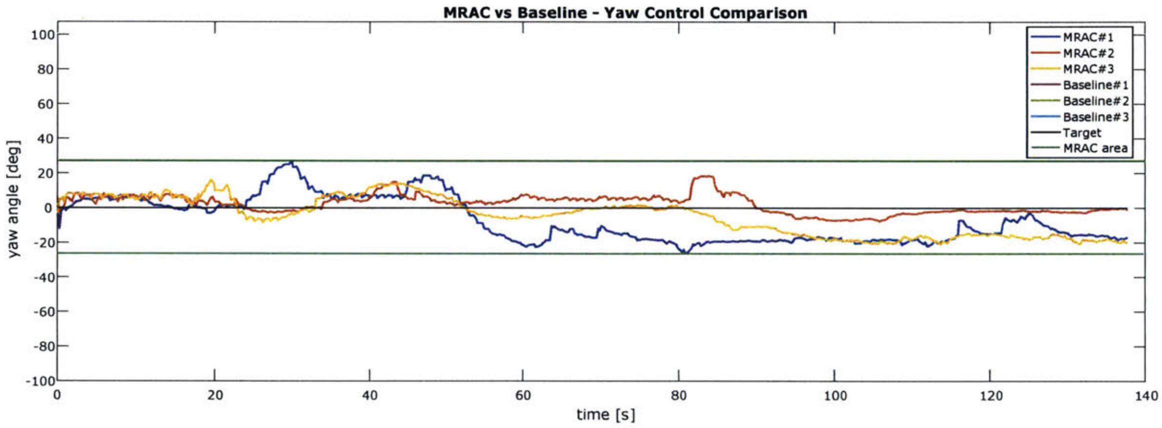


Figure 3-19: Attitude control performance comparison between all configuration runs for both controllers. The green lines encompass the region in which the MRAC was able to maintain the yaw angle, for all configurations.

Although the errors are definitely not negligible, the MRAC tracking performance largely surpasses the baseline controllers'. It can be seen that the yaw tracking error for the adaptive approach is kept within some bounds delineated by the green lines, and that smaller and less energetic oscillations are present throughout the tests. A steadier trend in the yaw history is observed for the MRAC, especially after the translation-induced transients have passed; towards the end of the tests, the histories for the yaw angles tend to flat-line for the adaptive controller, meaning that the

Table 3.1: Position and attitude tracking error statistics computed for the SPHERES baseline PD controller's compendium of glass table tests.

Baseline Controller Error Statistics			
	mean - μ	std - σ	var - σ^2
X-Pos Ctrl	9.92 cm	19.18 cm	368 cm^2
Y-Pos Ctrl	11.61 cm	24.76 cm	613 cm^2
Yaw Control	-1.23^o	30.92^o	956$^{o^2}$

Table 3.2: Position and attitude tracking error statistics computed for the adaptive approach compendium of glass table tests using an MRAC control law.

MRAC Controller Error Statistics			
	mean - μ	std - σ	var - σ^2
X-Pos Ctrl	-2.2 cm	7.94 cm	64 cm^2
Y-Pos Ctrl	7.99 cm	15.17 cm	230 cm^2
Yaw Control	-3.01^o	10.77^o	116$^{o^2}$

satellite was able to attitude-hold.

Using the bulk of information acquired from the glass table tests, statistical measures for the tracking errors are able to be computed. The data corresponding to the baseline PD controller is shown in Table 3.1, while the one corresponding to the adaptive approach can be found in Table 3.2.

Summing-up The implementation and glass table testing results obtained indicate that a move towards less conventional control techniques, such as an adaptive and learning approach, appears to be quite promising. Even when the PD controller was expected to outperform the adaptive approach for the test run with hardware configuration #1 (since it was entirely tailored for those specific physical properties), the MRAC was able to yield smaller tracking errors and an overall more acceptable performance.

The implications these results have are very important, since they indicate that the knowledge of the system's actual true parameter values are not as well known as expected. These large uncertainties governing the system's operations and the control authority that the algorithms are able to exert upon it are the ones that this work intends to lessen, in order to be able to reach the performance requirements warranted by space missions such as autonomous docking and on-orbit robotic assembly.

3.3 Summary

An in-depth motivation through InSPIRE-II test sessions' analysis was firstly introduced to elucidate on the main problems encountered during on-orbit operations. In order to overcome the uncertainties that govern the system, an MRAC control approach was first analyzed through simulation to assess its learning capabilities and subsequently implemented on ground hardware to perform a direct comparison to the baseline SPHERES PD controller. The obtained results show promise in terms of increasing tracking performance, and are further analyzed in the next section to determine the best way to leverage them for working towards reliable autonomous in-space robotic assembly operations.

Chapter 4

Probabilistic Operation through Composite Adaptation

Based upon the conclusions drawn from the aforementioned analyses, and to determine how the learning approach impacts the controllability of the system, a probabilistic approach for assessing the control performance level of a system will be introduced. A metric to gauge the effectiveness of control algorithms employed for in-space robotic assembly that allows for a more incremental development cycle is proposed, and preliminary results for its utilization are shown.

Additionally, the limitations for operating within a stochastic environment exhibited by the MRAC approach are shown through simulated examples, and an improvement upon the algorithm that would allow for a better adaptation behavior is formulated.

4.1 Data Analysis through Probability Models

It is unnatural to think of the docking problem in a deterministic sense, since it will never be possible for neither the chaser nor the target satellites to be able to perfectly hold their pointing directions towards a single and corresponding mutual location, and then just simply thrust to finish the docking maneuver. Hence, a probabilistic approach on the analysis of a satellite's control performance can be taken, in terms of assigning a metric that, on expectation, would allow one to describe the likelihood for an objective to be achieved, such as autonomous docking.

By having some specifications describing the characteristics of a system, i.e., physical properties, performance capabilities, etc., it is possible to develop a set of requirements that specify the attributes warranted for the success of a certain objective. This deterministic group of statements can then be translated into a probability model that accounts for the variation that the system is able to tolerate and still achieve a mission success outcome.

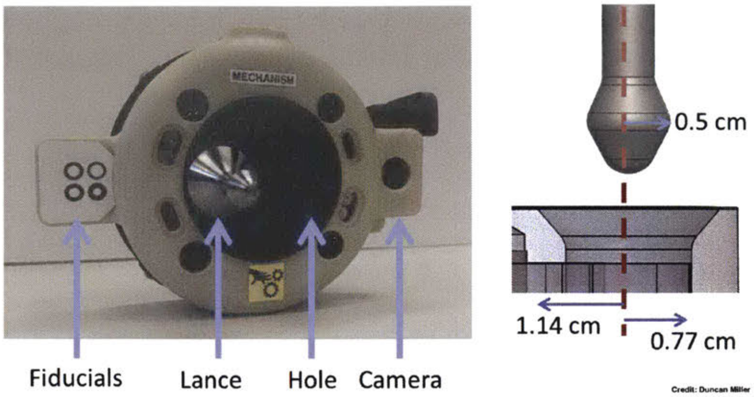
Concurrently, data regarding the controllable aspect of the system is to be analyzed, and the useful information that is able to be gathered needs to furthermore be fitted to a second probability model. This allows for some probabilistic comparison to be performed upon the two models, and a metric to be computed given the two pieces of information.

4.1.1 Attitude Control Performance Analysis for InSPIRE-II

Utilizing the physical characteristics of the SPHERES docking ports, shown in Fig. 4-1, an analysis can be performed to determine the off-nominal situations under which the docking maneuver can still be successful. An example of these tolerable errors intrinsic to the system would be the capacity of successfully accomplishing a docking maneuver even when a SPHERES docking port comes in slightly tilted, or with an off-centered position.

From this characterization, maximum error bounds on the satellites' state vector can be computed. For the attitude control of the spacecraft, a value for the maximum angle deviation allowed (θ_{max}) for the chaser to still perform a successful docking maneuver can be calculated to be $\theta_{max} \approx 2.5^o$. This implies that, at the time of docking (while position-holding in front of the target), the chaser's attitude controller performance should yield errors of less than 2.5^o in all axes (for the full derivation, please refer to Appendix C).

The data utilized for the elaboration of the probability models for both the target and the attitude error distributions consists of over 4880 data points spanning around



Credit: Duncan Miller

Figure 4-1: The SPHERES Universal Docking Port (UDP) along with the physical dimensions of the lance and hole pair.

five to six test runs, per satellite, per test session.

Target Distribution By reasoning about the problem in a stochastic manner, this latter requirement can be posed as a Gaussian probability density function (pdf); instead of using a strict bound, probability masses can be employed. Hence, a Gaussian pdf with a $3\sigma = \theta_{max}$ parameter can be utilized in lieu of the requirement. By setting the θ_{max} value to be three times the normal distribution's standard deviation (σ), it is being specified that at least 99.7% (basically ALL) of the attitude error performance data points obtained should be lower than said value. This would yield a probability model encoded with the requirements and described by

$$\text{target distribution} :\sim \mathcal{N}\left(\mu = 0, \sigma = \frac{\theta_{max}}{3}\right).$$

Attitude Error Distributions In order to perform a comparison against the target distribution, probability models of the system need to be generated. For this, test data or real time data can be analyzed to determine the statistics behind the performance of a spacecraft. In the InSPIRE-II case, given the observed error between the attitude tracking estimates and references in each of the three degrees of freedom, a probability distribution is able to be determined for the roll ϕ, pitch θ, and yaw ψ control of the two ensembles (SPHERES + VAS + UDP and SPHERES + Halo and friends).

Table 4.1: Attitude tracking error statistics for all 3-Axes computed for the Blue SPHERES during both Test Session 86: Halo Checkout and Test Session 92: Docking Port Science 1.5.

Blue SPHERES Attitude Control Error Statistics

		mean - μ	std - σ	var - σ^2
UDP	Roll Error ($\Delta\phi$)	-2.2 cm	7.94 cm	64 cm^2
	Pitch Error ($\Delta\theta$)	7.99 cm	15.17 cm	230 cm^2
	Yaw Error ($\Delta\psi$)	-3.01^o	10.77^o	116$^{o^2}$
Halo	Roll Error ($\Delta\phi$)	-2.2 cm	7.94 cm	64 cm^2
	Pitch Error ($\Delta\theta$)	7.99 cm	15.17 cm	230 cm^2
	Yaw Error ($\Delta\psi$)	-3.01^o	10.77^o	116$^{o^2}$

Table 4.2: Attitude tracking error statistics for all 3-Axes computed for the Orange SPHERES during Test Session 86: Halo Checkout. No useful data was acquired during Test Session 92: Docking Port Science 1.5.

Orange SPHERES Attitude Control Error Statistics

		mean - μ	std - σ	var - σ^2
Halo	Roll Error ($\Delta\phi$)	-2.2 cm	7.94 cm	64 cm^2
	Pitch Error ($\Delta\theta$)	7.99 cm	15.17 cm	230 cm^2
	Yaw Error ($\Delta\psi$)	-3.01^o	10.77^o	116$^{o^2}$

Thus, one probability model for each SPHERE (Blue and Orange) for each configuration (Halo and UDP) should be able to be obtained. Unfortunately, there were no successful nor acceptable position-holding tests for the Orange SPHERE during TS92, so no model was able to be calculated for the Orange SPH + UDP hardware configuration. The error statistics for the three probability models are captured in Tables 4.1 and 4.2.

Comparison and Objectives After obtaining the probability models, the comparison with the target distribution can be performed. It is important to remember that what the target distribution expresses is the area on which the majority of the collected attitude data points prior to the docking maneuver (e.g., position holding right in front of the target) must lie in order to probabilistically say that the docking maneuver will succeed. It is completely dependent on the mechanical properties of the docking ports and the Halo and UDP configuration (distance from GC to lance tip, hole dimension, etc).

The comparison between the Blue SPHERES models and the target distribution is shown in Fig. 4-2, while the one for the Orange SPHERES can be found in Fig. 4-3. In this case, wider distributions signify a poorer performance (larger variance), meaning that the spread of the tracking error points is larger. Additionally one would want for the mean tracking error to be identical to zero, but offsets can certainly be observed in the data results.

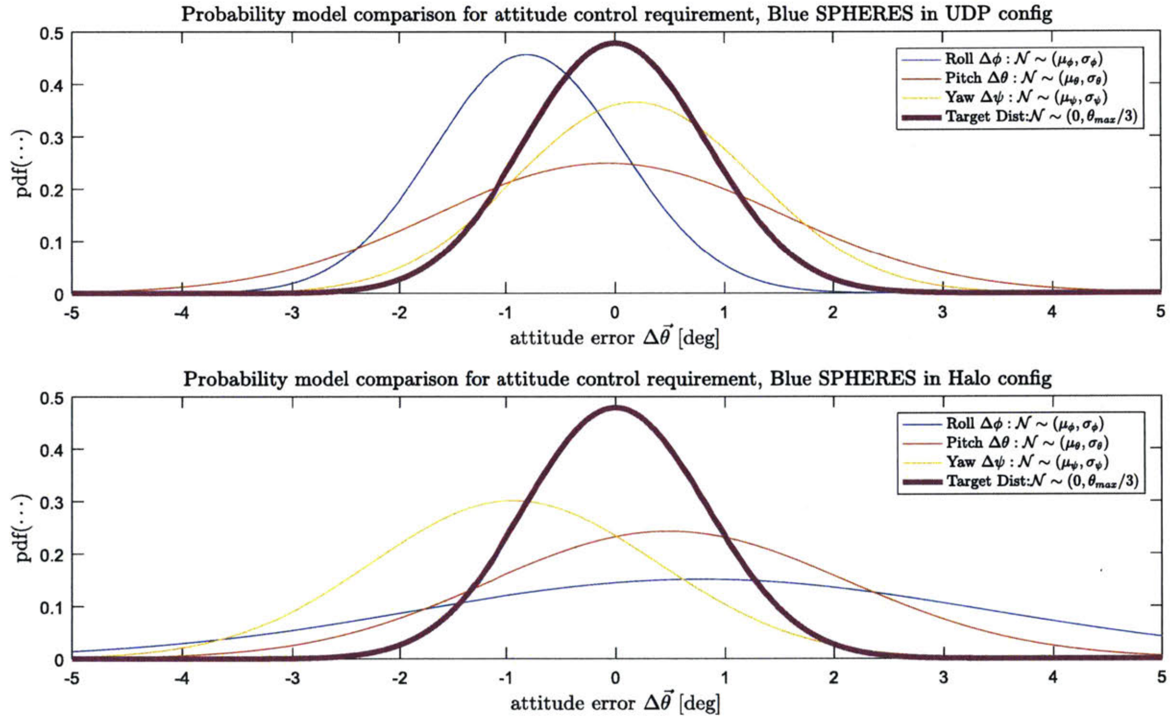


Figure 4-2: Probability model comparison for the Blue SPHERES attitude control performance in its UDP (top) and Halo (bottom) configuration.

The comparison between the probability models obtained for the Orange SPHERES is shown in Fig. 4-3, in which only the Halo configuration is plotted. From these results, it is quite interesting to observe that the roll ϕ control distribution for the Halo configuration is almost identical to the target distribution, not only in width (variance), but also in its center point ($\mu_\psi \approx 0^o!$). This signifies that the Orange satellite should, in expectation, have no problem with the docking maneuver in terms of controlling its roll ϕ attitude, since it accurately satisfies the derived requirement.

This result is very informative, since this same attitude axis could not be correctly controlled, at any moment, by the Orange and UDP configuration during TS92. The

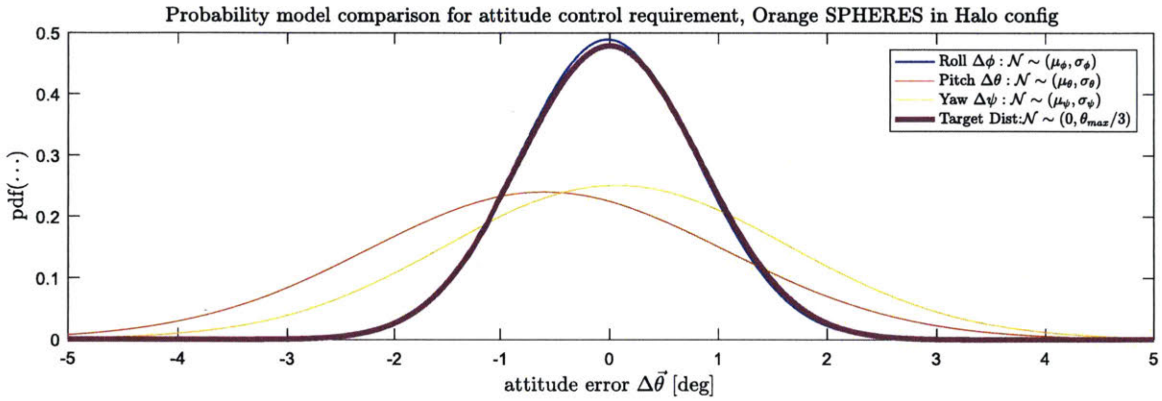


Figure 4-3: Probability model comparison for the Orange SPHERES attitude control performance in its Halo configuration; no useful UDP configuration data was obtained from TS92.

possibility that Orange's actuation capabilities were the culprit for the instabilities observed during this latter test session can be ruled out, leaving as possible failure causes both the poor system characterization of the ensemble and the control and path planning algorithms.

Another aspect that can be easily assessed is the difference in degree of controllability between the Halo and UDP configuration. For this, it is particularly informative to contrast the Orange SPHERES probability models to the ones for the Blue SPHERES. The comparison is shown in Fig. 4-4.

Just by inspection, it can easily be seen that the control authority exerted upon both systems is very comparable; in the roll ϕ direction, the Halo system is in fact much better controlled than the UDP ensemble. The pitch θ angle is in both cases the worst controlled degree of freedom, represented by a very wide distribution; for the UDP system, the probability model appears to be more aligned with the target distribution, meaning that the mean $\Delta\theta$ errors for the Blue satellite are closer to zero. The only clear difference is noticed in the yaw ψ angle, in which there is a better performance exhibited by the UDP system.

Overall, it is not particularly clear which configuration is more controllable, although it seems that the UDP system might be marginally better behaved. This assessment, paired with the fact that for TS86b no previous system identification tests were performed for the Halo system (best estimate parameter values from CAD

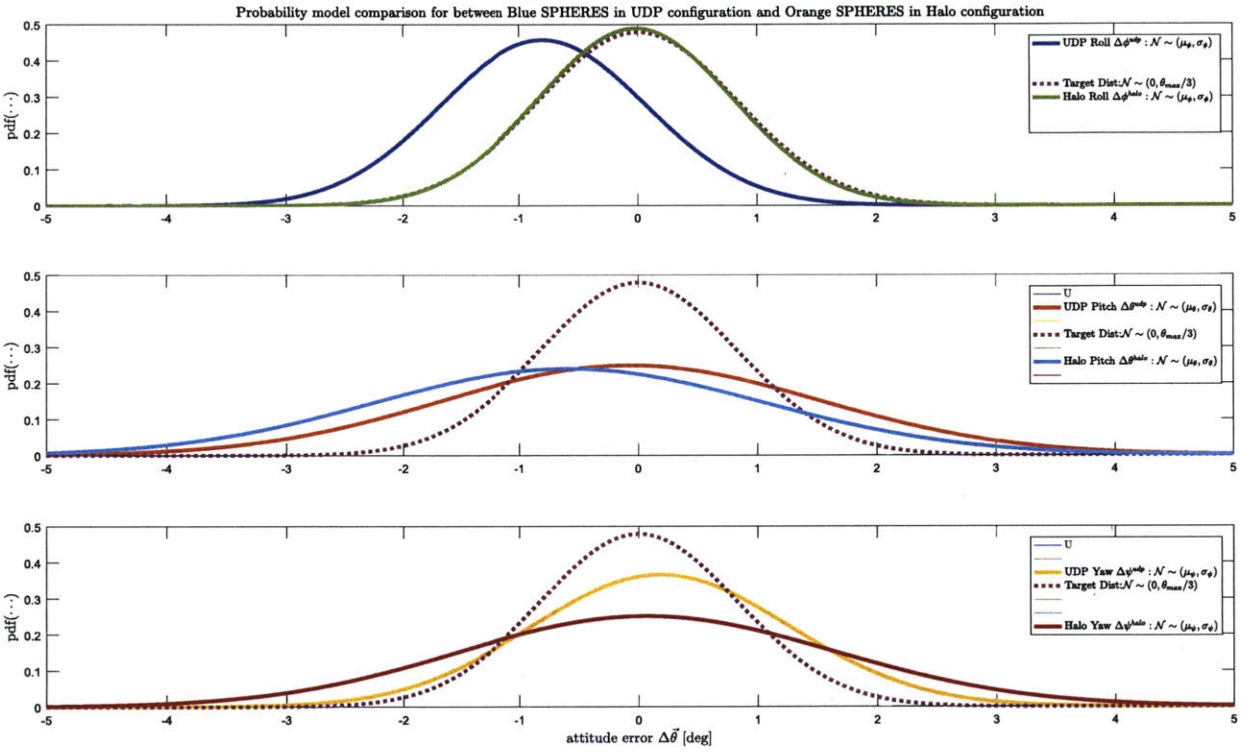


Figure 4-4: Probability model comparison between the two hardware configurations, in all three attitude degrees of freedom; the Blue SPHERES plus UDP configuration is assessed against the Orange SPHERES plus Halo configuration.

were used for inertia, mass, center of mass, etc, as no better guesses were available) allows for the conclusion that the Halo ensemble shows more promise in terms of performance improvement. This follows ones intuition, as the Halo ensemble's mass is much more evenly distributed along the XY-plane, and even though more massive, it can likely be rendered much more controllable than the UDP ensemble.

Summing-up A clear takeaway from this approach is the fact that if some guarantees on the likelihood of docking success are sought to be given, then an increase in control performance is warranted. In terms of the probability models, this means the reshaping of the distributions in order to increase a certain degree-of-freedom metric score. An approach with preliminary results on how to accomplish this aspect is proposed in this next section.

4.1.2 Likelihood Accretion through Learning

Through the analysis of test data from the InSPIRE-II ISS sessions, the observed lack of control authority upon the satellites was attributed to be due to the poor parameter estimates of our plant's model. This means that the level of uncertainty surrounding the physical properties of the system such as the center of mass, moments and products of inertia, ensemble mass, etc., are hindering the satellites capabilities of achieving the mission objectives. The fact that the system can't be characterized as accurately enough as demanded by traditional control methods' requirements warrants for solutions to be sought elsewhere; an adaptive approach is shown to yield promising results on this issue.

In order to overcome these problems, an adaptive approach in which the satellites gradually learn the unknown plant's parameters through control theory and estimation methods is proposed. It is then claimed that if on-board learning is performed, an increase on the likelihood of successfully achieving a docking maneuver is to be observed. This is denoted as likelihood accretion through learning.

Preliminary results from glass table testing. As compared to the InSPIRE-II analysis, this time no target distribution is used, since no requirements were flowed-down given the nature of the test, and sheer comparisons between each degree-of-freedom performance will be assessed for the two control approaches. Probability models will be generated with the statistics calculated from the glass table test runs shown in Tables 3.1 and 3.2.

Results for the x-position performance models between the baseline controller and the MRAC are shown in Fig.4-5. The fact of operating under gross uncertainties in terms of the physical properties of the system definitely affects the performance of the baseline PD controller. The noteworthy part is how, through on-board learning, the probability model undergoes a reshaping transformation which yields a lower-variance Gaussian distribution. Since our interest lies on the 3σ-probability mass spread and location, the fact that this type of behavior is observed is remarkable, since the increase in certainty implies, for instance, an increment in the likelihood of, in expectation, achieving a successful docking maneuver.

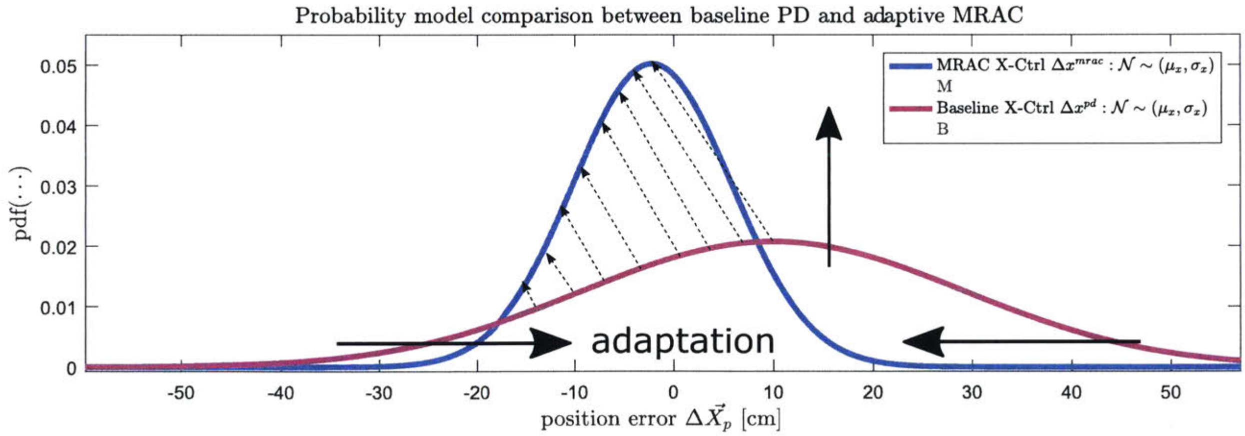


Figure 4-5: Glass table control performance on the x-position is shown for both the PD controller and the MRAC; it is apparent that by being able to adapt to the unknown mass and inertia parameters, a distribution with less variance is obtained.

Similarly, for the y-position control performance, the same type of behavior is obtained. The results for this degree-of-freedom are shown in Fig. 4-6. It's clear that not only is the certainty of the probability model increased, but also its mean is shifted towards the origin, which is the desired transformation to be obtained (a zero-mean error performance is the goal). Finally, and perhaps the most drastic, the increase in attitude control performance about the yaw ψ axis is shown in Fig. 4-7.

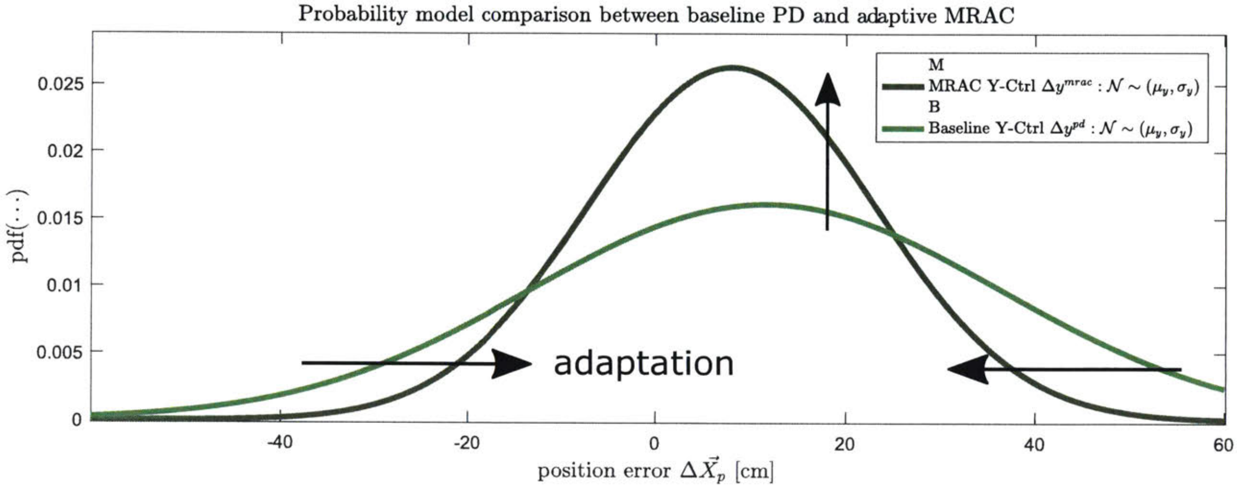


Figure 4-6: Glass table control performance on the y-position is shown for both the PD controller and the MRAC.

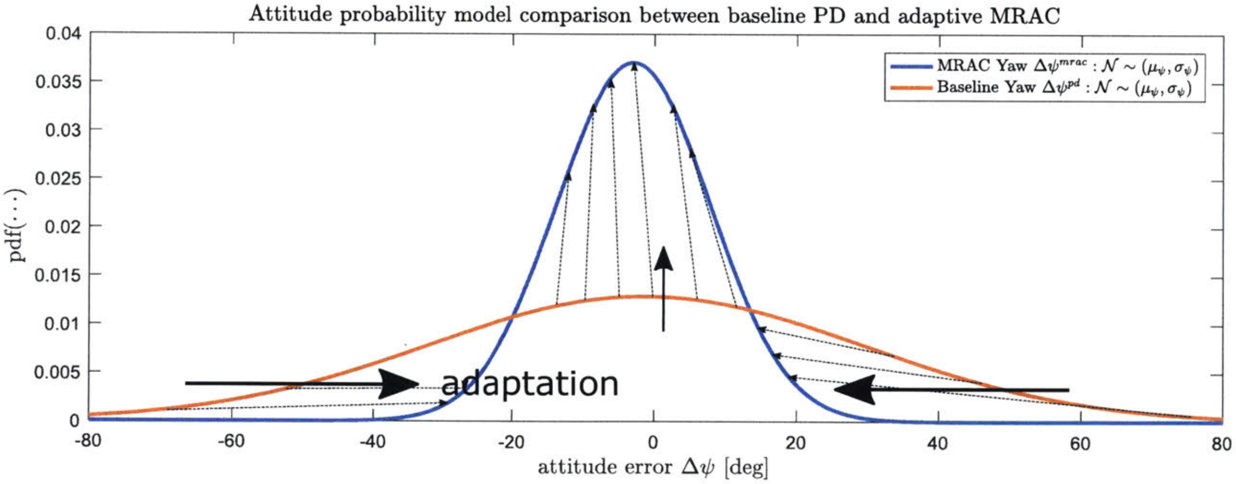


Figure 4-7: Distribution reshaping of the attitude control performance of a SPHERES satellite with uncertain mass and inertia properties; through the use of a paired adaptive and control law, better tracking performance is yielded, and thus an increase in the likelihood of successfully achieving the mission's objectives.

Summing-up Thus, an approach for incrementally analyzing and obtaining a performance metric given a set of requirements was shown. With it, a heuristic that expresses, in expectation, the likelihood of achieving a particular mission objective was introduced. Finally, through actual hardware implementation and glass table testing, it was demonstrated how an adaptive approach on control strategies employed by systems working under non-negligible parameter uncertainties is beneficial in terms of increasing the plant's tracking performance. By piecing everything together, an algorithm that allows the system to reason about its performance can be introduced; since autonomous docking is the main objective, the general algorithm is tailored to handle this problem.

4.1.3 Probabilistic Docking Algorithm

By considering the entire autonomy cycle and using the probabilistic metric at a higher level, rather than just at the lower control level, the information about the current performance of the system obtained from the probability assessment could potentially be utilized within algorithms such as the path planner, the scheduler, and a higher-level decision making executive, such as an on-board feasibility examiner.

The proposed structure for the algorithm can be shown in Algorithm 2, where the control system is included as the last step of the autonomy loop cycle. An overview on the mechanics of the functions utilized within the algorithm are provided in this section.

Algorithm 2 Probabilistic docking algorithm for autonomous operations.

```
procedure ProbDock
    targetDist $\sim \mathcal{N}(\mu^*, \sigma^*) \leftarrow$ computeTargetStats($\cdots$) # Tim's work?
    plantModels $\leftarrow \{\varnothing\}$
    for $d \in$ {degrees of freedom} do
        pdf($d$) $\sim \mathcal{N}(\mu_d, \sigma_d) \leftarrow$ analyzePerfmData($\cdots$)
        plantModels $\leftarrow$ plantModels $\cup$ pdf($d$)
    # obtain success likelihood and heuristic
    $\hat{\boldsymbol{\ell}}, \hat{h} \leftarrow$ CounselorModel(targetDist,plantModels)
    plan,actions $\leftarrow$ Planner($\hat{\boldsymbol{\ell}}, \hat{h}$)
    $\mathbf{u}, \hat{\mathbf{a}} \leftarrow$ runAdaptiveMethods(plan,actions)
```

`computeTargetStats(···)` The objective of this function is to compute a probability model of the target satellite by accounting for the mission objective requirements, the system's characteristics (both chaser and target), as well as the history of the available states of the target. This procedure was briefly demonstrated above in the calculation of the attitude performance target distribution, and the complete analysis steps are shown in Appendix C.

`analyzePerfmData(···)` The aforementioned probability models regarding the performance of the chaser satellite are determined by analyzing the state histories throughout the mission operation phases. Depending on the requirements driving the mission objectives', distinct analyses can be performed utilizing the real-time estimate of the state vector; in the previously shown example for attitude control performance analysis, position-holding roll, pitch, and yaw histories were employed for the computation of performance statistics, which were then translated into probability models.

`CounselorModel(targetDist, plantModels)` An analysis tool denoted as the «Counselor Model », whose objective is to inform how well the system is performing with respect to the mission objectives, takes as its arguments the target distribution and the plant models in order to compute a success likelihood metric. The full model derivation is shown in Appendix C.

`Planner(`$\hat{\ell}, \hat{h}$`)` In order to perform a fully autonomous mission and be able to cope with the stochasticity of operating within a changing environment, an on-board planning and scheduling algorithm needs to be employed. By passing the success likelihood metric and a heuristic as arguments to this internal algorithm, a better informed plan and set of subsequent actions can be generated to increase the success likelihood of achieving the docking maneuver.

`runAdaptiveMethods(plan, actions)` After computing the plan and the set of actions to be carried out, the control and learning algorithms, such as the Model Reference Adaptive Controller through its control and adaptation laws, translate these semantic representations into quantitative information using the mathematical framework laid in ch. 2.

The proposed probabilistic docking algorithm leverages both the adaptive control techniques presented in the previous section and the probabilistic analysis tool described in the current chapter. This algorithmic approach towards the autonomy cycle of a system allows for graph search ideas and techniques, such as the computation of optimal policies (e.g., LQR-Trees [64]) or informed search approaches (e.g., randomized A* [65]), to be employed in conjunction with the proposed learning methods.

Although promising results have been shown thus far, both through simulation and hardware implementation, it is shown bclow that the performance of the proposed learning methods is hindered by the stochastic nature of a real environment. This characteristics will be further analyzed, and a method for enhancing the solutions is proposed.

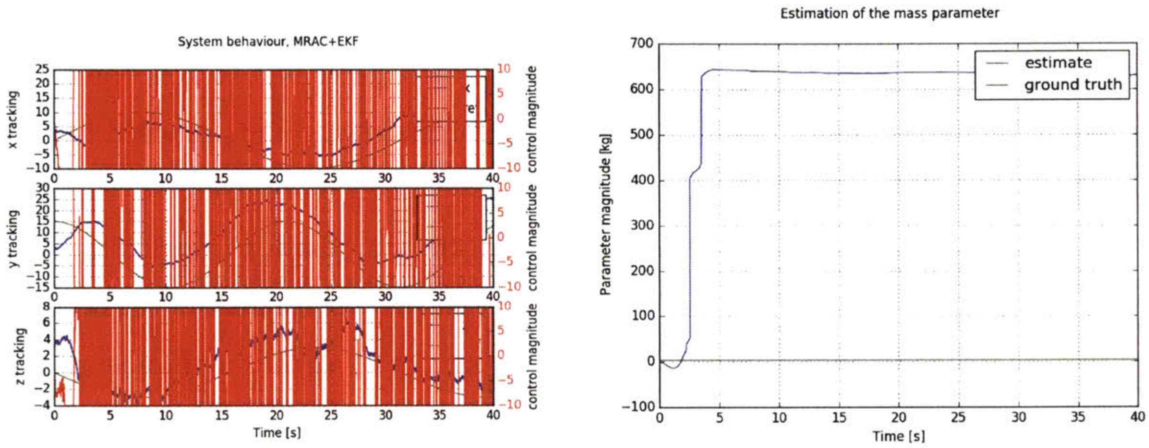
4.2 Stochastic Operation via Composite Adaptation

The transition of the adaptive approach from a deterministic framework to a stochastic environment is analyzed through simulations. A strategy to deal with the limitations arising from said change is introduced, and an augmented solution to the problem presented. Finally, simulations are employed to assess the learning behavior of the augmented algorithm's new formulation.

4.2.1 MRAC's Limitations in Stochastic Environments

To analyze the MRAC's behavior under imperfect and noisy conditions, a stochastic process model and measurement model will be considered, just as described throughout the problem formulation found in section 2.1.

If no modification is to be performed upon the deterministic version of the MRAC controller, then results such as the ones shown in Fig. 4-8 are to be expected. It is clear from Fig. 4-8(a) that the control input is severely suffering from a chattering type of response, in which the actuators are constantly and rapidly being saturated in both directions. As for the adaptation process, Fig. 4-8(b) shows a tremendous divergence in the estimation of the mass parameter value; this “explosive” response is known as a burst [66], and in this case is due to the effect of the measurement noise characteristics and imperfect state feedback on the system’s derivative properties.



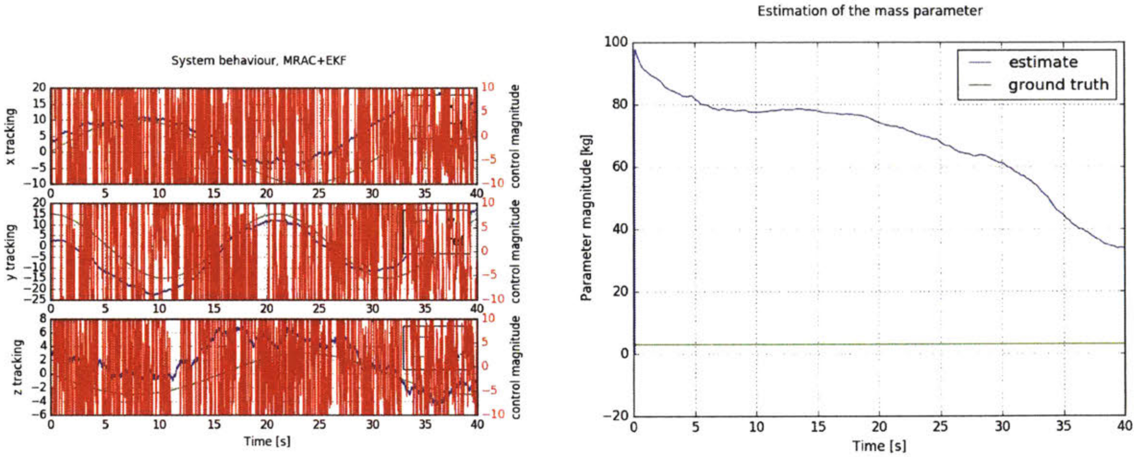
(a) State, reference and control input histories (green = reference, blue = states).

(b) Estimation history for the mass parameter adaptation ($\hat{m}_0 = 0\ kg$, true value $m^* = 3\ kg$).

Figure 4-8: Stochastic MRAC run #1 (noisy full-state feedback): bursting conditions caused by the stochastic nature of noisy (real) measurements rendering a system unstable; the mass estimate diverges by several orders of magnitude, and settles at an incorrect and large value. (Initial estimate $\hat{m}_0 = 0\ kg$, true mass parameter $m^* = 3\ kg$)

Although bursting conditions render the parameter estimates uninformative, the system will still exhibit a bounded response to the controller’s input, as is shown in Fig. 4-8. In this case, the tracking error cannot really be minimized, and only the reference signal’s trend is able to be tracked (sinusoidal shape). Thus, the MRAC guarantees offered by a full-state feedback approach, even under sufficiently rich conditions (the tracking error is never eliminated in this case), are not directly transferable to the stochastic case.

To overcome these adversities, the adaptation gain matrix parameter $\mathbf{P}_p$ introduced in Eq. A.9 is to be modified to control the rate of adaptation to be exerted upon the parameters under this uncertain conditions. One such example is shown in Figure 4-9, in which a slower adaptation rate is chosen for the mass estimate as compared to the simulation shown in Fig. 4-8.



(a) State, reference and control input histories (green = reference, blue = states).

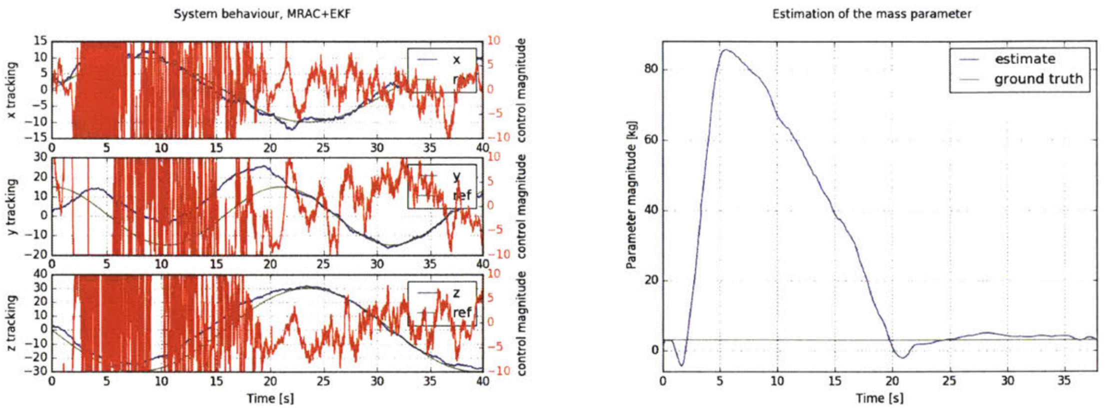
(b) Estimation history for the mass parameter adaptation ($\hat{m}_0 = 0\ kg$, true value $m^* = 3\ kg$).

Figure 4-9: Stochastic MRAC run #2: trial run showing a slow but correct adaptation estimate correction; this was achieved after tuning the adaptation gain matrix following the experiment shown in Fig. 4-8 (stochastic MRAC run #2). Initial estimate $\hat{m}_0 = 0\ kg$, true mass parameter $m^* = 3\ kg$.

From Fig. 4-9(b), it can be seen that the bursting behavior is still present at the beginning of the simulation, but as the controller performs the tracking maneuvers, the parameter estimate starts to slowly tend towards the ground truth. Nonetheless, the control action exerted upon the system — shown in red in Fig. 4-9 — still suffers from extreme chattering jolts, dramatically saturating the actuators in both directions throughout the duration of the test.

If the adaptation rate tuning is further refined — ad-hoc variations to the weighting matrix $\mathbf{P}_p$ to adhere to the desired actuator bandwidth — then smoother results are to be obtained. Showcasing such an effort is the simulation presented in Fig. 4-10.

Comparing this simulation to the previously shown ones offers quite a contrasting difference in terms of results. It is evident that given the parameter estimate value recovery towards the ground truth region shown in Fig. 4-10(b), the control input is able to avoid the saturation limits while simultaneously reducing the chattering frequency, which is mainly given by the sensors' and filter's characteristics. The burst is able to be contained within an order of magnitude of deviation from the true



(a) State, reference and control input histories (green = reference, blue = states).

(b) Estimation history for the mass parameter adaptation ($\hat{m}_0 = 0\ kg$, true value $m^* = 3\ kg$).

Figure 4-10: Stochastic MRAC run #3: trial run to be compared against the one shown in Fig. 4-9, showing a faster adaptation estimate correction and an eventual (almost) true value convergence; similarly, by modifying the adaptation gain matrix, this results are obtained. Initial estimate $\hat{m}_0 = 0\ kg$, true mass parameter $m^* = 3\ kg$.

parameter, and an overall better behaved response is apparent.

Nevertheless, the actuator bandwidth demands, the sudden parameter estimate bursts, and the poor tracking errors are still very much of a concern when the requirements of space missions are considered. Due to this limitations, further exploration into theoretical improvements to bound and contain the bursts will be performed in upcoming sections.

4.2.2 Parameter Identification through Kalman Framework

In order to probabilistically deal with the added signal uncertainties, a filtering approach was devised to optimally merge the measurement information with the dynamics process model output. Not only is the state able to be estimated, but an augmentation of the state vector with the model parameters of interest can be performed in order to estimate its value through the parameters' correlation with the rest of the state space. This additional source of information is then exploited to perform parameter identification upon the system; this second estimate of the parameters is then jointly used with the learning controller's output to stabilize the adaptation behavior during stochastic operations.

4.2.2.1 Discretization and State Augmentation

To perform the parameter identification step, an augmented Kalman Filter (AKF) is designed and implemented, augmenting the state vector with the mass of the satellite. Given the nature of the implementation, a first order discretization upon the laws of motion is performed, obtaining

$$\begin{aligned} \mathbf{v}^{(n+1)} &= \mathbf{v}^{(n)} + \frac{1}{m}\mathbf{F}^{(n)}\, dt, \quad \text{and} \\ \mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} + \mathbf{v}^{(n+1)}\, dt. \end{aligned}$$

Considering a discretization time step of Δt, the difference equations governing the system dynamics take the form

$$\begin{aligned} \mathbf{X}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \dot{z}_{k+1} \end{bmatrix} &= \begin{bmatrix} x_k + \dot{x}_{k+1} \cdot \Delta t \\ y_k + \dot{y}_{k+1} \cdot \Delta t \\ z_k + \dot{z}_{k+1} \cdot \Delta t \\ \dot{x}_k + \frac{1}{m} u_k^x \cdot \Delta t \\ \dot{y}_k + \frac{1}{m} u_k^y \cdot \Delta t \\ \dot{z}_k + \frac{1}{m} u_k^z \cdot \Delta t \end{bmatrix} + \mathbf{w}_k \\ &= f(\mathbf{X}_{k,k+1}, \mathbf{u}_k, m, \mathbf{w}_k, \Delta t). \end{aligned} \tag{4.1}$$

The augmented system then takes the form

$$\mathbf{X}^a_{k+1} = \begin{bmatrix} \mathbf{X}_{k+1} \\ m_{k+1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{X}_{k,k+1}, \mathbf{u}_k, m, \mathbf{w}_k, \Delta t) \\ m_k \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k \\ w_k^m \end{bmatrix}, \tag{4.2}$$

in which the persistence model is chosen for the parameter to be estimated, with the addition of an artificial small noise to induce a random walk that would allow for the estimate to keep updating and seek the best possible value given the acquired information [67]. The measurement equation then becomes

$$\begin{aligned}
\mathbf{Y}^a_{k+1} &= \tilde{\mathbf{H}}\mathbf{X}^a_{k+1} + \mathbf{v}_k \\
&= \begin{bmatrix} \mathbf{H} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}_{k+1} \\ m_{k+1} \end{bmatrix} + \mathbf{v}_k \\
&= \mathbf{H} \cdot \mathbf{X}_{k+1} + \mathbf{v}_k = \mathbf{y}_{k+1},
\end{aligned}$$

keeping our original measurements vector $\mathbf{y}_{k+1} \in \mathbb{R}^{n \times 1}$ by defining a null-space augmented observation operator of the form $\tilde{\mathbf{H}} = [\ \mathbf{H} \ \ 0\ ]$, which excludes the additional added states from the process.

4.2.2.2 Extended Kalman Filter

To work with the newly obtained nonlinear system, an Augmented Extended Kalman Filter (AEKF) will be formulated. The following time update equations were obtained

$$\hat{\mathbf{X}}^{a-}_{k+1} = f(\hat{\mathbf{X}}^{a+}_k, \mathbf{u}_k, m, w^m_k, \Delta t) \tag{4.3}$$

$$\mathbf{Q}^{a-}_{k+1} = \mathbf{A}^a_d\Big|_k \cdot \mathbf{Q}^{a+}_k \cdot \mathbf{A}^a_d\Big|_k^T + \mathbf{W}^a, \tag{4.4}$$

where the corresponding linearized system matrix takes the form

$$\mathbf{A}^a_d\Big|_k = \frac{\partial f}{\partial \mathbf{X}^a}\Big|_k = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -\frac{u^x_k \cdot \Delta t}{m^2_k} \\ 0 & 0 & 0 & 0 & 1 & 0 & -\frac{u^y_k \cdot \Delta t}{m^2_k} \\ 0 & 0 & 0 & 0 & 0 & 1 & -\frac{u^z_k \cdot \Delta t}{m^2_k} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.5}$$

The measurement update equations were taken to be

$$\hat{\mathbf{X}}_{k+1}^{a^+} = \hat{\mathbf{X}}_{k+1}^{a^-} + \mathbf{K}_{k+1}^{a}\left[\mathbf{y}_k - \tilde{\mathbf{H}}\cdot\hat{\mathbf{X}}_{k+1}^{a^-}\right] \tag{4.6}$$

$$\mathbf{Q}_{k+1}^{a^+} = \left[\mathbf{I} - \mathbf{K}_{k+1}^{a}\tilde{\mathbf{H}}\right]\mathbf{Q}_{k+1}^{a^-} \tag{4.7}$$

$$\mathbf{K}_{k+1}^{a} = \mathbf{Q}_{k+1}^{a^-}\tilde{\mathbf{H}}\left[\tilde{\mathbf{H}}\mathbf{Q}_{k+1}^{a^-}\tilde{\mathbf{H}}^T + \mathbf{R}_k\right]^{-1}, \tag{4.8}$$

where it is interesting to note how the Kalman gain and the *a posteriori* calculation remain decoupled, yielding

$$\begin{bmatrix}\mathbf{K_X}\\ \mathbf{K}_m\end{bmatrix} = \begin{bmatrix}\mathbf{Q_X}\mathbf{H}^T\left(\mathbf{H}\mathbf{Q_X}\mathbf{H}^T + \mathbf{R}\right)^{-1}\\ \mathbf{Q}_{\mathbf{X}m}^T\mathbf{H}^T\left(\mathbf{H}\mathbf{Q_X}\mathbf{H}^T + \mathbf{R}\right)^{-1}\end{bmatrix},$$

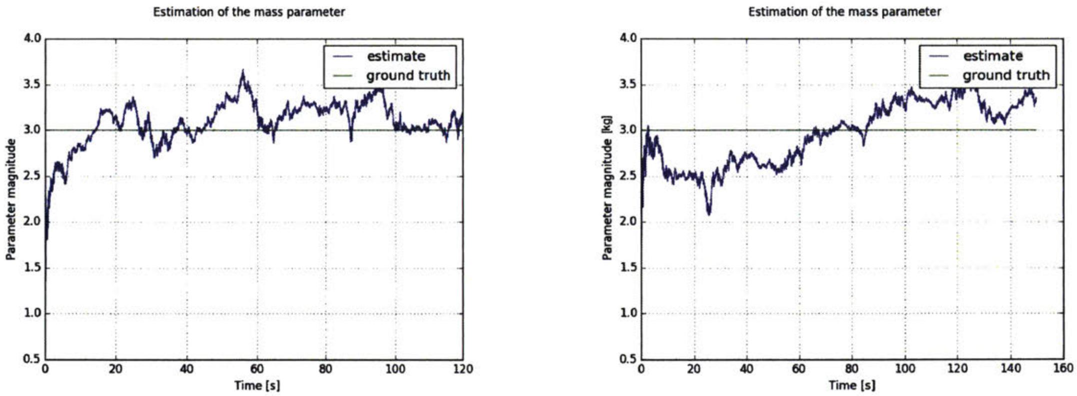
$$\text{where }\ \mathbf{Q}^a = \left[\begin{array}{c:c}\mathbf{Q_X} & \mathbf{Q}_{\mathbf{X}m}\\ \hdashline \mathbf{Q}_{\mathbf{X}m}^T & \mathbf{Q}_m\end{array}\right],$$

being it easy to see that the information to perform the inference over the parameter m to be estimated comes from the correlation between the system's states and the mass.

4.2.2.3 Parameter Estimates Behavior

Simulation results for an open-loop controlled plant are presented, in which the state was augmented to account for the parameter estimation of the spacecraft's mass. Figure 4-11 shows two entire simulation trials with very low artificial noise — just enough to allow it to search for the true value while keeping it within a close vicinity of the true value.

While the observed performance yields estimated values close to the true mass parameter of the spacecraft, it can easily be seen that the estimate is not truly converging towards a particular value. This is due to the fact that the parameter identification is being driven by the fictitious noise added to the persistent model of the augmented state.



(a) Sample behavior of the mass estimate.

(b) Filtering performed upon augmented state.

Figure 4-11: Full simulation runs for parameter estimation trials using the augmented extended Kalman filtering framework; open-loop control was exerted upon the system to verify the filter's performance (no tracking performed). It can be seen that the estimated value stays within a close vicinity of the true parameter value, but never converges. Initial estimate $\hat{m}_0 = 1\ kg$, true mass parameter $m = 3\ kg$.

The combination of the persistent model — which is chosen due to the assumption that the mass is a constant value to be identified — with the additive random perturbation is the reason that the estimate presents a random walk type of trajectory; the step size of this stochastic motion is a function of the fictitious noise magnitude. This added stochastic component is the main limitation of the approach, since no convergence guarantees or asymptotic trends can be offered.

Another important characteristic of this parameter estimation approach is the form in which the inferences are performed; by using the information provided by the correlation matrices of the augmented space and the state space, any wrong data association can potentially wreak havoc in the value of the parameter estimate (somewhat analogous of an erroneous loop-closure within the SLAM framework [68]). Therefore, without stability proofs and guarantees, the use of this technique within risk-averse systems, such as spacecraft, is definitely not advised. To enable this inference to be employed, a joint solution involving adaptive controllers is explored.

4.2.3 Composite Adaptation

So far, a learning approach through an adaptive controller was shown yields a significant performance improvement, given its ability to overcome large system uncertainties. Unfortunately, whenever the system operates within a stochastic environment, some issues such, such as bursting, were observed. To overcome these problems, the estimation results obtained through the parameter identification framework will be employed to contain the bursts and offer an improved performance. The combination of these two techniques is known as composite adaptation [10].

4.2.3.1 Information and Sources for Inference

The fact that the system is able to learn using both adaptive controllers as well as parameter estimation schemes signifies that either there is a single source of information from which the two algorithms draw inferences, or that each algorithm extracts information from a separate source. As it can easily be seen from their formulation, the latter is the correct assessment.

The Model Reference Adaptive Controller formulation was shown exploits the tracking error encapsulated in the time-varying surface $\mathbf{s}(t)$, which was defined as a weighted sum of errors and their derivatives. On the contrary, given that the estimator is agnostic to the type of control being used, the information source used by parameter estimation schemes is the so-called output prediction error, commonly referred to as the innovation term in the Kalman framework, defined as $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \tilde{\mathbf{H}} \cdot \hat{\mathbf{X}}_k^{a^-}$. The architecture of the composite adaptation scheme is shown in Fig. 4-12.

Such an adaptation scheme doesn't only maintain the guarantees offered by the adaptive controller, such as global stability [69], but also improves its convergence performance by using the innovation term for adaptation.

4.2.3.2 System Parameterization

In order to employ the composite adaptation technique, a correct parameterization of the system is to be performed. Typically a linear parametrization model is utilized, in

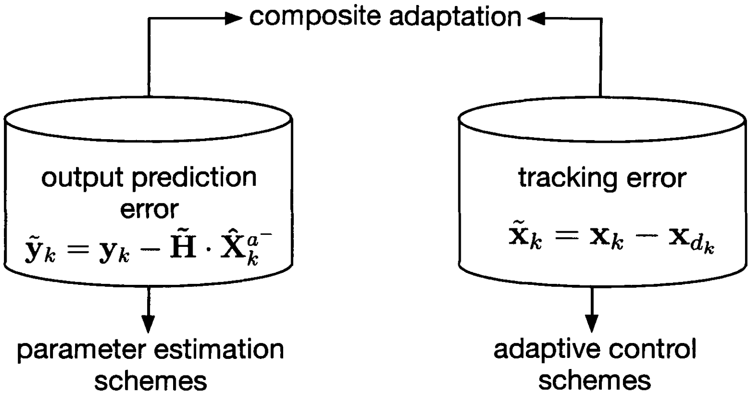

composite adaptation
output prediction error
$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \tilde{\mathbf{H}} \cdot \hat{\mathbf{X}}_k^{a-}$
tracking error
$\tilde{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x}_{d_k}$
parameter estimation schemes
adaptive control schemes

Figure 4-12: Diagram showcasing which information sources are being leveraged by which algorithms. Composite adaptation uses both sources to improve upon its performance.

which the vector is comprised of the unknown and wanted parameters of the system.

To leverage both approaches, the adaptation equations from both techniques are taken and merged to obtain

$$\left.\begin{aligned} \text{MRAC: } \dot{\hat{a}}(t) &= -\mathbf{P}\mathbf{Y}^T\mathbf{s}(t) \\ \text{estimation: } \dot{\hat{a}}(t) &= -\mathbf{\Gamma}\mathbf{W}^T(t)\tilde{\mathbf{y}}(t) \end{aligned}\right\} \text{ Composite: } \dot{\hat{a}}(t) = -\mathbf{P}'(t) \cdot \left(\mathbf{Y}^T\mathbf{s} + \mathbf{W}^T\mathbf{R}(t)\tilde{\mathbf{y}}\right), \tag{4.9}$$

where $\mathbf{P}'$ denotes the system's new uniformly positive definite adaptation gain, $\mathbf{s}$(t) is the adaptive control surface posed as a function of the tracking errors, $\tilde{\mathbf{y}}$(t) is the Kalman filter innovation term or output prediction error, and $\mathbf{W}$(t) represents the signal matrix obtained after the system parametrization

$$\mathbf{\Theta}(t) = \mathbf{W}(t) \cdot \mathbf{\Phi} \longrightarrow \mathbf{\Theta}(t) = \begin{bmatrix} \mathbf{F}(t) \\ \int \mathbf{F}(t)\, dt \end{bmatrix} \cdot m,$$

with $\mathbf{\Theta}(t)$ being the system output vector, and $\mathbf{\Phi}$ the regressor for the parameters of interest. The matrix $\mathbf{R}(t)$ is a uniformly positive definite matrix indicating the level of importance of the current prediction errors.s

4.2.3.3 Simulation Results

To assess the performance of the enhanced algorithm operating within a stochastic environment, in which not only are a noisy process model and corrupted measurements present, but also high uncertainties in the physical parameters of the system. In addition, a prior knowledge upon the initial mass estimate of $\hat{m}_0 = 1\ kg$ will be given, with a ground truth value of $m^* = 3\ kg$.

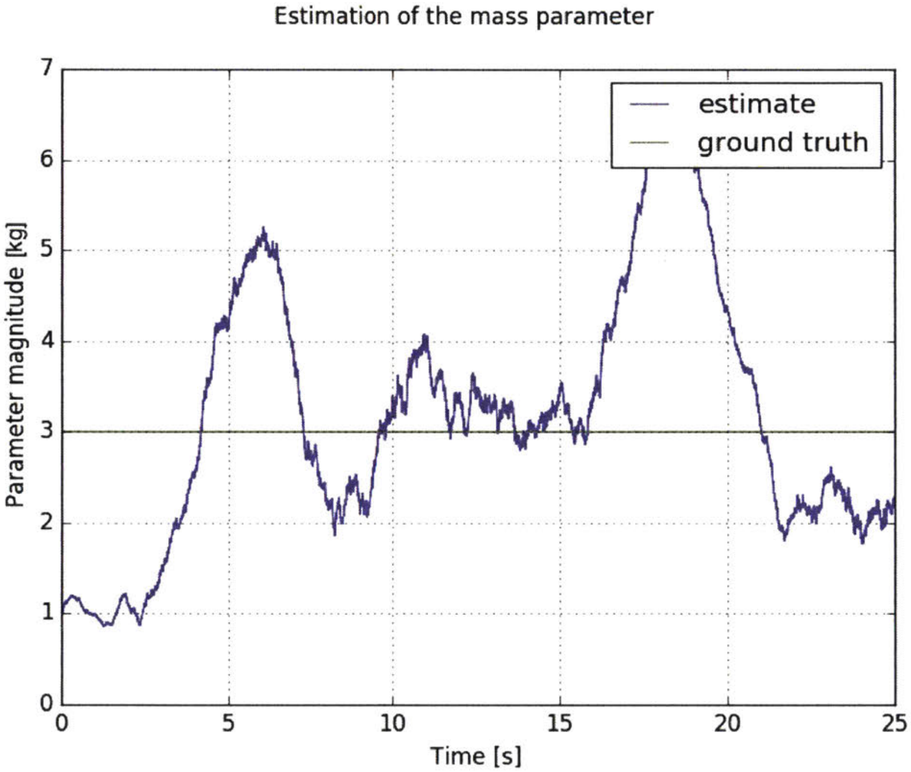
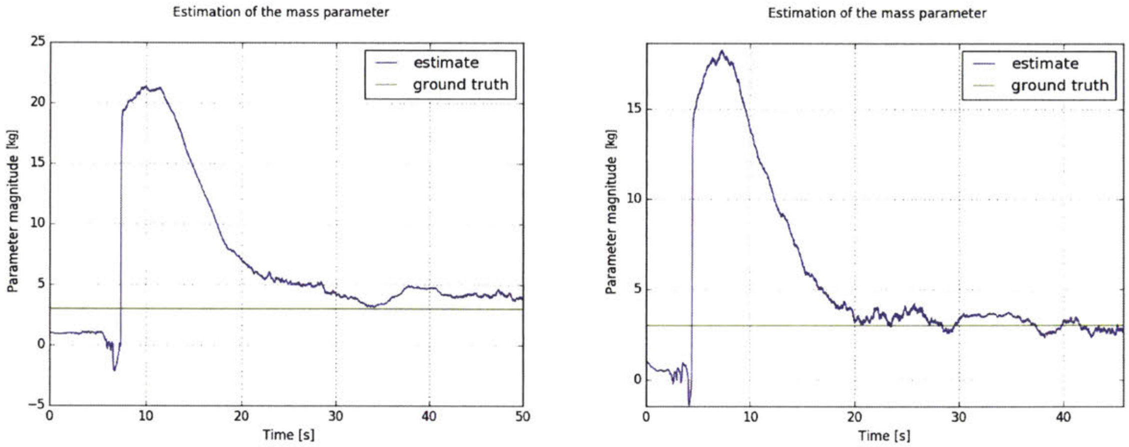


Figure 4-13: Composite system run #1: a lax but bounded adaptation profile for the mass estimate is exhibited by the composite adaptation technique under the presence of noisy measurements. No bursts or divergence behavior is observed throughout the full duration of the run. Initial estimate $\hat{m}_0 = 1\ kg$, true mass parameter $m = 3\ kg$.

A first attempt on a composite adaptive approach is shown in Fig. 4-13. The main takeaway from this initial simulation's performance is the absence of bursts and diverging behavior previously observed in the stochastic MRAC simulations. The oscillatory behavior signifies that the weight on the rate of adaptation is not sufficient to drive the parameter estimate towards its true value, since equal importance was given to the tracking error and the output prediction error. The stochastic random walk component imparted by the parameter identification part of the scheme is clearly noticeable.



(a) Composite system run #2.

(b) Composite system run #3.

Figure 4-14: A more aggressive adaptation profile is obtained by adjusting the λ, k, and $\mathbf{P}'$ gain values for a better estimation performance; some residual offset error is shown in (a), while a closer adaptation towards the ground truth value is shown in (b). Initial estimate $\hat{m}_0 = 1\ kg$, true mass parameter $m = 3\ kg$.

The lax adaptation profile shown in Fig. 4-13 can be defeated by modifying the composite adaptive controller tuning parameters. By choosing higher weights on the sliding surface contribution, more aggressive adaptation profiles are obtained, as shown in Fig. 4-14. A slight overshoot is present at first, which is characteristic of the stochastic MRAC, and then, the converging properties take over to drive the estimate towards the ground truth value.

4.3 Applicability and Importance

4.3.1 Summary

By analyzing the on-orbit data from InSPIRE-II SPHERES test sessions, a lack of controllability that impairs the enablement of successful and reliable autonomous docking operations was discovered. Subsequently, using probabilistic model analysis, it was found that by enabling on-line learning algorithms through an adaptive MRAC approach, uncertainties in the systems' parameters could be overcome and a clear increase in tracking performance could be obtained.

Nonetheless, limitations in the controller's formulation while operating in stochastic environments impede a sole MRAC approach from overcoming a performance barrier imposed by the disrupted learning behavior mainly characterized by sudden bursts and divergence patterns. In order to trounce this roadblock and obtain a better learning scheme, a composite adaptation solution that leverages both the adaptive controller's information as well as a paired estimator's information was shown to exhibit the sought parameter adaptation properties within a stochastic environment.

Some of the benefits obtained when considering the proposed approach for on-orbit operations over conventional path planning and control methods are illustrated through examples in the following section.

4.3.2 Expansibility

In-space robotic assembly can be thought of as a succession of multiple docking maneuvers to be carried out between systems of distinct spacecraft and pieces. Given the inherent uncertainty in the physical properties of each assembly element, the confidence on the characteristics of the full ensemble will start to decrease as the number of satellites or the number of reconfiguration maneuvers increase. This is pictorially and simply depicted in Fig. 4-15, in which the belief on the assembly's full mass starts decreasing as the system continues to get incrementally built.

If the final system is comprised of a small number of well known pieces, then system identification methods (e.g., mass identification through known thruster firing sequences) could potentially still be realizable, but as the number of components start to increase, an exponential number of combinations becomes available and running system identification tests on every single one of them rapidly becomes intractable (especially on-orbit). Hence, learning techniques and adaptive algorithms that are able to concurrently stabilize the system while fully characterizing the ensemble, such as the proposed adaptive composite algorithm, were demonstrated to be an elegant solution to the problem.

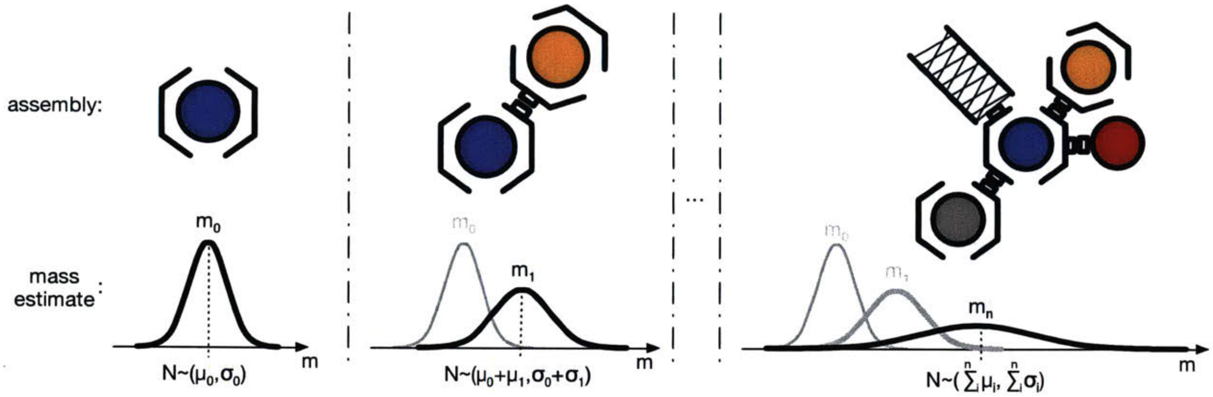


Figure 4-15: Spacecraft ensembles with a corresponding belief on their mass value. As the number of pieces increase, the certainty in the estimate starts decreasing.

4.3.3 Simplified, Smoothed Probabilistic Trajectories

The impact that the proposed composite adaptive probabilistic approach has upon a mission objective likelihood of success, or even feasibility, can be demonstrated through the analysis of a simple docking maneuver. A typical algorithm provides a reference trajectory that is a function of the target satellite's state and motion, as can be shown in Fig. 4-16. This "live-tracking" behavior, as the chaser starts to close in to the target, can likely demand large relative displacements to the chaser at a certain frequency $\boldsymbol{\omega}$, given that it is highly unlikely that the target is deterministically and statically pointing in one single orientation. If the target's chattering behavior crosses a certain threshold, the delta-V maneuvers required from the chaser might exceed its maximum thrust capabilities, thus not being able to satisfy the objective's requirement, rendering the mission to be unfeasible.

Conversely, if the probabilistic approach described throughout the previous sections is to be taken for the same scenario, a feasible objective is to be obtained by not imposing unwarranted requirements upon the chaser (impossible delta-V maneuvers given the chaser's available total thrust). Instead of commanding a "live-track" reference trajectory, a probability model can be fitted over the observed or known behavior of the target satellite, and a simpler, smoothed reference can be computed ; an example of this is shown in Fig. 4-17 (the mean of the chattering movement of

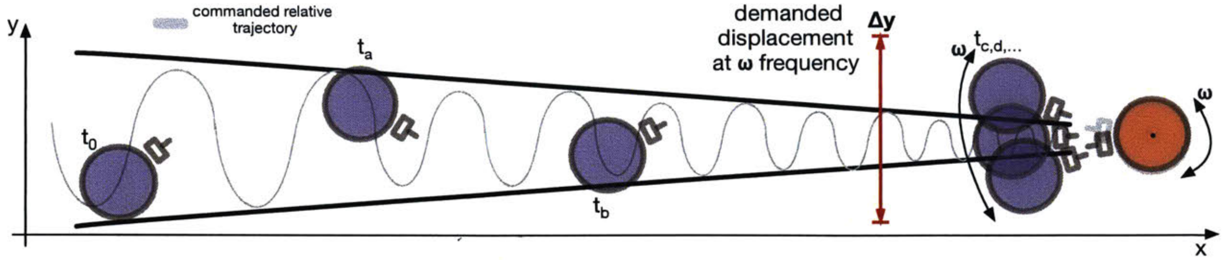

y
commanded relative trajectory
t_a
t_0
t_b
demanded displacement at ω frequency
Δy
ω
t_c,d,...
ω
x

Figure 4-16: Typical relative reference trajectory computed using the target's satellite state information; in can be seen that by "live-tracking" the target satellite, the chaser spacecraft is required to perform large delta-v maneuvers, which in some cases might render a mission to be infeasible.

the target, in this case).

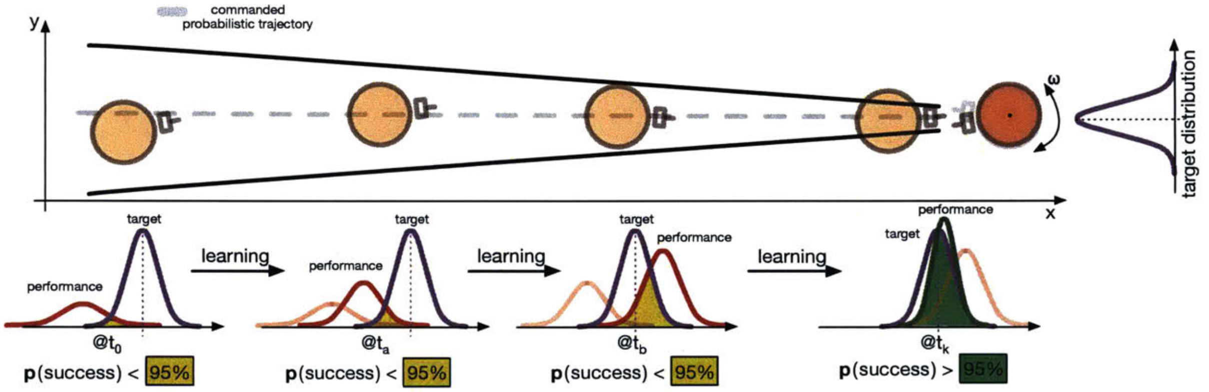


Figure 4-17: Simplified and smoothed relative reference trajectory example obtained using the proposed probabilistic and learning approach for in-space robotic assembly.

By utilizing a composite adaptive approach, the chaser satellite will be able to generate a probability model of its own performance, and modify it on-board through its learning capabilities. At the moment of docking, the chaser will be able to assess whether its hitherto performance satisfies the requirements to probabilistically say that, in expectation, a successful docking can be carried out, further aiding the decision making processes and the full autonomy subsystem of the spacecraft.

Chapter 5

Conclusion

Through the analysis of on-orbit data obtained from SPHERES test sessions, the complexity that arises from having to deal with large system uncertainties — such as unknown thruster forces, position of the center of mass, moments and products of inertia, etc — was able to be shown. For in-space robotic assembly operations, such a scenario is highly likely to be obtained; an example of this was shown in Fig. 4-15, in which the mass estimate variance of the growing assembly starts to increase given the compounded uncertainties of the available prior information.

In order to overcome these adversities, an adaptive controller technique based on a Model Reference Adaptive Control law was formulated and implemented using the SPHERES ground testing facilities. It was demonstrated that the ability to cope with gross uncertainties in the system's mass properties was greatly benefited by the enablement of on-board learning algorithms to be executed in real-time utilizing the formulated techniques.

An algorithm that employs the formulated technique and the proposed probabilistic operation approach for in-space robotic assembly situations was elaborated, and was analyzed by applying it to the InSPIRE-II test sessions data. Although an improvement upon the likelihood of successfully achieving a docking maneuver was observed, a barrier in the performance increment was encountered when analyzing the learning properties of the MRAC under stochastic environments.

It was found that by combining the learning approach granted to the adaptive controller by the tracking error information source with a learning approach based off of an output prediction error information source, the system parameters' adaptation behavior could be greatly improved, being able to avoid adverse conditions such as bursting phenomena or diverging actions. The combination of these techniques is known as composite adaptation.

In spite of having demonstrated the composite adaptive algorithm's behavior only through stochastic simulation environments, the observed results appear to be very promising for being able to obtain an algorithm that would allow for reliable in-space robotic assembly operations. Further research on the topic will continue to be carried out, and a list of possible future work will be delineated in the following sections.

5.1 Contributions

The main contributions of this work are:

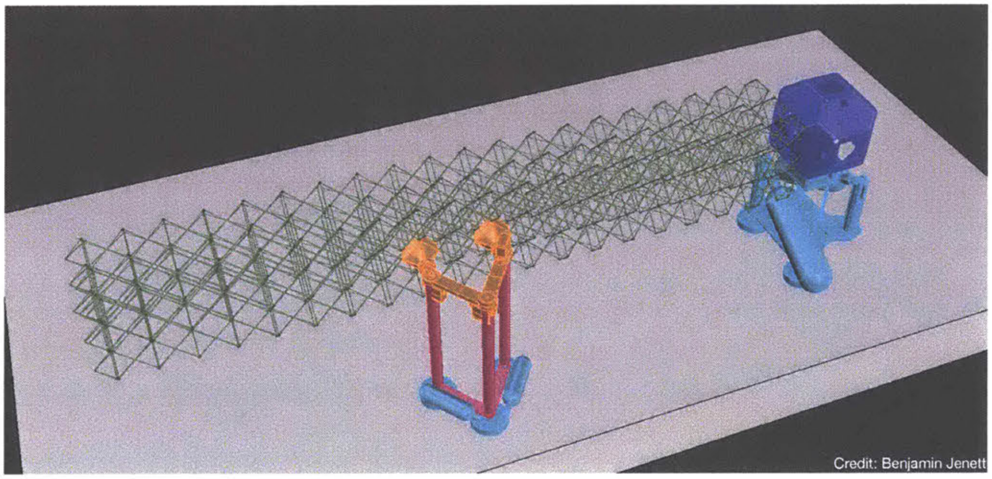
- Demonstration of the benefits and impact of adaptive controller approaches for dealing with non-negligible system uncertainties shown through improvements in SPHERES ground testing operations.
- Hardware validation of a learning algorithm approach enabled by Model Reference Adaptive Controller techniques.
- The proposal of an algorithm to augment the MRAC's learning capabilities by pairing the controller with a parameter identification technique via a Composite Adaptation approach, and its validation through simulation.
- A probabilistic approach for on-board decision making through feasibility and performance analysis consisting on the derivation of probability models from the target satellite's characteristics and motion, and from the chaser satellite's sensed performance information.
- A method for computing a metric using probability density functions to specify, in expectation, the likelihood of success of an in-space robotic autonomous

docking maneuver.

- The introduction of a path planning scheme in which simplified and smoothed trajectories for in-space robotic assembly are generated by analyzing the target satellites' motion using probability models.

5.2 Future Work

- Hardware demonstration of the proposed algorithm and techniques through ground and on-orbit test sessions, with the introduction of more complex, dynamically changing ensembles, as the one presented in Fig. 5-1.



Credit: Benjamin Jenett

Figure 5-1: Sketch of ground testing of autonomous on-orbit assembly systems.

- Further exploration into the implementation of a full autonomy cycle algorithm (e.g., probabilistic docking algorithm (2)) with proposed metrics, heuristic, planner, and adaptive learning controllers.

- In order to accelerate the convergence properties of the adaptation, and to further prevent undesirable drifting and bursting conditions, a parameter projection method to keep the estimates within a convex bounded set could be explored [70].

- To account for highly time-varying parameters (e.g., in the middle of reconfiguration maneuvers), exponential resetting and forgetting techniques can be implemented within the parameter identification framework, allowing for an improvement in performance throughout poorly excited regimes of operation [71].

- Adaptive learning scheduling through e-modification techniques for avoiding over- and un-learning behaviors is an important characteristic to include to the algorithm [72].
- For the probabilistic docking algorithm, temporal constraints can be added to the planning strategy in order to determine a probability model with the feasible time windows in which an action could be successfully performed.

Appendix A

MRAC Stability Analysis

A.1 Position Controller

Since the formulation of the problem involves a non-autonomous system (after time-discretization and with time-varying parameters such as the mass), traditional Lyapunov tools and theorems — like the Invariant Set Theorem — cannot be used to reach any conclusions [10]. Instead, Barbalat's lemma will be employed to evaluate Lyapunov-like functions and analyze the convergence and stability of the system.

Lemma A.1.1. *Barbalat's lemma*

Let $t \mapsto F(t)$ be a differentiable function with a finite limit as $t \to \infty$. If $\dot{F}$ is uniformly continuous, then $\dot{F}(t) \to 0$ as $t \to \infty$ [73].

A simple modification for control systems' stability analysis yields the following:

Lemma A.1.2. *Let M be a continuous, positive definite function and x be an absolutely continuous function on* R. *If $||x(\cdot)||_{\mathrm{L}_\infty} < \infty$, and $\lim_{T\to\infty} \int_0^T M(x(t))\, dt < \infty$, then $x(t) \to 0$ as $t \to \infty$ [73].*

An immediate and very practically useful corollary follows from these lemmas, which can be stated as:

Corollary A.1.2.1. *If the differentiable function $F(t)$ has a finite limit as $t \to \infty$, and is such that $\ddot{F}$ exists and is bounded, then $\dot{F}(t) \to 0$ as $t \to \infty$ [74].*

Remark. *From these results, it can be easily seen that an analogous to a Lyapunov analysis invariant set theorem can be expressed. Therefore, if a scalar function* $V(\mathbf{x}, t)$ *satisfies the conditions*

- $V(\mathbf{x}, t)$ *is lower bounded*
- $\dot{V}(\mathbf{x}, t)$ *is negative semi-definite*
- $\dot{V}(\mathbf{x}, t)$ *is uniformly continuous in time*

then $\dot{V}(\mathbf{x}, t) \to 0$ *as* $t \to \infty$.

Traditionally, Lyapunov functions (Lyapunov-like, in this case) that resemble some sort of energy expression (bounded quantity) are chosen as initial candidates. In this case, since it is also of interest to prove the controller's stability and proper convergence for an ideal tracking performance, the initial candidate function will be in terms of the surface $\mathbf{s}_p(t)$ so that not only an energy-like function is obtained, but also a strictly positive squared error of both the position and the velocity, yielding

$$V_p(\mathbf{s}_p, t) = \frac{1}{2}\, \mathbf{s}_p^T(t) \cdot m \cdot \mathbf{s}_p(t), \tag{A.1}$$

where $\mathbf{s}_p(t)$ is just our defined surface, m is the unknown mass of the system, and the $1/2$ is just added for convenience, given the fact that the derivative of the quadratic function will need to be taken, and any extra terms are not desired to be carried around. It is trivial to see how this function allows for the satisfaction of the first condition, given that a squared-error function is lower bounded by 0.

Taking the derivative of the function, and conveniently using Eq. (2.14) to substitute the value of the tracking error surface and Eq. (2.12) for the systems' dynamics, an expression of the form

$$\begin{aligned}
\dot{V}_p(\mathbf{s}_p, t) &= \frac{1}{2} \cdot \left(2\mathbf{s}_p^T(t) \cdot m \cdot \dot{\mathbf{s}}_p(t)\right) \\
&= \mathbf{s}_p^T(t) m \left(\ddot{\mathbf{x}}_p(t) - \ddot{\mathbf{x}}_r(t)\right) \\
&= \mathbf{s}_p^T(t) m \left(\frac{1}{m}\mathbf{F}(t) - \ddot{\mathbf{x}}_r(t) - \frac{\alpha}{m} \cdot \dot{\mathbf{x}}_p(t) - \frac{\beta}{m} \cdot \mathbf{x}_p(t)\right) \\
&= \mathbf{s}_p^T(t) \left(\mathbf{F}(t) - \overbrace{m}^{\text{unknown}} \cdot \underbrace{\ddot{\mathbf{x}}_r(t)}_{\text{known}} - \overbrace{\alpha}^{\textit{unknown}} \cdot \underbrace{\dot{\mathbf{x}}_p(t)}_{\text{known}} - \overbrace{\beta}^{\textit{unknown}} \cdot \underbrace{\mathbf{x}_p(t)}_{\text{known}}\right)
\end{aligned} \tag{A.2}$$

is obtained. Is it important to reiterate that a negative semi-definite derivative is sought. From the last line of Eq. (A.2) it can be seen that the right hand side of the expression within the parenthesis can be broken up into two parts; a vector of unknown comprised of the model parameters m, α, β, and a known part encompassing $\ddot{\mathbf{x}}_r(t), \dot{\mathbf{x}}_p(t), \mathbf{x}_p(t)$, which are only functions of the current and desired states.

To further simplify the expression, the definition of the vector of unknowns

$$\mathbf{a}_p = [\, m \;\; \alpha \;\; \beta \,]^T, \tag{A.3}$$

and the definition of a matrix of known expressions

$$\begin{aligned}
\mathbf{Y}_p(t) &= [\, \ddot{\mathbf{x}}_r(t) \;\; \dot{\mathbf{x}}_p(t) \;\; \mathbf{x}_p(t) \,]^T \\
&= \begin{bmatrix} \ddot{x}_r(t) & \dot{x}(t) & x(t) \\ \ddot{y}_r(t) & \dot{y}(t) & y(t) \\ \ddot{z}_r(t) & \dot{z}(t) & z(t) \end{bmatrix}
\end{aligned} \tag{A.4}$$

allows for Eq. (A.2) to be rewritten as

$$\dot{V}_p(\mathbf{s}_p, t) = \mathbf{s}_p^T(t) \left(\mathbf{F}(t) - \mathbf{Y}_p(t) \cdot \mathbf{a}_p\right), \tag{A.5}$$

where $\mathbf{F}(t)$ is the control input to the system, which can be any arbitrary expression that the control system engineer conjures. Thus, keeping in mind that a negative semi-definite function is desired, one could simply define a control law of the form

$$\begin{gathered}\mathbf{F}(t) \equiv \mathbf{u}(t) := \mathbf{Y}_p(t) \cdot \mathbf{a}_p - k \cdot \mathbf{s}_p(t), \quad \text{where} \\ k \in \mathrm{R}^+ : \text{strictly positive user-defined parameter}\end{gathered} \tag{A.6}$$

which, after substitution into Eq (A.5), would yield

$$\begin{aligned}\dot{V}_p(\mathbf{s}_p, t) &= \mathbf{s}_p^T(t)\,(\cancel{\mathbf{Y}_p(t) \cdot \mathbf{a}_p} - k \cdot \mathbf{s}_p(t) - \cancel{\mathbf{Y}_p(t) \cdot \mathbf{a}_p}) \\ &= \boxed{\; -k\ \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t) \leq 0 \;}\end{aligned} \tag{A.7}$$

the negative semi-definite derivative function to the Lyapunov-like candidate, satisfying another condition of Barbalat's lemma.

Unfortunately, given the fact the the parameters in $\mathbf{a}_p$ are unknown, and therefore not exact, the performed cancellation cannot be carried on. Instead, only time-varying estimates of the parameters are available, which are defined as

$$\hat{\mathbf{a}}_p(t) = \begin{bmatrix} \hat{m}(t) & \hat{\alpha}(t) & \hat{\beta}(t) \end{bmatrix}^T, \tag{A.8}$$

yielding a derivative of the form

$$\begin{aligned}\dot{V}_p(\mathbf{s}_p, t) &= \mathbf{s}_p^T(t)\,(\mathbf{Y}_p(t) \cdot \hat{\mathbf{a}}_p(t) - k \cdot \mathbf{s}_p(t) - \mathbf{Y}_p(t) \cdot \mathbf{a}_p) \\ &= \mathbf{s}_p^T(t) \left(-k \cdot \mathbf{s}_p(t) + \mathbf{Y}_p(t) \cdot \left(\underbrace{\hat{\mathbf{a}}_p(t) - \mathbf{a}_p}_{:= \tilde{\mathbf{a}}_p(t)} \right) \right) \\ &= \boxed{\; \underbrace{-k\ \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t)}_{\preceq} + \underbrace{\mathbf{s}_p^T(t) \cdot \mathbf{Y}_p(t) \cdot \tilde{\mathbf{a}}_p(t)}_{\text{arbitrary}} \nleq 0 \;}\end{aligned} \tag{A.9}$$

It can be noted that by simply eliminating the second term of Eq. (A.9), the expression would be able to satisfy the negative semi-definiteness property of the lemma's second condition. In order to do this, some modifications to Eq. (A.2) need to be made; specifically, it is sought to add a term such that, after taking the derivative, the second part of the expression gets canceled.

Thus, rewriting Eq. (A.1), the complete Lyapunov-like candidate function is

$$\begin{gathered}\bar{V}_p(\mathbf{s}_p, t) = \frac{1}{2}\, \mathbf{s}_p^T(t) \cdot m \cdot \mathbf{s}_p(t) + \overbrace{\frac{1}{2}\, \tilde{\mathbf{a}}_p^T(t) \cdot \mathbf{P}_p^{-1} \cdot \tilde{\mathbf{a}}_p^T(t)}^{\text{added term}}, \quad \text{where} \\ \inf \bar{V}_p(\mathbf{s}_p, t) = 0, \quad \text{and} \\ \mathbf{P}_p \succ 0 \in \mathrm{R}^{3\times 3} : \quad \text{user-defined system parameter matrix}\end{gathered} \tag{A.10}$$

which again satisfies the first condition of Barbalat's lemma by being a multivariate quadratic function in terms of the position, velocity, and parameter estimates squared errors, lower bounded by zero.

Now, by taking the derivative of the candidate function

$$\begin{aligned}\dot{\bar{V}}_p(\mathbf{s}_p, t) &= \frac{1}{2} \cdot \left(2\mathbf{s}_p^T(t) \cdot m \cdot \dot{\mathbf{s}}_p(t)\right) + \frac{1}{2} \cdot \left(2\dot{\hat{\mathbf{a}}}_p^T(t) \cdot \mathbf{P}_p^{-1} \cdot \tilde{\mathbf{a}}_p(t)\right) \\ &= \underbrace{-k\, \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t) + \mathbf{s}_p^T(t) \cdot \mathbf{Y}_p(t) \cdot \tilde{\mathbf{a}}_p(t)}_{V_p(\mathbf{s}_p,t)} + \underbrace{\dot{\hat{\mathbf{a}}}_p^T(t) \cdot \mathbf{P}_p^{-1} \cdot \tilde{\mathbf{a}}_p(t)}_{\bar{V}_p(\mathbf{s}_p,t)\ \text{extra term}} \\ &= -k\, \mathbf{s}_p^T(t) \cdot \mathbf{s}_p(t) + \left(\mathbf{s}_p^T(t) \cdot \mathbf{Y}_p(t) + \dot{\hat{\mathbf{a}}}_p^T(t) \cdot \mathbf{P}_p^{-1}\right) \cdot \tilde{\mathbf{a}}_p(t),\end{aligned} \tag{A.11}$$

a term with the rate of change of the parameter estimates $\dot{\hat{\mathbf{a}}}_p(t)$ is obtained; this term is called the adaptation of the unknown parameters' vector, given that it will express how the unknown coefficients of the differential equation will be changing with respect to time.

Given that this adaptation "law" is, similarly to the control law, to be chosen by the designer, one could ensure to choose a form that would eliminate the extra term in Eq. (A.11) in order to satisfy the second condition of Barbalat's lemma. With this in mind, if the adaptation law

$$\begin{aligned}\dot{\hat{\mathbf{a}}}_p(t) &:= -\mathbf{P}_p \cdot \mathbf{Y}_p(t)^T \cdot \mathbf{s}_p(t) \\ \Rightarrow \dot{\hat{\mathbf{a}}}_p^T(t) &= -\mathbf{s}_p^T(t) \cdot \mathbf{Y}_p(t) \cdot \mathbf{P}_p\end{aligned} \tag{A.12}$$

is chosen, when substituted into Eq. (A.11), the sought cancellation

$$\begin{aligned}\dot{\bar{V}}_p(\mathbf{s}_p, t) &= -k\ \mathbf{s}_p^T(t)\cdot\mathbf{s}_p(t) + \left(\mathbf{s}_p^T(t)\cdot\mathbf{Y}_p(t) + \dot{\hat{\mathbf{a}}}_p^T(t)\cdot\mathbf{P}_p^{-1}\right)\cdot\tilde{\mathbf{a}}_p(t)\\ &= -k\ \mathbf{s}_p^T(t)\cdot\mathbf{s}_p(t) + \left(\mathbf{s}_p^T(t)\cdot\mathbf{Y}_p(t) - \mathbf{s}_p^T(t)\cdot\mathbf{Y}_p(t)\cdot\cancel{\mathbf{P}_p}\cdot\cancel{\mathbf{P}_p^{-1}}\right)\cdot\tilde{\mathbf{a}}_p(t)\\ &= -k\ \mathbf{s}_p^T(t)\cdot\mathbf{s}_p(t) + \left(\cancel{\mathbf{s}_p^T(t)\cdot\mathbf{Y}_p(t)} - \cancel{\mathbf{s}_p^T(t)\cdot\mathbf{Y}_p(t)}\right)\cdot\tilde{\mathbf{a}}_p(t)\\ &= \boxed{-k\ \mathbf{s}_p^T(t)\cdot\mathbf{s}_p(t)\leq 0\ \blacksquare}\end{aligned} \tag{A.13}$$

can be performed, and a negative semi-definite function is finally obtained, successfully satisfying Barbalat's lemma second condition.

Finally, it is needed to show that the aforementioned third condition of Barbalat's lemma — $\exists\,\delta \mid \sup\ddot{\bar{V}}_p(\mathbf{s}_p, t) = \delta$ — holds to prove that the proposed controller yields a stable system, and that the parameters converge to their true values.

By calculating the second time derivative of the candidate function $\ddot{\bar{V}}_p(\mathbf{s}_p, t) = -2k\ \mathbf{s}_p^T(t)\cdot\dot{\mathbf{s}}_p(t)$ and since it was already proven that $\inf\bar{V}_p(\mathbf{s}_p, t) = 0$, i.e., $\bar{V}_p(\mathbf{s}_p, t)\succeq 0$, paired with the fact that the monotonically decreasing function $\dot{\bar{V}}_p(\mathbf{s}_p, t)\preceq 0$, which indicates that there exists some γ such that $\sup\bar{V}_p(\mathbf{s}_p, t) = \gamma$, it can be concluded that both $\mathbf{s}_p(t)$ and $\mathbf{a}_p(t)$ are bounded. Since the closed-loop dynamics of the system can be expressed as $\dot{\mathbf{s}}_p(t) = f(\mathbf{s}_p, \mathbf{a}_p)$ (a function of bounded values), it follows that $\dot{\mathbf{s}}_p(t)$ is also bounded.

This indicates that, as the lemma and corollaries state, as $t\to\infty$ the asymptotic entailments

$$\dot{\bar{V}}_p(\mathbf{s}_p, t)\to 0 \Longrightarrow \left\{\begin{matrix}\tilde{\mathbf{a}}_\mathbf{p}(\mathbf{t})\to 0\\ \mathbf{s}_p(t)\to 0\end{matrix}\right\} \Longrightarrow \left\{\begin{matrix}\tilde{\mathbf{x}}_\mathbf{p}(\mathbf{t})\to 0\\ \dot{\tilde{\mathbf{x}}}_p(t)\to 0\end{matrix}\right\}$$

hold, and the controller tends towards a stable perfect tracking state, provided the references are bounded and stable (BIBO) ■.

Appendix B

MRAC Glass Table Testing Data

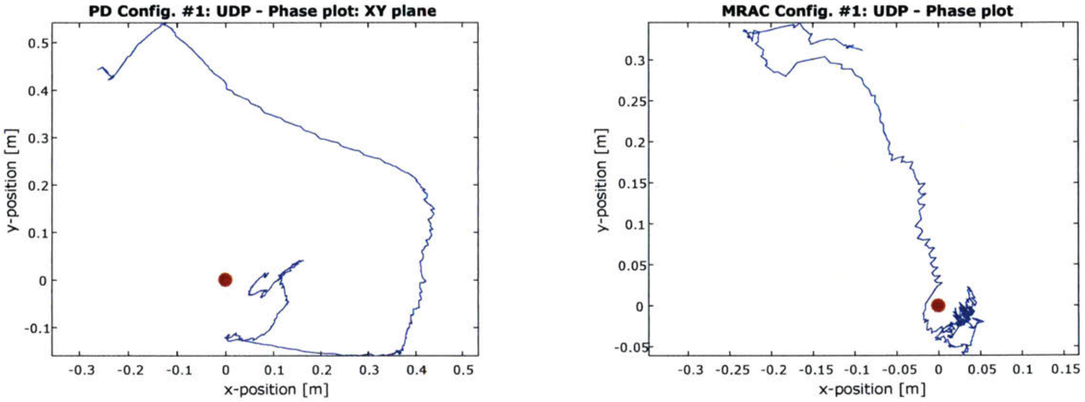
After finalizing the algorithm's implementation, multiple runs for each controller, for each configuration, were performed in order to collect data sets, video recordings, and assess their performance difference. This section will start by showing a recording per test for both the baseline SPHERES PD controller and the Model Reference Adaptive controller. Afterwards, the matching plots for each test will be analyzed, and to conclude, performance metrics in terms of error characteristics will be determined using the compounded tests for each of the controllers.

B.1 Configuration #1

In this section the data obtained for both the MRAC and the baseline SPHERES PD controller will be shown.

Figures B-4 and B-5 correspond to the hardware configuration #1; the satellite's behavior during each run can be observed in the embedded videos.

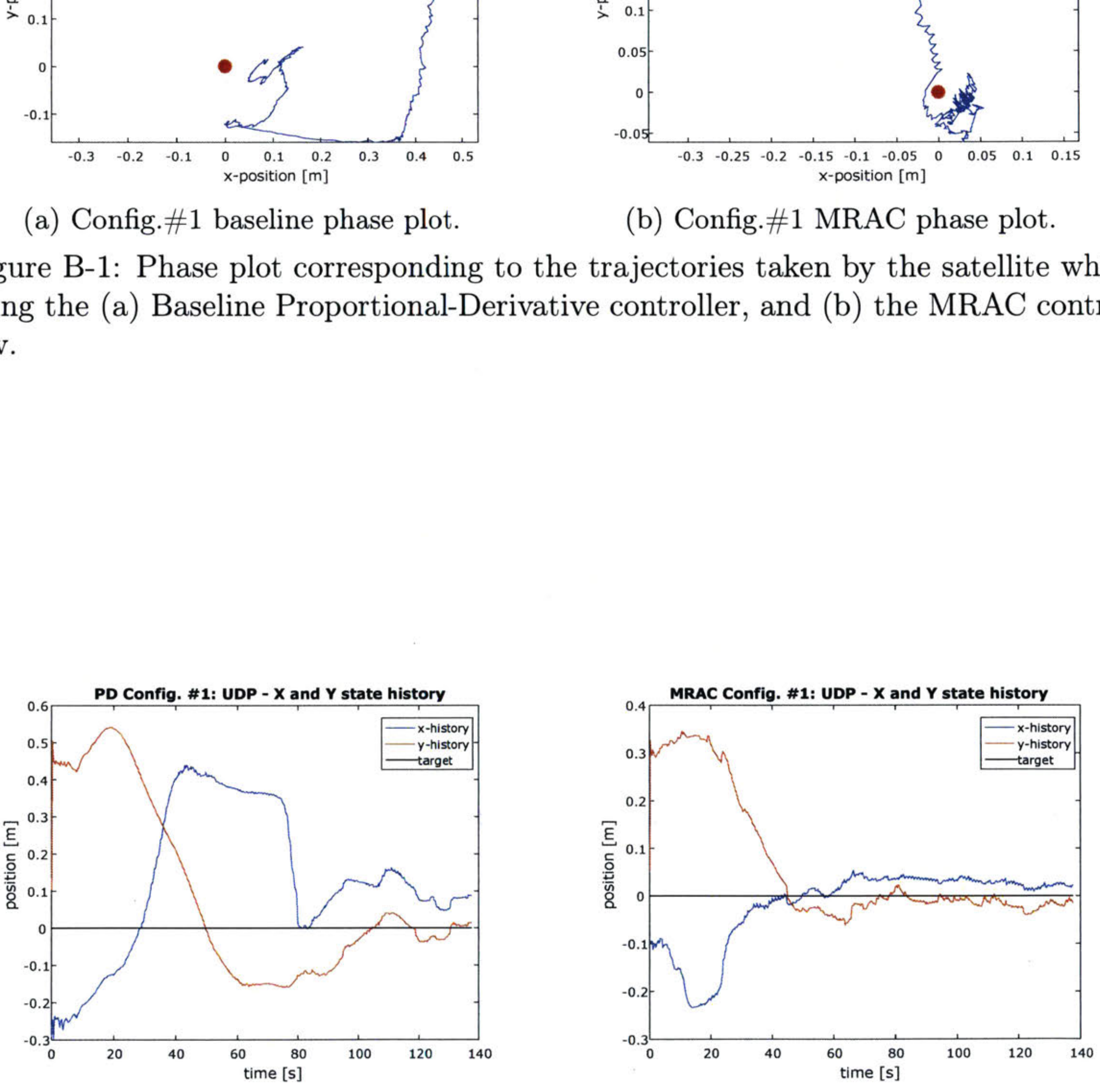
As it can be clearly seen from the video shown in Fig. B-4, even though the baseline PD controller was using the physical properties that correspond to this exact hardware configuration, the satellite exhibited notable overshoots while trying to translate from its initial position to the origin. Given the fact that position and attitude are coupled, large deviations in the yaw angle are induced by the translation motion which are not able to be corrected by the controller throughout the remaining test time.



(a) Config.#1 baseline phase plot.

(b) Config.#1 MRAC phase plot.

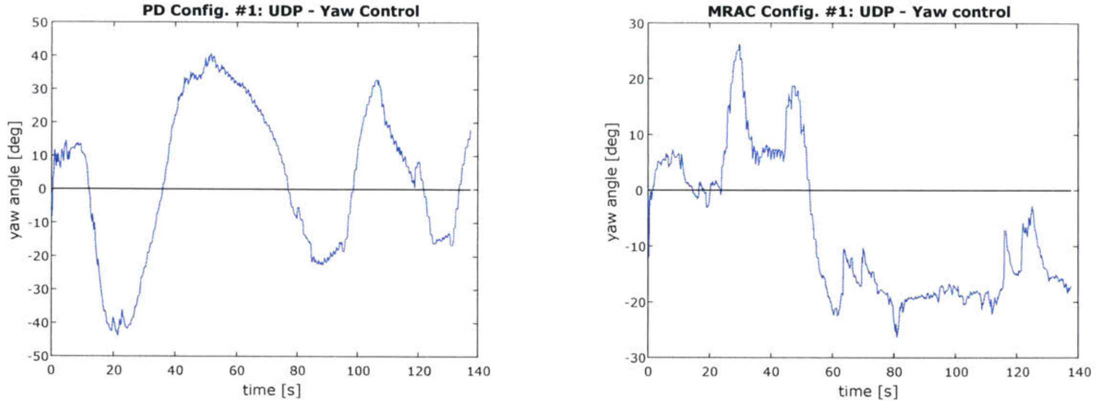
Figure B-1: Phase plot corresponding to the trajectories taken by the satellite while using the (a) Baseline Proportional-Derivative controller, and (b) the MRAC control law.



(a) Config.#1 baseline xy-states.

(b) Config.#1 MRAC xy-states.

Figure B-2: X and Y state histories for Configuration #1 under (a) SPHERES PD controller and (b) MRAC controller.



(a) Config.#1 baseline yaw history.

(b) Config.#1 MRAC yaw history.

Figure B-3: Yaw control performance exhibited for the UDP configuration using the (a) SPHERES PD controller and (b) the MRAC.

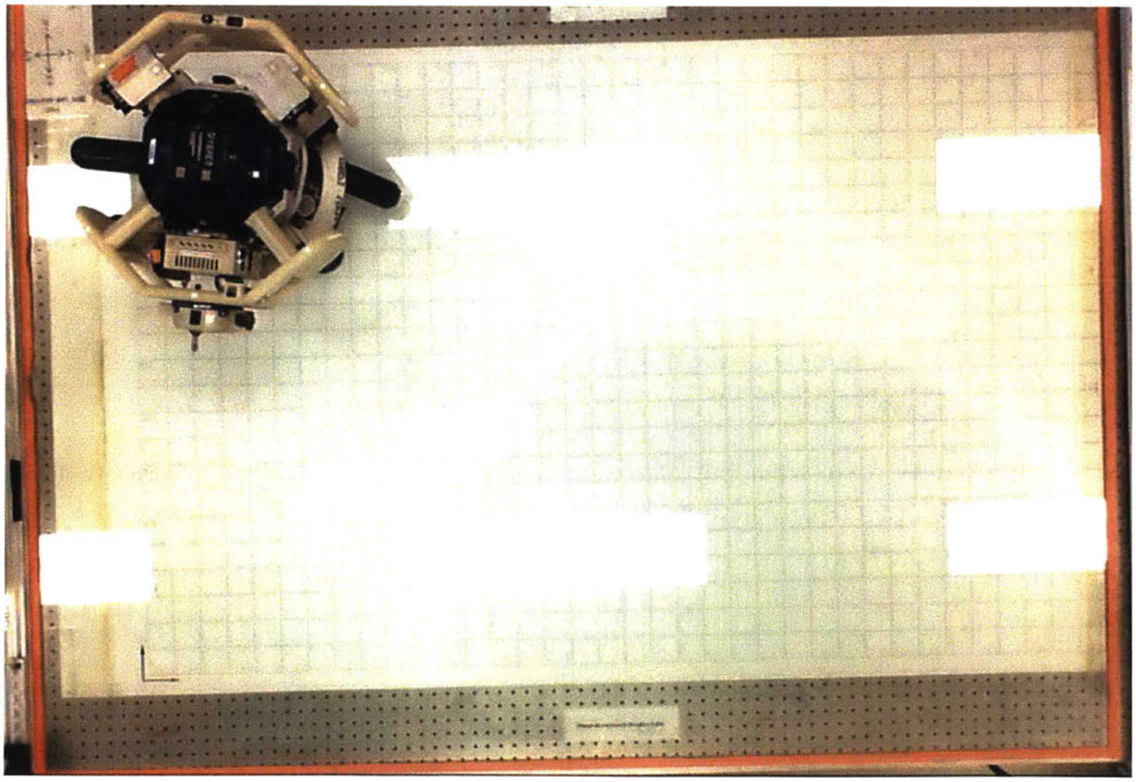


Figure B-4: Baseline PD controller test with hardware configuration #1 — Halo Expansion device equipped with a Universal Docking Port; to be compared with video shown in Fig. B-5. (Video speed: ×10)

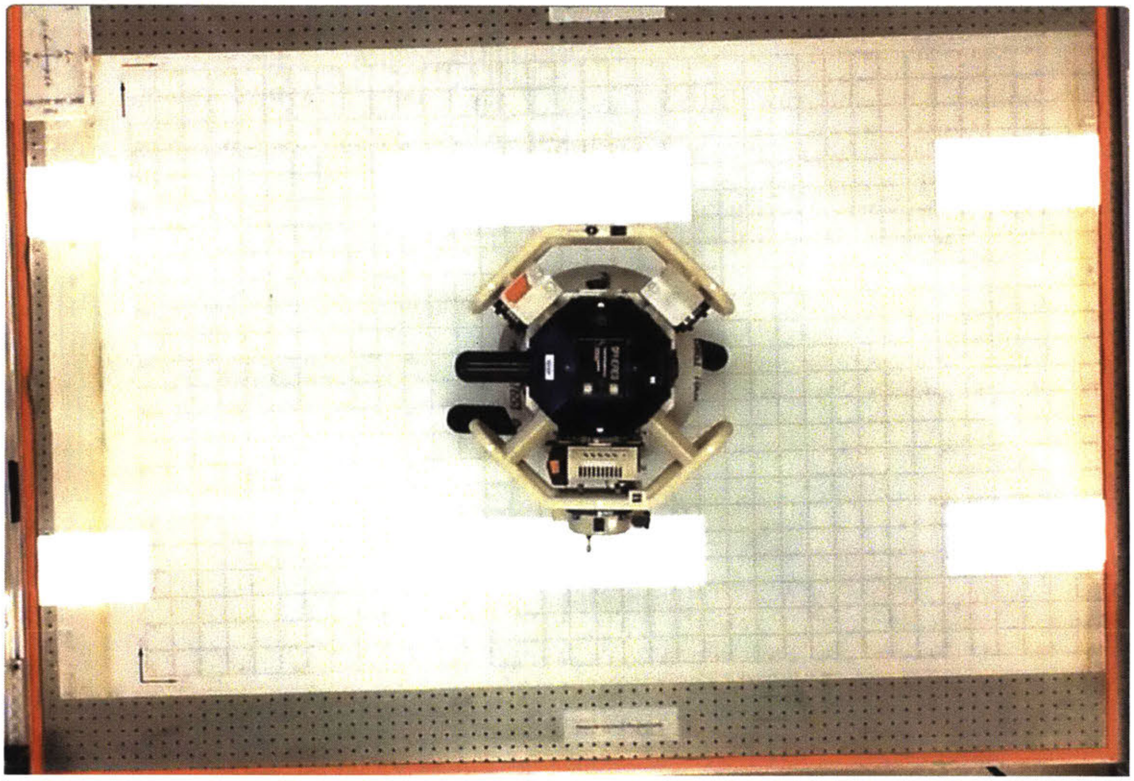
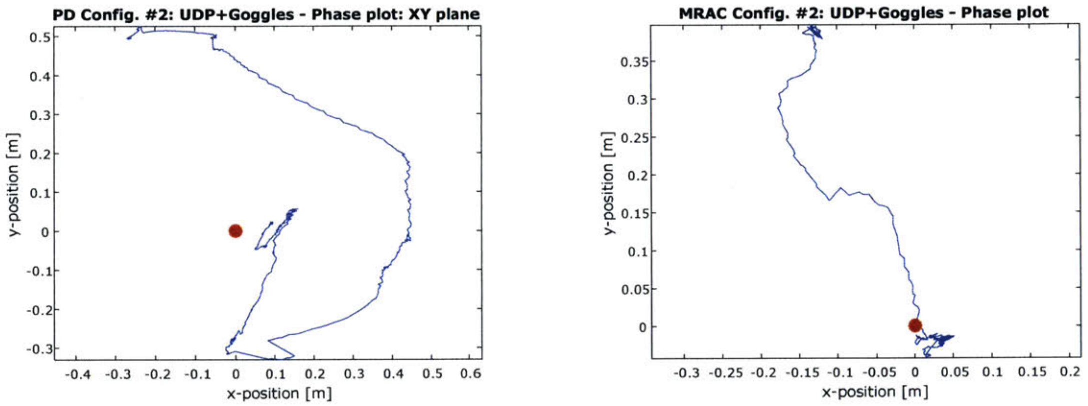


Figure B-5: MRAC test with hardware configuration #1 — Halo Expansion device equipped with a Universal Docking Port; to be compared with video shown in Fig. B-4. (Video speed: ×10)

Similarly, during the MRAC controller's run for configuration #1, shown in Fig. B-5, a large deviation on the yaw angle is induced by the initial translation motion from the glass table's corner to the center of the volume. Nonetheless, the adaptive controller presents a smoother response to these large errors, avoiding the back-and-forth overshoot behavior presented by the PD controller. It is important to remember that the baseline controller was using the best estimate for the physical parameters corresponding to this hardware configuration, while the MRAC was initialized with both its mass and inertia matrix equal to zero.

B.2 Configuration #2



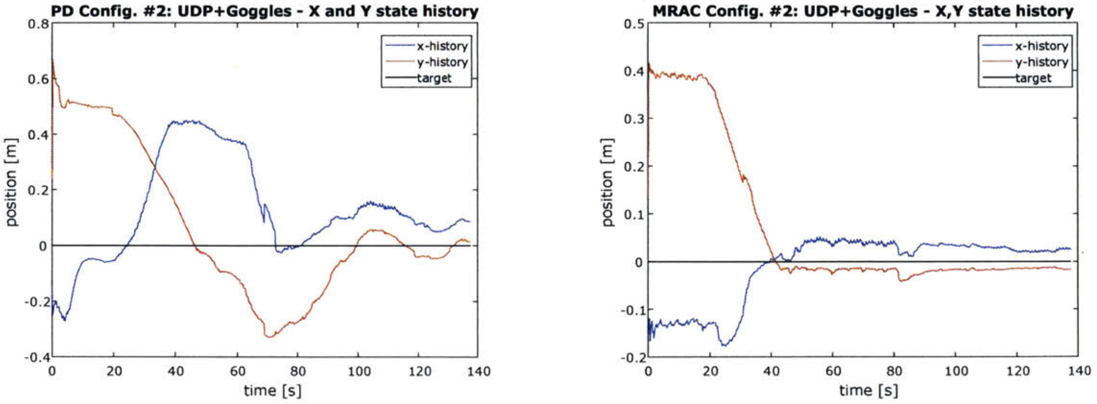
(a) Config.#2 baseline phase plot.

(b) Config.#2 MRAC phase plot.

Figure B-6: Phase plot corresponding to the trajectories taken by the satellite while using the (a) Baseline Proportional-Derivative controller, and (b) the MRAC control law.

For the second hardware configuration, comprised of the previously attached docking port and the addition of a stereo-vision optics mount, Figures B-9 and B-10 contain the test trials' videos.

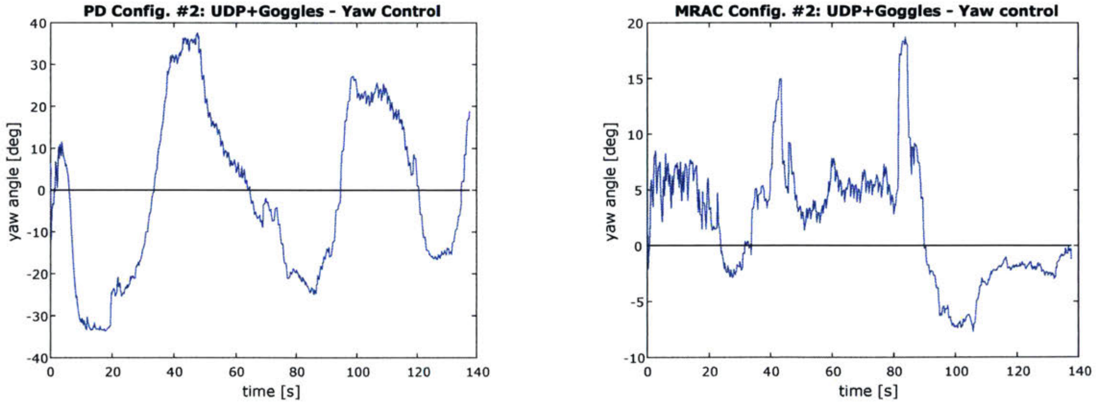
A very similar performance to the one obtained for configuration #1 using the baseline controller is exhibited throughout the test for the PD controller's configuration #2 run, shown in Fig. B-9; large yaw angle deviations while translating, followed by a slight overshoot in the negative y-axis. This time, a more aggressive behavior is observed in attitude, clearly recognizable by the faster oscillation frequency arising



(a) Config.#2 baseline xy-states.

(b) Config.#2 MRAC xy-states.

Figure B-7: X and Y state histories for Configuration #2 under (a) SPHERES PD controller and (b) MRAC controller.



(a) Config.#2 baseline yaw history.

(b) Config.#2 MRAC yaw history.

Figure B-8: Yaw control performance exhibited for the UDP + Optics Mount configuration using the (a) SPHERES PD controller and (b) the MRAC.

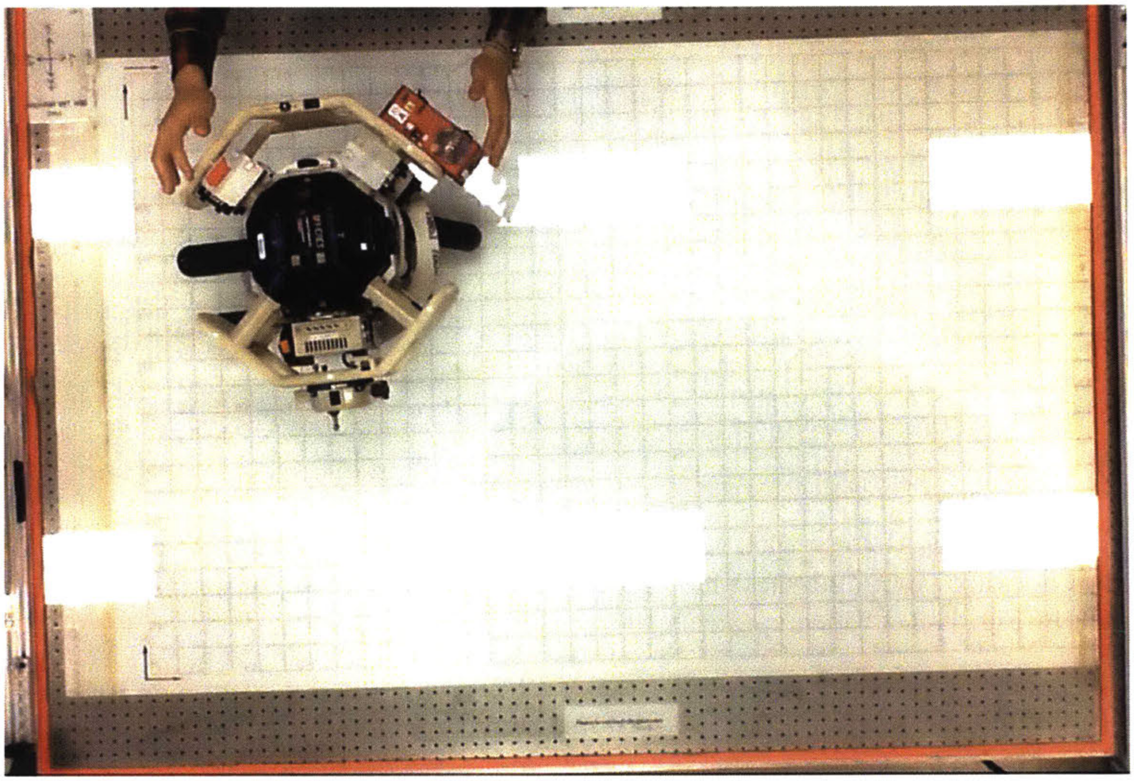


Figure B-9: Baseline PD controller test with hardware configuration #2 — Halo Expansion device with a docking port, and the VERTIGO stereo-cameras; to be compared with video shown in Fig. B-10. (Video speed: ×10) from incorrectly tuned gains (likely higher gains in this case). The SPHERES is able to reach the origin and hold its position within some small ball of a few centimeters.

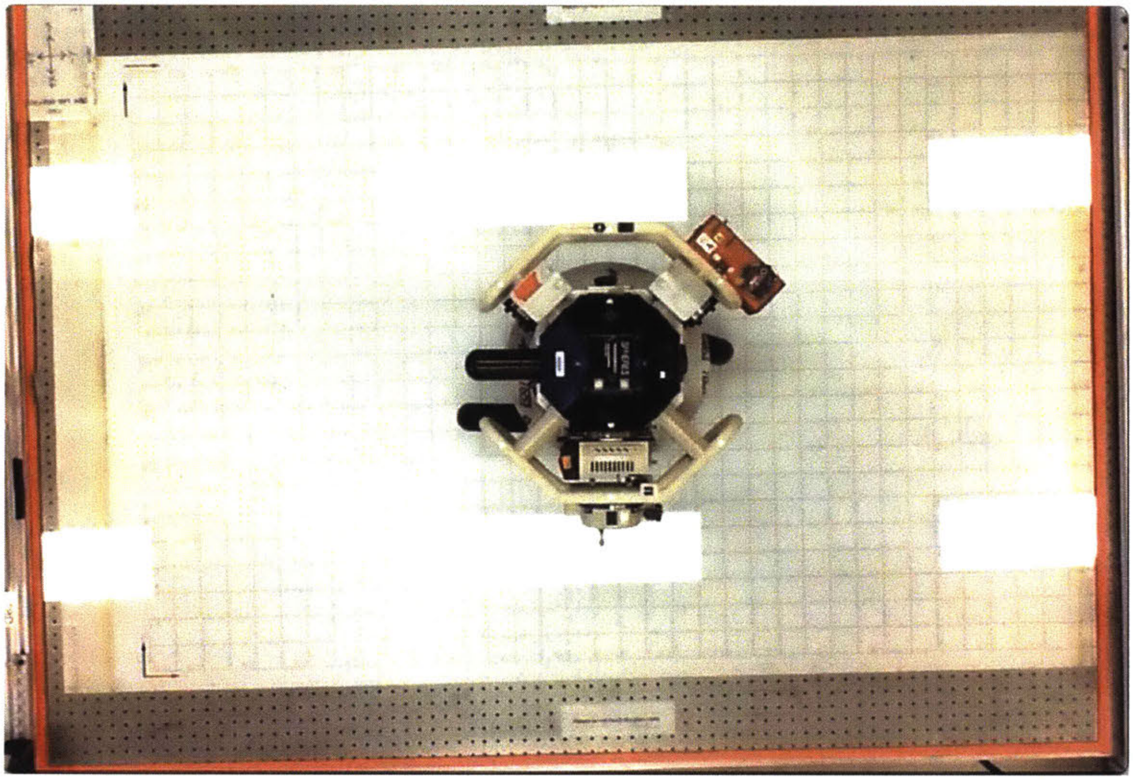
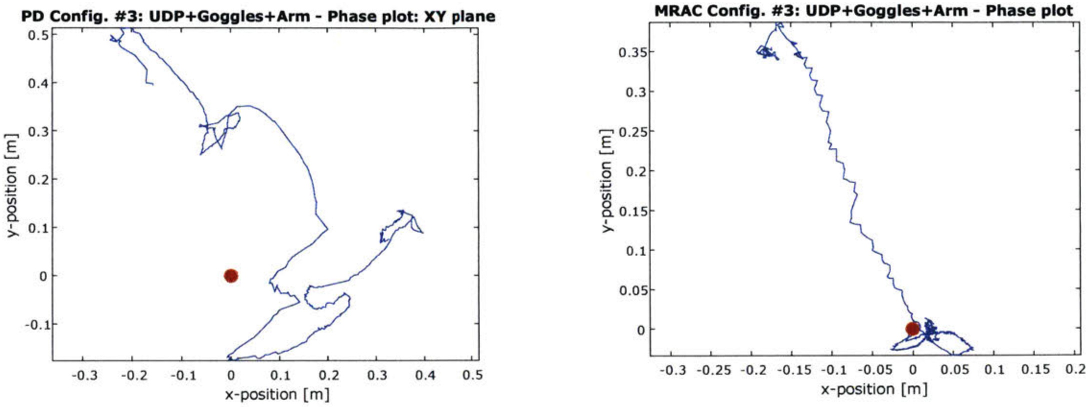


Figure B-10: MRAC test with hardware configuration #2 — Halo Expansion device with a docking port, and the VERTIGO stereo-cameras; to be compared with video shown in Fig. B-9. (Video speed: ×10)

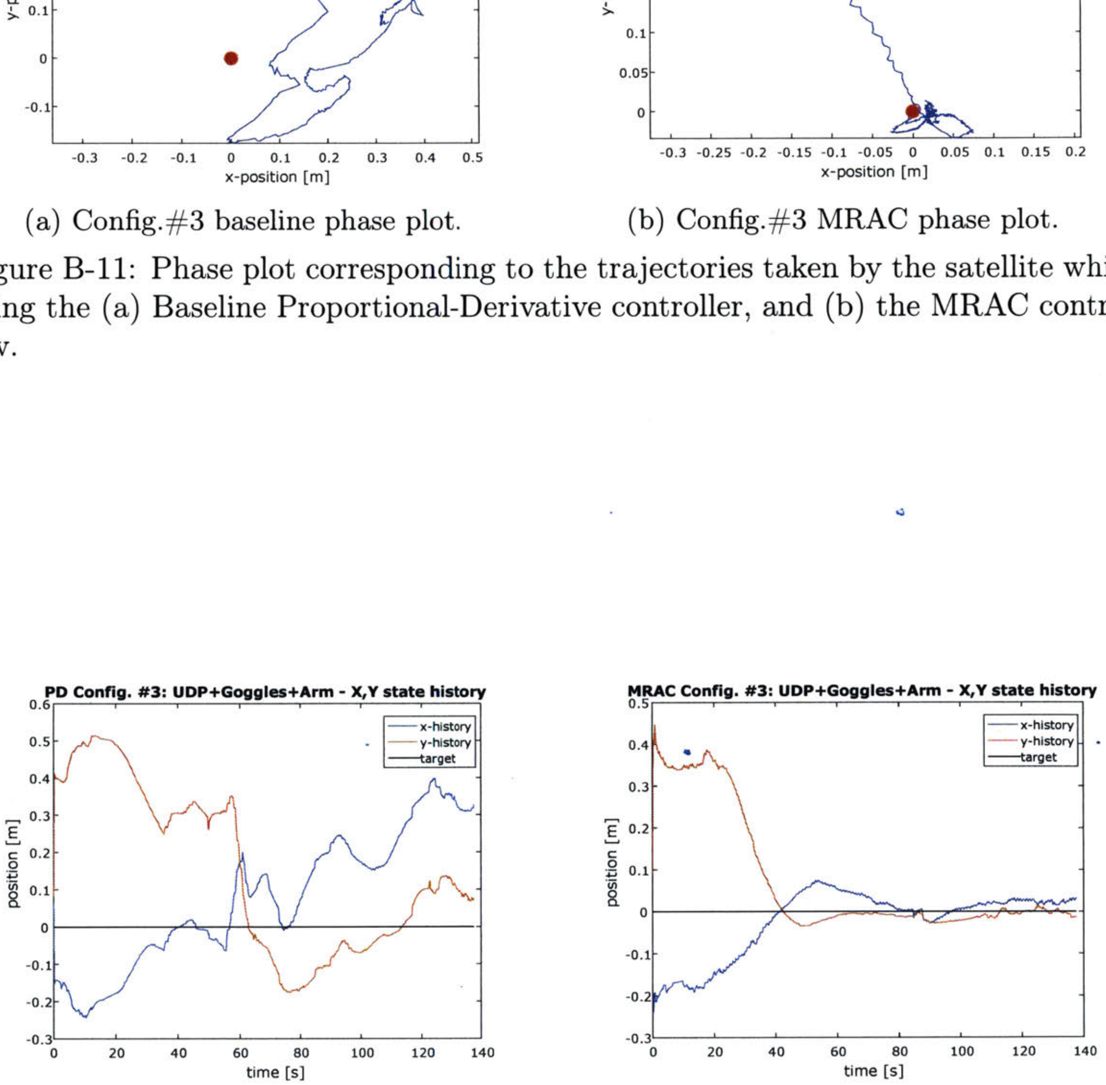
Probably one of the better performing runs throughout the testing trials, the behavior obtained by the adaptive controller using configuration #2, which can be observed in Fig. B-10, is truly superb. The satellite is able to almost perfectly hold its attitude while translating, incurring nearly zero errors in its yaw angle, and once it reaches the origin, to hold its position very steadily. There are some very faint oscillations in the yaw angle during the position hold maneuver, but overall, an incredibly stable performance is exhibited.



(a) Config.#3 baseline phase plot.

(b) Config.#3 MRAC phase plot.

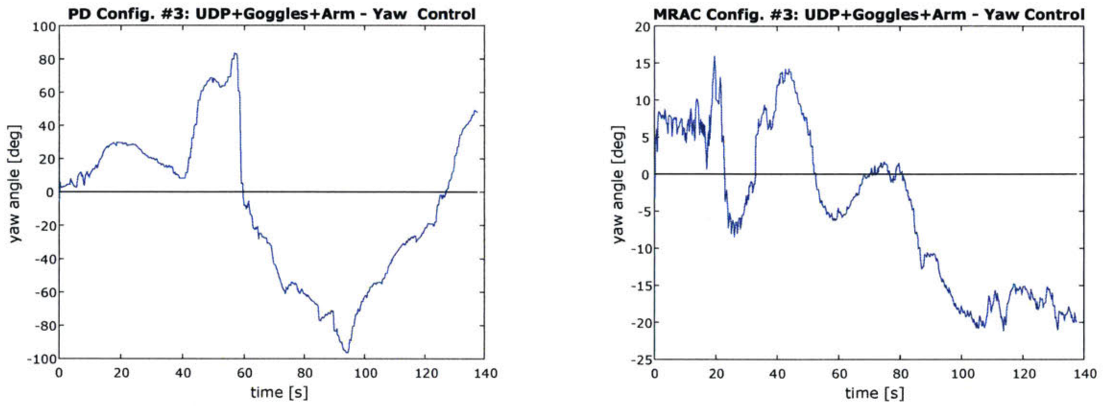
Figure B-11: Phase plot corresponding to the trajectories taken by the satellite while using the (a) Baseline Proportional-Derivative controller, and (b) the MRAC control law.



(a) Config.#3 baseline xy-states.

(b) Config.#3 MRAC xy-states.

Figure B-12: X and Y state histories for Configuration #3 under (a) SPHERES PD controller and (b) MRAC controller.



(a) Config.#3 baseline yaw history.

(b) Config.#3 MRAC yaw history.

Figure B-13: Yaw control performance exhibited for the UDP + Optics Mount + Robotic Arm configuration using the (a) SPHERES PD controller and (b) the MRAC.

B.3 Configuration #3

For the last configuration — which includes the Universal Docking Port, the stereovision optics mount, and a relatively large robotic arm — the performance of both controllers can be observed in Figures B-14 and B-15. Arguably the hardest of the three, given the non-negligible change in the inertia properties induced by the addition of the robotic arm, this test yields drastically different results for the two approaches.

Basically self-explanatory, the video of the baseline PD controller shown in Fig. B-14 makes it clear that given the addition of the latest attachment the system is rendered uncontrollable by the traditional controller's approach. Even though it is able to stop the spinning motion and counteract it, not enough time was alloted to the test for the controller to exhibit converging stability properties.

Surprisingly, the story for the adaptive controller throughout configuration #3's trial run turned out to be quite different. Translation is again achieved with almost no attitude error, and position holding is initially kept at a steady point, quite accurately. After some instants, it seems as if the weight and inertia of the robotic arm start inducing a spin in the yawing axis; the satellite tries to compensate for the error but it cannot completely, and then starts to slowly diverge from the set point. Despite that, the performance difference in comparison to the baseline PD controller is quite

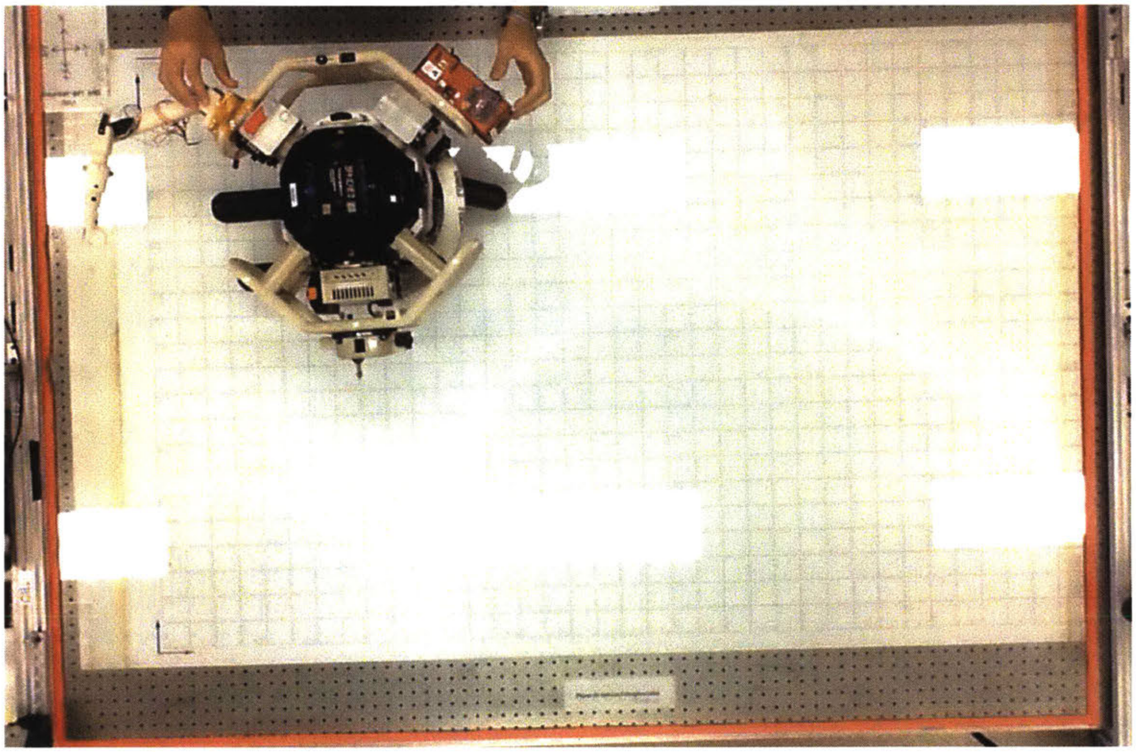


Figure B-14: Baseline PD controller test with hardware configuration #3 — Halo Expansion device with a docking port, optics mount, and a robotic arm; to be compared with video shown in Fig. B-15. (Video speed: ×10)

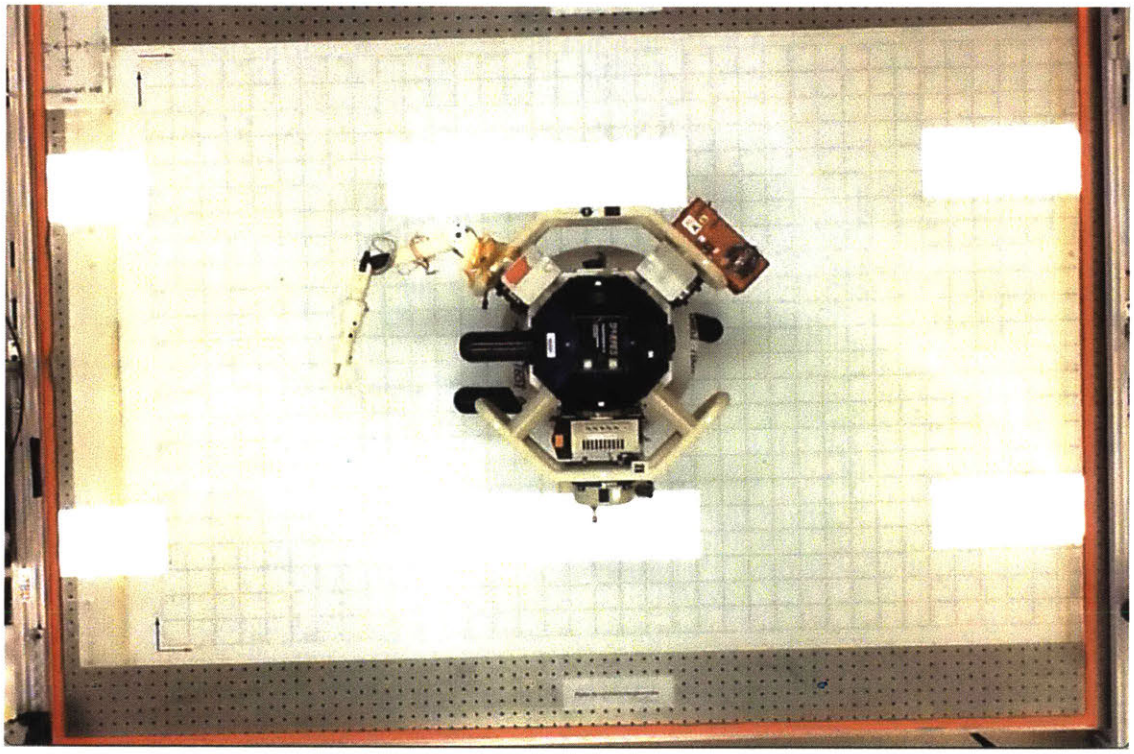
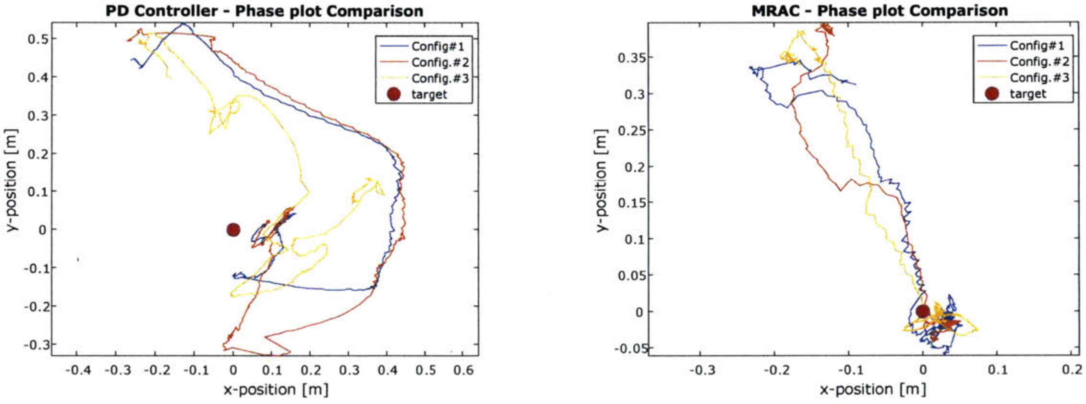


Figure B-15: MRAC test with hardware configuration #3 — Halo Expansion device with a docking port, optics mount, and a robotic arm; to be compared with video shown in Fig.B-14. (Video speed: ×10) significant, yielding its results to be satisfactory.

B.4 Summary Plots

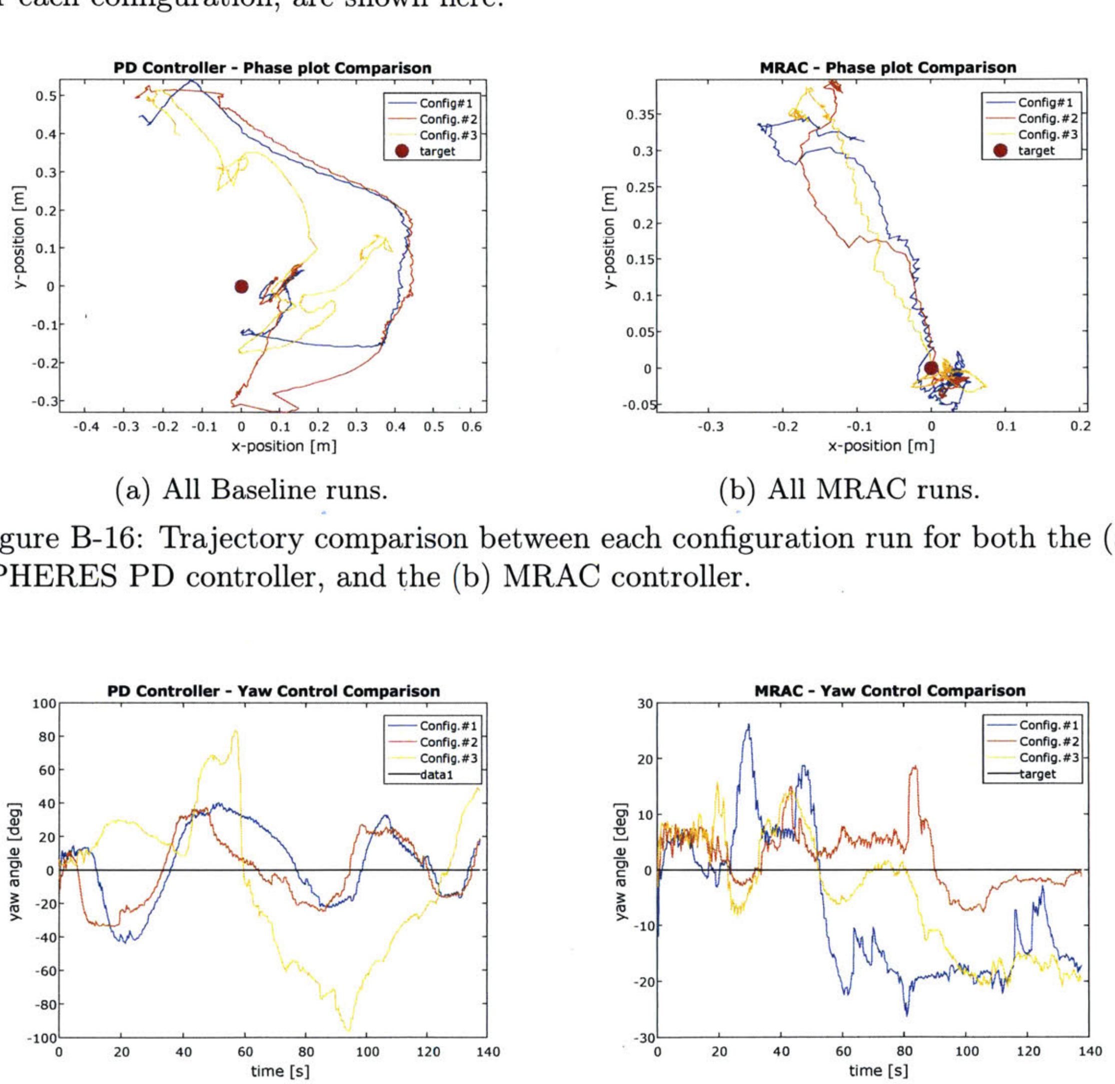
The complete plots detailing the x- and y-state histories, the attitude control performance (yaw angle), and the individual phase plots corresponding to each of the tests, for each configuration, are shown here.



(a) All Baseline runs.

(b) All MRAC runs.

Figure B-16: Trajectory comparison between each configuration run for both the (a) SPHERES PD controller, and the (b) MRAC controller.



(a) All Baseline runs.

(b) All MRAC runs.

Figure B-17: Attitude control performance comparison between each configuration run for both the (a) SPHERES PD controller, and the (b) MRAC controller.

Appendix C

Probability Analysis Derivation

The fact that carrying out autonomous docking maneuvers poses a prerequisite to almost every InSPIRE-II program objective, successfully and robustly accomplishing it has been taken as the only "boolean" metric for these latest test sessions (UDP Science, Halo Checkout, etc.); if one is able to dock, then a successful status is obtained, otherwise, independent to the actions that occurred within the session, a failure is given. This type of standard is both uninformative and useless within the incremental approach that SPHERES Test Sessions follow, in which distinct levels of complexities are handled throughout the course of different sessions to gradually advance towards achieving a said objective.

Another intuitive reason behind the approach is the fact that a chaser satellite does not want to actively track and match the docking axis corresponding to a target satellite's docking face, since at close proximities, where small angle deviations translate into large translation distances for the chaser's center of mass, and limit cycles arising from even the smallest of oscillations expressed by the target would be present. The situation would then only worsen when phase lags due to the discrete nature of modern control systems and effects such as actuator saturation are taken into consideration.

By fitting performance data to a probability density function, such as a Gaussian distribution, a granular improvement metric that assigns a likelihood for achieving a docking maneuver can be obtained (i.e., for assigning some value to how well the current system is behaving).

C.1 «Counselor Model»: Ideas Behind the Heuristic

What this approach aims to perform is to provide the system with a heuristic, that is, some sort of metric or piece of information — derived from the mission objectives, requirements, and system characteristics — from which autonomy and decision making algorithms can draw inferences to choose appropriate and best-informed courses of action.

The approach will be referred to as the "Counselor Model", as it will provide advices but will not have any decision making capabilities. The idea behind the concept is pretty intuitive, and a high level overview of is presented in Fig. C-1.

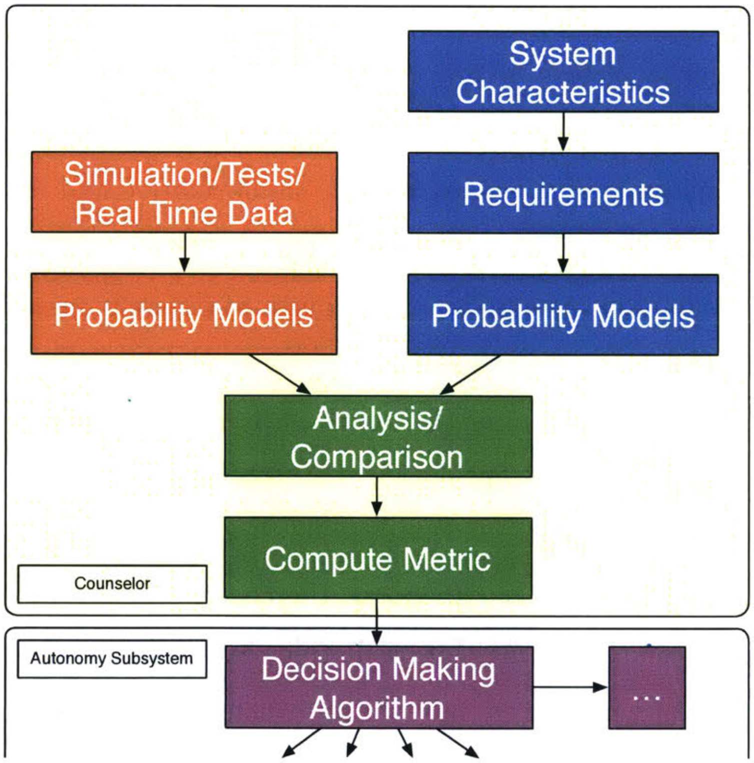


Figure C-1: Structure and building blocks upon which the "Counselor Model" is based.

By having some specifications describing the characteristics of a system, i.e., physical properties, performance capabilities, etc., it possible to develop a set of requirements that specify the attributes warranted for the success of a certain objective. This deterministic group of statements can then be translated into a probability model that accounts for the variation that the system is able to tolerate and still achieve a mission success outcome.

Concurrently, data regarding the controllable aspect of the system is to be analyzed, and the useful information that is able to be gathered needs to furthermore be fitted to second probability model. This allows for some probabilistic comparison to be performed upon the two models, and a metric to be computed given the two pieces of information. The point of computing this additional heuristic is to then pass it along to the autonomy subsystem of the control module for additional evaluations to be made available.

C.2 Example using InSPIRE-II Data

The application of this approach is demonstrated by using data obtained from the InSPIRE-II on-orbit test sessions.

System Characteristics The first step is to analyze the characteristics of the system given the mission objectives. In this case — since docking is taken as the ultimate objective — the docking port characteristics of the UDP, as well as the SPHERES ensemble physical dimensions are taken into account; the dimensions of the docking port's lance and hole pair are shown in Fig. C-2, and are defined as the hole's diameter $d_{udp} = 2.28\ cm$ and radius $r_{udp} = 1.14cm$.

Along with the dimensions imposed by the mechanical interface, the entire SPHERES ensemble needs to then be taken into account. A high-level pictorial overview along with important distances is presented in Fig. C-3. It is of importance to know the vector connecting the geometric center of the satellite (which is used to localize the craft in the ISS reference frame using global metrology) to the tip of the docking port's lance; this distance is denoted as geometric center to lance tip GC2LT.

Additionally, assuming that the chaser satellite is aligned along the target satel-

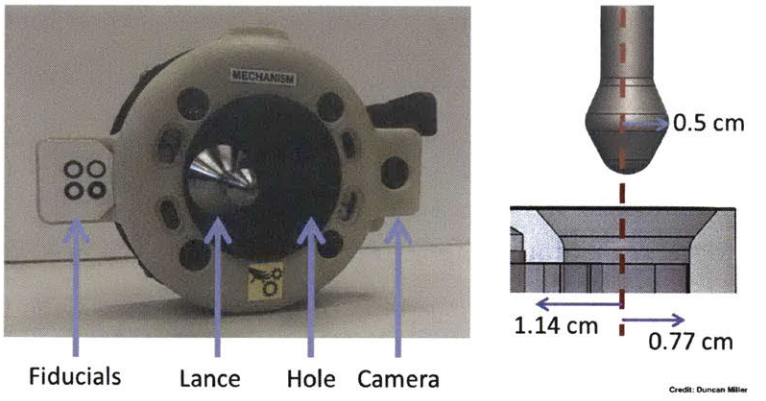


Figure C-2: The SPHERES Universal Docking Port (UDP) along with the physical dimensions of the lance and hole pair.

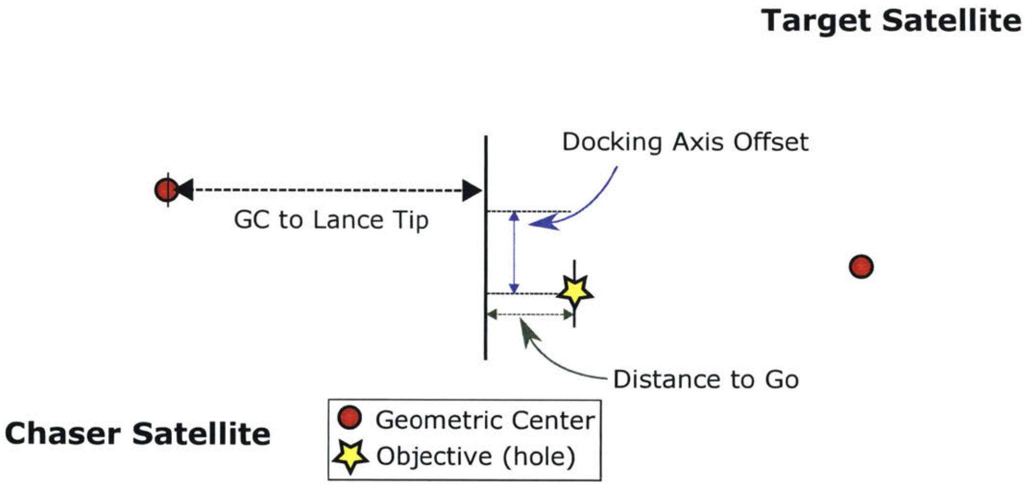


Figure C-3: Diagram depicting a Chaser and a Target Satellite, and characteristic distances for analyzing the docking problem.

lite's docking axis, the distance to go (D2G) can be taken to be the distance between the chaser's lance tip to the target's docking port face (green distance in Fig. C-3). Lastly, in case that the satellites are correctly aligned in attitude, but an offset between the GC2LT axis and the target's docking axis exists, the distance to be corrected will be denoted as the docking axis offset (DockAx_Off, blue distance in Fig. C-3).

After introducing these variables, an approximation of how the distances are related can be formed. The relationships are outlined in Fig. C-4, where

$$\ell \equiv \texttt{GC2LC} + \texttt{D2G} \,,$$

and given the fact that $\ell \gg \mathbf{r}$ ($\ell \approx 35\ cm$ vs $\mathbf{r} \approx 1\ cm$), the approximation $\mathbf{r} \approx r_{udp}$ can be taken. A value for the maximum angle deviation allowed (θ_{max}) for the chaser to still perform a successful docking maneuver is then calculated as

$$\theta_{max} = \frac{\mathbf{r}}{\ell} \approx \frac{r_{udp} = 1.14\ cm}{\texttt{GC2LC} + \texttt{D2G}},$$

and with the value for ℓ at very close proximities (chaser waiting in front of the target to finalize the docking maneuver) being roughly $\texttt{GC2LC} + \texttt{D2G} \approx 35\ cm$, a maximum deviation angle of $\theta_{max} = 1.866^o$ is obtained.

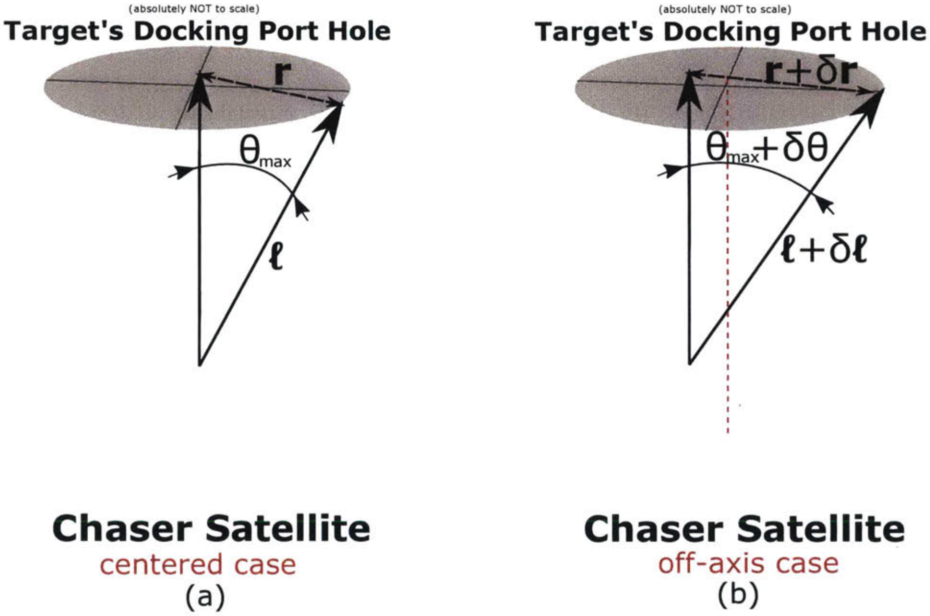

(absolutely NOT to scale)
Target's Docking Port Hole
r
θmax
ℓ
Chaser Satellite
centered case
(a)
(absolutely NOT to scale)
Target's Docking Port Hole
r+δr
θmax+δθ
ℓ+δℓ
Chaser Satellite
off-axis case
(b)

Figure C-4: Diagram depicting the maximum allowed angle deviation in the chaser's attitude for docking. A (a) centered case and an (b) off-axis case are presented, given that docking can still be achieved with the latter conditions.

However, as it is depicted in Fig. C-4(b), a successful docking maneuver can also occur even whenever the chaser satellite is slightly off-axis, that is, whenever a small value for `DockAx_Off` is present. This signifies that slightly higher tolerances on the allowed values for the possible deviations can be alloted, which are represented by the $\delta\mathbf{r}, \delta\theta$, and $\delta\ell$ terms in Fig. C-4(b). By including this variations, the new value for the maximum allowed deviation would be $\theta_{max} = 1.5\ cm/35\ cm \approx 2.5^o$.

Requirements After the system's characteristics have been analyzed, requirements spurred by the obtained specifications can be fabricated. In this case, a maximum deviation angle θ_{max} at the last stage of the approach maneuver conducted by the chaser satellite was obtained. This means that, at the time of docking, an attitude requirement can be imposed upon the chaser's control performance:

- *At the time of docking, the chaser satellite shall present attitude deviations no greater than* 2.5^o *from the target's docking axis*, or
- *At the time of docking, the chaser satellite shall attitude control errors no greater than* 2.5^o*, as measured from the target's docking axis.*

Given that this requirement applies to the scenario where the chaser satellite is at very close proximity to the target and ready to perform the final docking maneuver, by performing a derivation approach it can be directly translated into a lower-level requirement that specifies an attitude control performance demand for the satellite's position-hold capabilities; for instance

- *The chaser's position-holding maneuver shall yield attitude errors of less than* 2.5^o *in all axes.*

Requirement-Driven Probability Model After crafting the necessary set of requirements, a probability model can be built on top of its specifications; for this example, normal distributions will be employed.

By reasoning about the problem in a stochastic manner, this latter requirement can be posed as a Gaussian probability density function (pdf); instead of using a strict bound, probability masses can be employed. Hence, a Gaussian pdf with a $3\sigma = \theta_{max}$ parameter can be utilized in lieu of the requirement. By setting the θ_{max} value to be three times the normal distribution's standard deviation (σ), it is being specified that at least 99.7% (basically ALL) of the attitude error performance data points obtained should be lower than said value. This would yield a probability model encoded with the requirements and described by a

$$\text{target distribution} :\sim \mathcal{N}\left(\mu = 0, \sigma = \frac{\theta_{max}}{3}\right),$$

which is pictorially shown in Fig. C-5 along with the less stringent target distributions with variance defined as $2\sigma = \theta_{max}$ and $\sigma = \theta_{max}$.

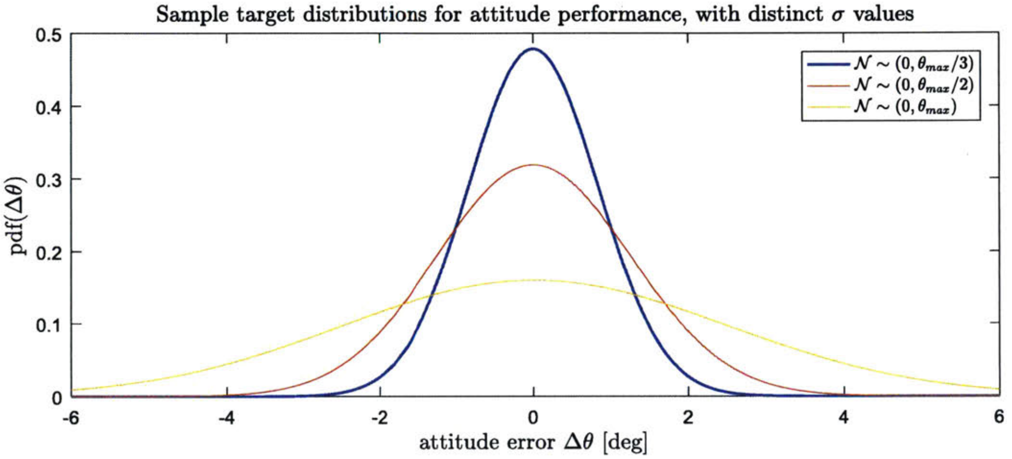


Figure C-5: Sample target distributions obtained from the attitude control performance requirement through the strategy depicted in the block diagram from Fig. C-1.

Test/Real Time Data → Probability Model Now that the requirements have been translated into probability models, test data or real time data from the actual system needs to be adequately analyzed, as depicted in Fig. C-1. Given that requirements have been imposed into the position-holding maneuvers of the chaser satellite, data from those maneuvers performed during Test Session 86: Halo Checkout and Test Session 92: Docking Port Science 1.5 is utilized.

Each maneuvers for both test sessions was analyzed and data from the satellites' performance, after transients, was included into the analysis; some of the test data utilized for the analysis is shown in Fig. C-6. The objective now is to fit all of this information into a probability model, and by determining the statistics corresponding to the error between attitude tracking commands and references in each of the three attitude degrees of freedom, a probability distribution is able to be determined for the roll, pitch, and yaw control of the two ensembles (SPHERES + VAS + UDP and SPHERES + Halo and friends).

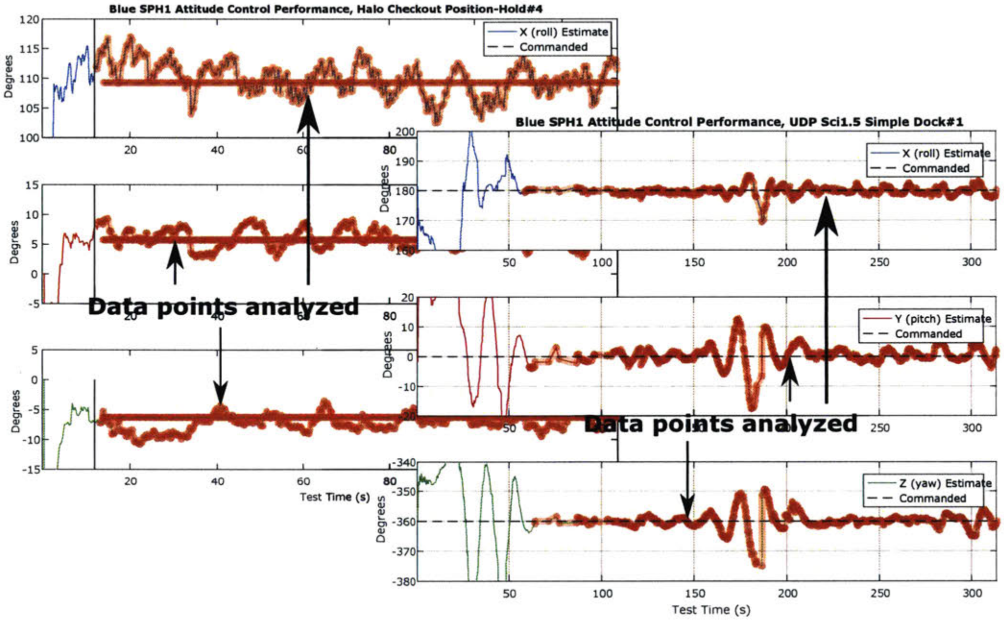


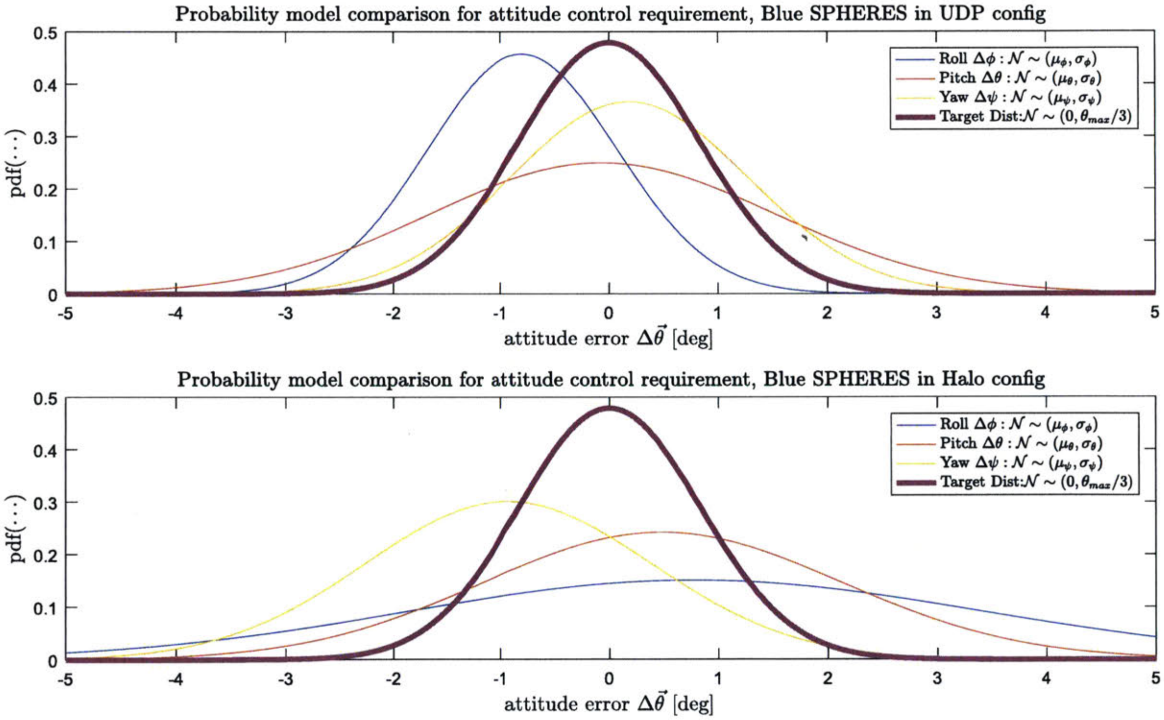
Figure C-6: Two sample data sets analyzed for the Blue SPHERES attitude control error performance.

Analysis/Comparison The error variance and standard deviation from the collected data sets can then be compared against the target distribution corresponding to the tolerable errors; just for recap, this latter one was defined as the distribution that has as a 3σ value the maximum allowed angle deviation under which docking can still be successful, taking into consideration that docking can also take place even when the SPHERES are off-axis.

Thus, one probability model for each SPHERE (Blue and Orange) for each configuration (Halo and UDP) should be able to be obtained. Unfortunately, there were no successful nor acceptable position-holding tests for the Orange SPHERE during TS92, so no model was able to be calculated for the Orange SPH + UDP hardware configuration.

After obtaining the probability models, the comparison with the target distribution can be performed. It is important to remember that what the target distribution expresses is the area on which the majority of the collected attitude data points prior to the docking maneuver (e.g., position holding right in front of the target) must lie in order to probabilistically say that the docking maneuver will succeed. It is completely dependent on the mechanical properties of the docking ports and the Halo and UDP configuration (distance from GC to lance tip, hole dimension, etc).

The comparison for the Blue satellite in its two configurations is presented in Fig. C-7; it can be seen that the best controlled axis for the UDP configuration is in the roll ϕ direction, followed by the yaw ψ axis, yielding the pitch θ degree of freedom as the most poorly controlled one (widest pdf). Still, by just observing the distributions' first moment, it can also be seen that the roll ϕ direction is the most shifted one, while the pitch θ aligns almost perfectly with the target distribution's mean (meaning that the mean attitude error in pitch is very close to zero).


Probability model comparison for attitude control requirement, Blue SPHERES in UDP config

Probability model comparison for attitude control requirement, Blue SPHERES in Halo config

Figure C-7: Probability model comparison for the Blue SPHERES attitude control performance in its UDP (top) and Halo (bottom) configuration.

Contrastingly, for Blue's Halo configuration, the worst controlled axis is the roll ϕ. The pitch θ angle in this case turns out to be in second place, and is represented by a curve almost identical to the one for the UDP configuration, except for a slight mean μ_θ shift. The best controlled attitude degree of freedom for the Halo device and Blue SPHRES pair is the yaw ψ angle, with a slight better performance than in pitch, but further away from the target's mean.

The comparison between the probability models obtained for the Orange SPHERES is shown in Fig. C-8, in which only the Halo configuration is plotted. From these results, it is quite interesting to observe that the roll ϕ control distribution for the Halo configuration is almost identical to the target distribution, not only in width (variance), but also in its center point ($\mu_{\psi} \approx 0^{o}$!). This signifies that the Orange satellite should, in expectation, have no problem with the docking maneuver in terms of controlling its roll ϕ attitude, since it accurately satisfies the derived requirement.

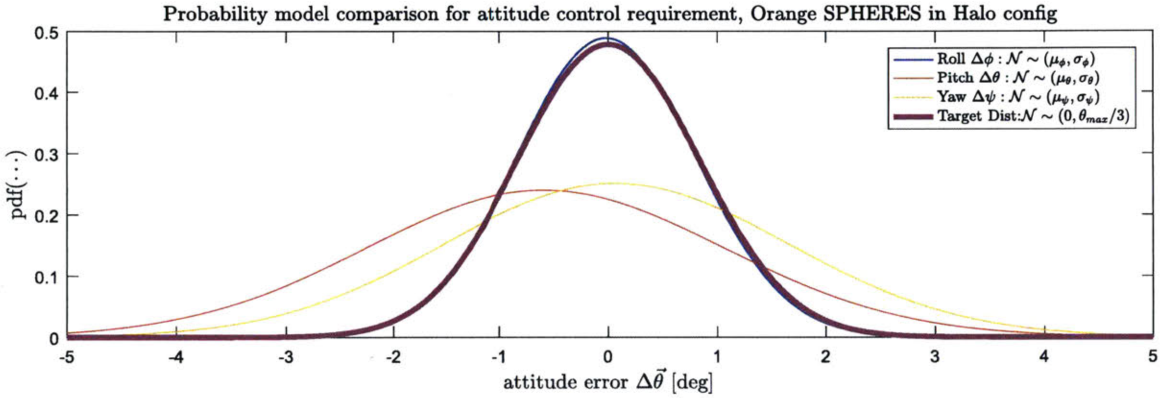


Figure C-8: Probability model comparison for the Orange SPHERES attitude control performance in its Halo configuration; no useful UDP configuration data was obtained from TS92.

This result is very informative, since this same attitude axis could not be correctly controlled, at any moment, by the Orange and UDP configuration during TS92. The possibility that Orange's actuation capabilities were the culprit for the instabilities observed during this latter test session can be ruled out, leaving as possible failure causes both the poor system characterization of the ensemble and the control and path planning algorithms.

Another aspect that can be easily assessed is the difference in degree of controllability between the Halo and UDP configuration. For this, it is particularly informative to contrast the Orange SPHERES probability models to the ones for the Blue SPHERES. The comparison is shown in Fig. C-9.

Just by inspection, it can easily be seen that the control authority exerted upon

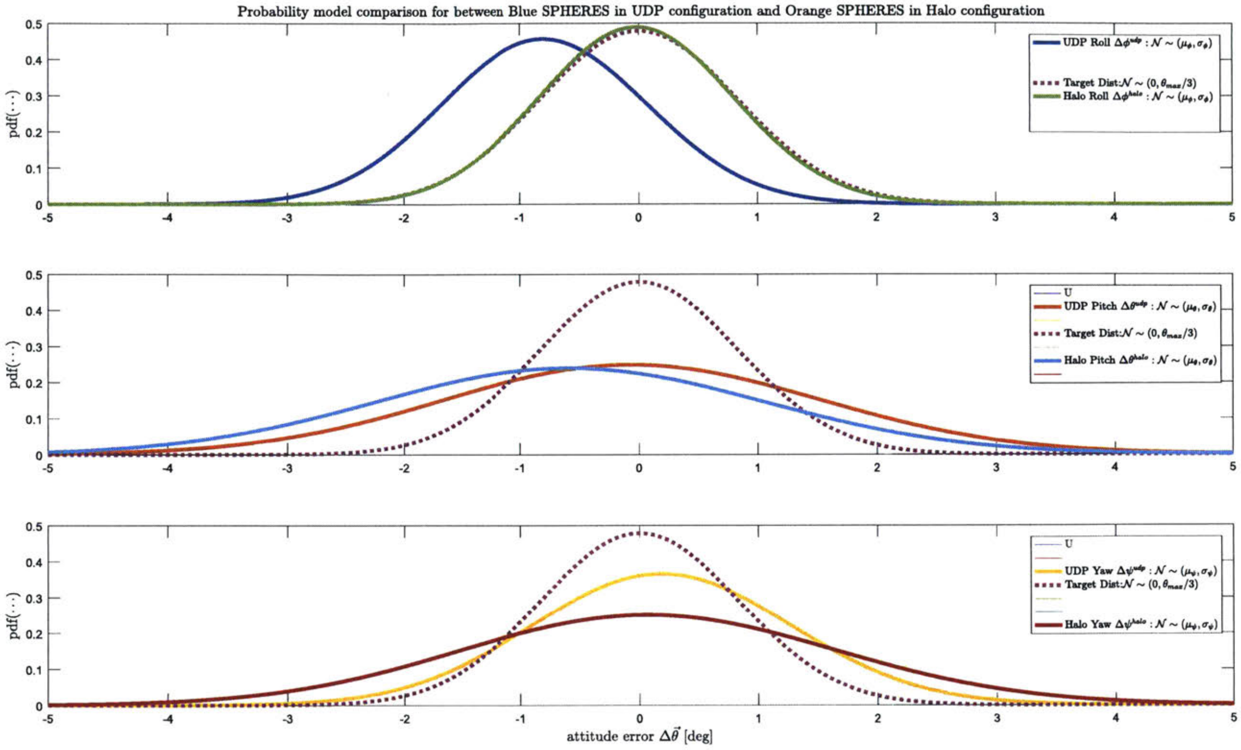


Figure C-9: Probability model comparison between the two hardware configurations, in all three attitude degrees of freedom; the Blue SPHERES plus UDP configuration is assessed against the Orange SPHERES plus Halo configuration.

both systems is very comparable; in the roll ϕ direction, the Halo system is in fact much better controlled than the UDP ensemble. The pitch θ angle is in both cases the worst controlled degree of freedom, represented by a very wide distribution; for the UDP system, the probability model appears to be more aligned with the target distribution, meaning that the mean $\Delta\theta$ errors for the Blue satellite are closer to zero. The only clear difference is noticed in the yaw ψ angle, in which there is a better performance exhibited by the UDP system.

Overall, it is not particularly clear which configuration is more controllable, although it seems that the UDP system might be marginally better behaved. This assessment, paired with the fact that for TS86b no previous system identification tests were performed for the Halo system (best estimate parameter values from CAD were used for inertia, mass, center of mass, etc, as no better guesses were available) allows for the conclusion that the Halo ensemble shows more promise in terms of performance improvement. This follows ones intuition, as the Halo ensemble's mass is much more evenly distributed along the XY-plane, and even though more massive, it can likely be rendered much more controllable than the UDP ensemble.

A clear takeaway from this approach is the fact that if some guarantees on the likelihood of docking success are sought to be given, then an increase in control performance is warranted. In terms of the probability models, this means the reshaping of the distributions in order to increase a certain degree-of-freedom metric score.

Bibliography

[1] "Space Robotics: IEEE Robotics & Automation Society," http://www.ieee-ras.org/space-robotics, accessed: 2017-05-10.

[2] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[3] NASA, "Technology roadmaps: Robotics and autonomous systems," National Aeronautics and Space Administration, Tech. Rep. TA 4, July 2015.

[4] F. Aghili, "A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 634–649, 2012.

[5] B. E. Tweddle, A. Saenz-Otero, J. J. Leonard, and D. W. Miller, "Factor graph modeling of rigid-body dynamics for localization, mapping, and parameter estimation of a spinning object in space," *Journal of Field Robotics*, vol. 32, no. 6, pp. 897–933, 2015.

[6] C. Henshaw, L. Healy, and S. Roderick, "Liive: A small, low-cost autonomous inspection vehicle," in *AIAA SPACE 2009 Conference & Exposition*, 2009, p. 6544.

[7] A. Teran and C. Jewison, "Performance analysis of distinct control and estimation frameworks for spacecraft proximity operations involving severe uncertainties (accepted for publication)," in *Proceedings of 10th ESA International Guidance Navigation and Control Conference.* ESA, 2017.

[8] K. J. Åström and P. Eykhoff, "System identificationa survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[9] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 423–438, 2004.

[10] J. Slotine and W. Li, *Applied Nonlinear Control*, ser. Prentice-Hall International Editions. Prentice-Hall, 1991. [Online]. Available: https://books.google.com/books?id=HddxQgAACAAJ

[11] L. Ljung, "Prediction error estimation methods," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 11–21, 2002.

[12] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.

[13] A. Saenz-Otero and D. W. Miller, "Spheres: a platform for formation-flight research," in *Optics & Photonics 2005.* International Society for Optics and Photonics, 2005, pp. 58 990O–58 990O.

[14] A. Saenz-Otero, "Design principles for the development of space technology maturation laboratories aboard the international space station," *Massachusetts Institute of Technology, Department of Aeronautics and Astronatucis, Ph. D. Thesis, Cambridge, MA*, 2005.

[15] B. P. McCarthy, "Flight hardware development for a space-based robotic assembly and servicing testbed," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[16] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[17] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

[18] Z. Xu, N. Qi, and Y. Chen, "Parameter estimation of a three-axis spacecraft simulator using recursive least-squares approach with tracking differentiator and extended kalman filter," *Acta Astronautica*, vol. 117, pp. 254–262, 2015.

[19] D. Kim, S. Yang, and S. Lee, "Rigid body inertia estimation using extended kalman and savitzky-golay filters," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[20] W. Li, D. Wang, and C. Liu, "System identification of large flexible appendage on satellite for autonomous control," *IEEE International Conference on Control and Automation, ICCA*, pp. 724–728, 2013.

[21] M. Kiani and S. H. Pourtakdoust, "Spacecraft attitude and system identification via marginal modified unscented Kalman Filter utilizing the sun and calibrated three-axis-magnetometer sensors," vol. 21, pp. 1451–1460, 2014.

[22] P. R. Chandler, M. Pachter, and M. Mears, "System-Identification for Adaptive and Reconfigurable Control," *Journal of Guidance Control and Dynamics*, vol. 18, no. 3, pp. 516–524, 1995.

[23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics.* MIT press, 2005.

[24] N. V. Hoffer, C. Coopmans, A. M. Jensen, and Y. Chen, "A survey and categorization of small low-cost unmanned aerial vehicle system identification," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, p. 129, 2014.

[25] J. Kocijan, "System identification with gp models," in *Modelling and Control of Dynamic Systems Using Gaussian Process Models.* Springer, 2016, pp. 21–102.

[26] B. Mu, S.-Y. Liu, L. Paull, J. Leonard, and J. P. How, "Slam with objects using a nonparametric pose graph," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on.* IEEE, 2016, pp. 4602–4609.

[27] A. Venkatraman, W. Sun, M. Hebert, B. Boots, and J. A. Bagnell, "Inference machines for nonparametric filter learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 2016, 2016.

[28] A. K. Tanwani and S. Calinon, "Small variance asymptotics for non-parametric online robot learning," *arXiv preprint arXiv:1610.02468*, 2016.

[29] P. A. Ioannou and E. B. Kosmatopoulos, *Adaptive control.* Wiley Online Library, 2006.

[30] G. Tao, *Adaptive control design and analysis.* John Wiley & Sons, 2003, vol. 37.

[31] C. Tsiligiannis and S. Svoronos, "Multivariable self-tuning control via the right interactor matrix," *IEEE transactions on automatic control*, vol. 31, no. 10, pp. 987–989, 1986.

[32] P. W. Gibbens, C. A. Schwartz, and M. Fu, "Achieving diagonal interactor matrix for multivariable linear systems with uncertain parameters," *Automatica*, vol. 29, no. 6, pp. 1547–1550, 1993.

[33] W. Kase and Y. Mutoh, "A design of discrete-time multivariable iterative learning control systems with interactor matrix," 2010.

[34] S. Li and G. Tao, "Discrete-time multivariable mrac schemes with output sensor uncertainty compensation," in *Decision and Control (CDC), 2010 49th IEEE Conference on.* IEEE, 2010, pp. 1761–1766.

[35] D. Maiti, J. Guo, and G. Tao, "A discrete-time multivariable state feedback mrac design with application to linearized aircraft models with damage," in *American Control Conference (ACC), 2011.* IEEE, 2011, pp. 606–611.

[36] T. Yucelen and A. J. Calise, "Adaptive control with loop transfer recovery: a kalman filter approach," in *Proceedings of the 2011 American Control Conference.* IEEE, 2011, pp. 1736–1741.

[37] T. C. Nguyen-Huynh and I. Sharf, "Adaptive reactionless motion for space manipulator when capturing an unknown tumbling target," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4202–4207, 2011.

[38] ——, "Adaptive Reactionless Motion and Parameter Identification in Postcapture of Space Debris," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 404–414, 2013. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.57856

[39] S. Ulrich, A. Saenz-Otero, and I. Barkana, "Passivity-based adaptive control of robotic spacecraft for proximity operations under uncertainties," *Journal of Guidance, Control, and Dynamics*, pp. 1444–1453, 2016.

[40] P. Singla, K. Subbarao, and J. L. Junkins, "Adaptive output feedback control for spacecraft rendezvous and docking under measurement uncertainty," *Journal of guidance, control, and dynamics*, vol. 29, no. 4, pp. 892–902, 2006.

[41] K. Zhang and M. A. Demetriou, "Synthesis of adaptive controllers for spacecraft rendezvous maneuvers using nonlinear models of relative motion," in *Decision and Control (CDC), 2016 IEEE 55th Conference on.* IEEE, 2016, pp. 395–400.

[42] P. Fortescue, G. Swinerd, and J. Stark, *Spacecraft systems engineering.* John Wiley & Sons, 2011.

[43] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems.* CRC press, 2011.

[44] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers.* Princeton university press, 2010.

[45] L. Desborough and R. Miller, "Increasing customer value of industrial control performance monitoring-honeywell's experience," in *AIChE symposium series.* New York; American Institute of Chemical Engineers; 1998, 2002, pp. 169–189.

[46] K. Großekatthöfer and Z. Yoon, "Introduction into quaternions for spacecraft attitude representation," *TU Berlin*, vol. 16, 2012.

[47] B. Wie, *Space Vehicle Dynamics and Controls.* American Institute of Aeronautics and Astronautics, 2008, ch. 7.6: Optimal Jet Selection.

[48] B. Wie and P. M. Barba, "Quaternion feedback for spacecraft large angle maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 3, pp. 360–365, 1985.

[49] P. A. Ioannou and J. Sun, *Robust adaptive control.* Courier Corporation, 2012.

[50] J.-J. E. Slotine, "Sliding controller design for non-linear systems," *International Journal of control*, vol. 40, no. 2, pp. 421–434, 1984.

[51] D. E. Kirk, *Optimal control theory: an introduction.* Courier Corporation, 2012.

[52] C. M. Jewison, "Guidance and control for multi-stage rendezvous and docking operations in the presence of uncertainty," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA 02139, 5 2017.

[53] R. Shishko and R. Aster, "Nasa systems engineering handbook," *NASA Special Publication*, vol. 6105, 1995.

[54] A. S. Otero, A. Chen, D. W. Miller, and M. Hilstad, "Spheres: Development of an iss laboratory for formation flight and docking research," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 1–1.

[55] C. P. Mandy, H. Sakamoto, A. Saenz-Otero, and D. W. Miller, "Implementation of satellite formation flight algorithms using spheres aboard the international space station," 2007.

[56] S.-J. Chung, J.-J. E. Slotine, and D. W. Miller, "Nonlinear model reduction and decentralized control of tethered formation flight," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, pp. 390–400, 2007.

[57] S. Nolet and D. W. Miller, "Autonomous docking experiments using the spheres testbed inside the iss," in *Defense and Security Symposium.* International Society for Optics and Photonics, 2007, pp. 65 550P–65 550P.

[58] J. G. Katz, A. Saenz-Otero, and D. W. Miller, "Development and demonstration of an autonomous collision avoidance algorithm aboard the iss," in *Aerospace Conference, 2011 IEEE.* IEEE, 2011, pp. 1–6.

[59] B. Wie, H. Weiss, and A. Arapostathis, "Quarternion feedback regulator for spacecraft eigenaxis rotations," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 3, pp. 375–380, 1989.

[60] A. Teran and W. Sanchez, "On-board parameter learning using a model reference adaptive position and attitude controller," in *Guidance Navigation and Control, 2017 Proceedings of the IEEE Aerospace Conference.* IEEE, 2017.

[61] R. Bitmead, "Persistence of excitation conditions and the convergence of adaptive schemes," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 183–191, 1984.

[62] V. Adetola and M. Guay, "Parameter convergence in adaptive extremum-seeking control," *Automatica*, vol. 43, no. 1, pp. 105–110, 2007.

[63] J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space mission engineering: the new SMAD.* Microcosm Press, 2011.

[64] R. Tedrake, "Lqr-trees: Feedback motion planning on sparse randomized trees," 2009.

[65] J. M. Phillips, L. E. Kavraki, and N. Bedrossian, "Probabilistic optimization applied to spacecraft rendezvous and docking," *Advances in the Astronautical Sciences*, vol. 114, pp. 261–274, 2003.

[66] B. D. Anderson, "Adaptive systems, lack of persistency of excitation and bursting phenomena," *Automatica*, vol. 21, no. 3, pp. 247–258, 1985.

[67] G. Marafioti, S. Tebbani, D. Beauvois, G. Becerra, A. Isambert, and M. Hovd, "Unscented kalman filter state and parameter estimation in a photobioreactor for microalgae production," *IFAC Proceedings Volumes*, vol. 42, no. 11, pp. 804–809, 2009.

[68] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.* IEEE, 2006, pp. 3562–3568.

[69] J.-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[70] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and adaptive control design.* Wiley, 1995.

[71] M. E. Salgado, G. C. Goodwin, and R. H. Middleton, "Modified least squares algorithm incorporating exponential resetting and forgetting," *International Journal of Control*, vol. 47, no. 2, pp. 477–491, 1988.

[72] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, 2013.

[73] M. Hou, G. Duan, and M. Guo, "New versions of barbalats lemma with applications," *Journal of Control Theory and Applications*, vol. 8, no. 4, pp. 545–547, 2010.

[74] E. Lavretsky, "Invariance-like theorems," May 2005.