Toward Co-Design of Autonomous Aerospace Cyber-Physical Systems

by

Justin M. Bradley

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2014

Doctoral Committee:

Associate Professor Ella M. Atkins, Chair
Professor Dennis S. Bernstein
Assistant Professor James W. Cutler
Professor Ilya V. Kolmanovsky
Professor Kang G. Shin

For Hal was the nervous system of the ship; without his supervision, Discovery would be a mechanical corpse.
- Arthur C. Clarke “2001: A Space Odyssey”

2014

Dedication

This work is dedicated to the many people the world over who find they don't really fit into the boxes their community has constructed for them.

ACKNOWLEDGMENTS

No work of this magnitude is accomplished without the support of many people. In my case this is particularly true thanks to the exceedingly difficult circumstances which the last four years has brought me. The University of Michigan, the Aerospace Dept., and its faculty and staff have been gracious, kind, helpful, accommodating, and fair. The members of my committee have provided useful feedback along the way that has helped shaped this work.

The support I've received from individuals seems to never end. Ali Nasir, Cat McGhan, Allen Hicken, Ryan Eubank, and others have spent hours talking with me and helping in many ways. Derrick Yeo filled a gap in my life that few have filled before. Thank you Derrick, I truly could not have done this PhD without you. C.J. Sorenson has been a source of great strength and encouragement since the 1^{st} grade and his support during this PhD was endless.

Perhaps the constant support through all of my time here has been my adviser, Ella Atkins. Ella has been more than just an adviser, she has been a friend, a resource, and an encouragment.

My mom, Sally, my dad, Phil, have both been my cheerleaders as well as sources of emotional, and physical strength. My brother, Josh, has become a friend and support in ways I never expected. Carrie entered my life in an unusual way and we rapidly became friends which blossomed into romance. Carrie continues to be a bedrock of love and encouragment to me.

Finally, I'd like to thank my children, Emily, Madison, and Caleb. You are my pride and joy and I love you dearly and fiercely.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

A/D Analog-to-Digital

AI Artificial Intelligence

ASPEN Automated Scheduling and Planning ENvironment

CASPER Continuous Activity Scheduling, Planning, Execution, and Replanning

CPS Cyber-Physical System

D/A Digital-to-Analog

DARE Discrete-time Algebraic Riccati Equation

DDE Delay Differential Equation

DLQR Discrete Linear Quadratic Regulator

EDF Earliest Deadline First

FDE Functional Differential Equation

FPRB Forward-Propagation Riccati-based

GSDLQR Gain-Scheduled Discrete Linear Quadratic Regulator

GPS Global Positioning System

HBSS Heuristic-Biased Stochastic Sampling

HSTS Heuristic Scheduling Testbed System

I/O Input/Output

LMI Linear Matrix Inequality

LQR Linear Quadratic Regulator

LEO Low Earth Orbit

LVLH Local Vertical Local Horizontal

MDP Markov Decision Process

MAV Micro-Air Vehicle

NCS Networked Control System

ODE Ordinary Differential Equation

OLHP Open Left Half Plane

ORHP Open Right Half Plane

QoS Quality of Service

RMS Rate-Monotonic Scheduling

RTOS Real-Time Operating System

RTS Real-Time System

UAS Unmanned Aircraft System

WCET Worst-Case Execution Time

ZOH Zero-Order Hold

NOMENCLATURE

Subscripts

c	Cyber
p	Physical
CPS	Cyber-Physical System
0	Initial condition
r	Reference values

Chapter 2 Symbols (in order of appearance)

τ_1	Control task
$\mathbf{x}_p$	State vector for physical system
$\mathbf{u}_p$	Control input for physical system
T_{τ_1}	Period of task τ_1
r_{τ_1}	Task execution rate in Hz of task τ_1
τ_2	Real-time system task achieving important mission goal
τ_3	Additional real-time system task
t	Time
$\mathbf{A}_p$, $\mathbf{B}_p$	System matrices for physical system
$\mathbf{u}_{p,\text{ZOH}}$	ZOH control input at time t which is held for time period T_{τ_1}
δ	System delay
δ^*	System delay resulting in instability
ω_s	System sampling rate
ω_b	System bandwidth
H	Hybrid system
Q	Discrete state set
Σ	Dynamical subsystem
Inv	Invariants that must be true to remain in a particular state
J	State transition behaviors
$Init$	Initial conditions

Chapter 3 Symbols (in order of appearance)

Σ	Dynamical subsystem
$\mathbf{x}_p$	State vector for physical system
$\mathbf{u}_p$	Control input for physical system
k	Spring stiffness
c	Damping parameter
λ	System eigenvalues
k_p	Gains for physical system
T_s	Sampling time
T_{si}	Time required to calculate control input
T_{sc}	Time for A/D and D/A conversions
ω_s	System sampling rate
ω_b	System bandwidth
y_c	Fraction of all control loop activities completed
f	Frequency at which the control loop is executed
$n(t)$	Cycle or control loop count
u_c	Control input for cyber system
$x_{c,1}$	Fraction of control loop calculations completed in this cycle
$x_{c,2}$	Frequency at which the control loop executes
$\mathbf{u}$	Combined physical and cyber system control inputs
$x_{c2,min}$	Minimum allowed cyber rate
J	LQR Cost function
$\mathbf{Q}, \mathbf{R}$	LQR weighting matrices
$\mathbf{e}$	Integrator states
$\mathbf{C}$	Output matrix
g	Acceleration of gravity
l	Length of pendulum
$\mu(f_s)$	Average physical system error over a specified time interval
e_p	Physical state error
f_{sw}	Switching frequency
$c_{x_{p1},x_{c2}}$	Dependence of x_{p1} on x_{c2}
$c_{x_{p2},x_{c2}}$	Dependence of x_{p2} on x_{c2}

Chapter 4 Symbols (in order of appearance)

$\mathbf{J}$	Inertia matrix
θ_1, θ_2, θ_3	CubeSat roll, pitch, yaw
ω_1, ω_2, ω_3	Elements of CubeSat angular velocity

ω_o	Orbital angular velocity
μ	Gravitational constant
R	Radius of orbit
$H_1^w,\ H_2^w,\ H_3^w$	Angular momentums of reaction microwheels
$\bar{H}_2^w$	Bias in pitch reaction microwheel
$M_1,\ M_2,\ M_3$	Torques applied by reaction microwheels
$\mathbf{A}_p,\ \mathbf{B}_p$	System matrices for physical system
$\mathbf{x}_p$	State vector for physical system
$\mathbf{u}_p$	Control input for physical system
τ_1	Control task
$\mathbf{\Phi}_p,\ \mathbf{\Gamma}_p$	Discrete system matrices for physical system
$\mathbf{K}_p$	Gain matrix for physical system
$\mathbf{Q}, \mathbf{R}$	LQR weighting matrices
$\mathbf{P}$	Positive definite solution to DARE
T_{τ_1}	Period of task τ_1
r_{τ_1}	Task execution rate in Hz of task τ_1
$r_{\tau_1,\min},\ r_{\tau_1,\max}$	Minimum and maximum sampling rate
x_c	State vector for cyber system
u_c	Control input for cyber system
$\mathbf{K}_{cp}$	Gain matrix coupling cyber to physical system
k_c	Cyber gain
$\mathbf{m}_p$	Physical state metric
t_f	Final time
$\mathbf{m}_{p,n}$	Normalized physical state metric
m_c	Cyber rate metric
$x_{c,\min},\ x_{c,\max}$	Minimum and maximum cyber state (rate)
$\mathbf{m}_{up}$	Control effort metric
U_RTS	Utilization of RTS
P_i	Mechanical power of i^{th} reaction microwheel
P_total	Total mechanical power for all reaction microwheels

Chapter 5 Symbols (in order of appearance)

E	Energy for the physical system
$P(v)$	Aircraft power as a function of velocity
v	Aircraft velocity (airspeed)
T	Total time required to complete mission

K	Aerodynamic parameter
S	Surface area of wing
W	Weight
ρ	Air density
C_{D_0}	Zero-lift drag coefficient
$P^s_{\max}$	Maximum power of the engine/motor at sea level
η	Propeller efficiency factor
ρ^s	Air density at sea level
δ_t	Throttle setting
J_p	Cost function for physical system
β_{p1}, β_{p2}	Weighting factors for physical system
τ_2	Real-time system task to capture and process imagery
r_{τ_2}	Task execution rate in Hz of task τ_2
$r_{\tau_2,\min}$, $r_{\tau_2,\max}$	Minimum and maximum sampling rate
k	Execution cycle of a task
U_{τ_2}	Cyber utilization cost term
H	Entropy cost
α	Tuning parameter in entropy cost
$\Omega(t,k)$	Overlap between successive image footprints
A	Area of an image
w	Width of an image
T_{τ_2}	Period of task τ_2
J_c	Cost function for cyber system
β_{c1}, β_{c2}	Weighting factors for cyber system
J	Combined CPS cost function
D	Total distance of mission
L	Lagrangian of constrained cost function
$\lambda_{1...4}$	Lagrange multipliers
$v_{\min}$, $v_{\max}$	Minimum and maximum velocities
e_0	Oswald efficiency factor
b	Wingspan
m	Mass of aircraft
AR	Aspect ratio
f	Focal length of camera
H_{dist}	Horizontal distance of imaging plane
V_{dist}	Vertical distance of imaging plane

ABSTRACT

Toward Co-Design of Autonomous Aerospace Cyber-Physical Systems

by

Justin M. Bradley

Chair: Ella M. Atkins

Modern vehicles are equipped with a complex suite of computing (cyber) and electromechanical (physical) systems. Holistic design, modeling, and optimization of such Cyber-Physical Systems (CPS) requires new techniques capable of integrated analysis across the full CPS. This dissertations introduces two methods for balancing cyber and physical resources in a step toward holistic co-design of CPS. First, an ordinary differential equation model abstraction of controller sampling rate is developed and added to the equations of motion of a physical system to form a holistic discrete-time-varying linear system representing the CPS controller. Using feedback control, this cyber effector, sampling rate, is then co-regulated alongside physical effectors in response to physical system tracking error. This technique is applied to a spring-mass-damper, inverted pendulum, and finally to attitude control of a small satellite (CubeSat). Additionally, two new controllers for discrete-time-varying systems are introduced; a gain-scheduled discrete-time linear regulator (DLQR) in which DLQR gains are scheduled over time-varying sampling rates, and a forward-propagation Riccati-based (FPRB) controller. The FPRB CPS controller shows promise in balancing cyber and physical resources.

Second, we propose a cost function of cyber and physical parameters to optimize an Unmanned Aircraft System (UAS) trajectory for a pipeline surveillance mission. Optimization parameters are UAV velocity and mission-critical surveillance task execution rate. Metrics for pipeline image information, energy, cyber utilization, and time comprise the cost function and Pareto fronts are analyzed to gain insight into cyber and physical tradeoffs for mission success. Finally, the cost function is optimized using numerical methods, and results from several cost weightings and Pareto front analyses are tabulated. We show that increased mission success can be achieved by considering both cyber and physical parameters together.

CHAPTER 1

Introduction

Cyber-Physical Systems (CPSs) are "the next generation of systems that require tight integration of computing, communication, and control technologies to achieve stability, performance, reliability, robustness, and efficiency in dealing with physical systems of many application domains" [1]. CPS research aims to more synergistically integrate control, computing, communications, and physical systems in novel ways that leverage interdependent behavior.

The potential impact of CPS research is far-reaching owing to the diverse platforms and applications to which it can be applied. CPS have been the topic of numerous reports of global interest [2–7]. Consumer devices such as smartphones, TV, and gaming systems respond to voice commands and wearable electronics are becoming the norm. Imbuing buildings with advanced sensors, networks, and intelligent energy management can reap great rewards in efficiency [8]. Advances in health care medical devices and systems can lower operating costs and increase reliability and patient care. New software-enabled functionality, increased connectivity, and physiologically closed-loop systems have the potential to reduce human error that can cost lives [9]. A new energy service system dubbed the "smart grid" promises to utilize CPS technologies to increase configurability, adaptability, reactiveness, and self-manageability [10], but will simultaneously require CPS breakthroughs in security to monitor, manage, and thwart threats both to the physical entities comprising the grid, as well as the cyber attacks on its networked components [11]. Most relevant to the work in this dissertation is the application of CPS research to vehicle and robotic systems. In this domain CPS research offers an increase in autonomy, reconfigurability, reliability, safety, energy efficiency, and robustness [12].

Human beings are the quintessential CPS possessing heavily interdependent cyber (mind) and physical (body) subsystems. Analogous to this mind-body paradigm, advanced robotic systems utilize both cyber and physical resources. However, unlike the symbiotic mind-body awareness humans have, to date cyber and physical subsystems are unaware or only

partially aware of the other. Most likely the cyber system receives performance feedback and calculates control inputs for the physical system. In this way the cyber system serves the needs of the physical system. This particular role has received a great amount of research attention most prominently in the form of control theory and Real-Time System (RTS) theory. It is also likely that the physical system is accomplishing a mission objective that services the goals of the cyber system, for example, surveillance, safe transportation, science data collection, etc. In this way the physical system serves the needs of the cyber system. This role, however, has historically been dominated by humans in-the-loop who design plans, set waypoints, or modify tasks either through an interface or direct software manipulation.

At the lower, reactive control level this partial awareness is manifested by the subservience of the cyber system to the needs of the physical system. For example, hard-deadline tasks in a real-time system are time-triggered (e.g. control tasks) and are therefore executed regardless of environmental or robotic system conditions even if the physical system is in steady state and needs only minimal control authority. While these abstractions have arguably led to a wealth of theory in control and real-time systems theory, allocation of cyber resources can be inefficient and may detract from servicing other tasks [13].

CPS research calls for new models, new abstractions, new performance metrics, new design methodologies, integration methods for large-scale systems, new methods of reasoning about uncertainty, and a revolution in how we think about computing [12, 14]. While the depth offered by separately modeling and analyzing physical and cyber subsystem behaviors is useful, aberrant system behavior (i.e. when laws of compositionality or composability do not hold) may be undesirable at best, and dangerous at worst. Accounting for as many subsystem interactions as possible can reduce the negative side effects of such behaviors as well as providing provable holistic system characteristics (e.g. stability) [15]. Integrated analyses can enable more efficient, safe, secure, and capable systems as we increase the level of autonomy in CPS devices and vehicles.

In this dissertation we describe our research to address some of the challenges facing CPS. We hope that through the proposed new abstractions, metrics, cost functions, and methods for balancing cyber and physical resources presented here that we spark the creativity of future researchers and inspire new technologies that will power the CPS of the future.

1.1 Research Objectives

We are pursuing the development of new models, frameworks, algorithms, and design techniques that allow for the co-design of holistic CPS through two main research thrusts:

- **CPS Co-Regulation Problem Statement**: develop new abstractions and models of control tasks for the cyber system such that an efficient feedback regulation process using a new CPS linear system representation can be realized. Develop new discrete-time-varying controllers capable of addressing the time-varying nature of the variable sampling rate and co-regulate both cyber and physical effectors through feedback control.
- **CPS Co-Optimization Problem Statement**: develop new metrics coupling cyber and physical resource use into a co-optimization framework for mission planning. Gain insight into the tradeoffs between physical and cyber parameters via Pareto front analysis and demonstrate how increased mission efficiency can be obtained by dual consideration of cyber and physical resources.

1.2 CPS Co-Regulation

CPS interface physics-based and digital computing world models. These systems may link digital devices with a human, vehicle or robotic device, or both. As embedded digital systems become more prevalent and cost effective, co-design of cyber and physical system properties shows promise of making the holistic system more capable and efficient. To-date, co-design has involved deciding how to allocate system resources to manage both its cyber and physical effectors. In the context of real-time scheduling, resources such as processors and communication channels are allocated across tasks/messages aimed at managing sensors, actuations, and information. In the context of the feedback control system, objectives of the physical system (e.g. disturbance rejection, tracking accuracy, etc.) are translated to computing actuator commands that minimize errors between reference and actual trajectories through physical space. Traditional feedback controllers model state as a vector of continuous-valued positions and velocities, relying on physics-based models to predict trends over time. "Cyber" controllers such as real-time schedulers assign a discrete set of tasks across discrete resource intervals to meet task goals, processing and communication link load constraints, etc. CPS co-design challenges us to identify and capture synergistic properties of cyber and physical systems. Figure 1.1 illustrates the traditional decoupled cyber and physical modeling and control processes but also proposes links to en-

able the "cyber" controllers to consider physical system properties and the "physics-based" controllers to consider cyber (computing resource) properties.

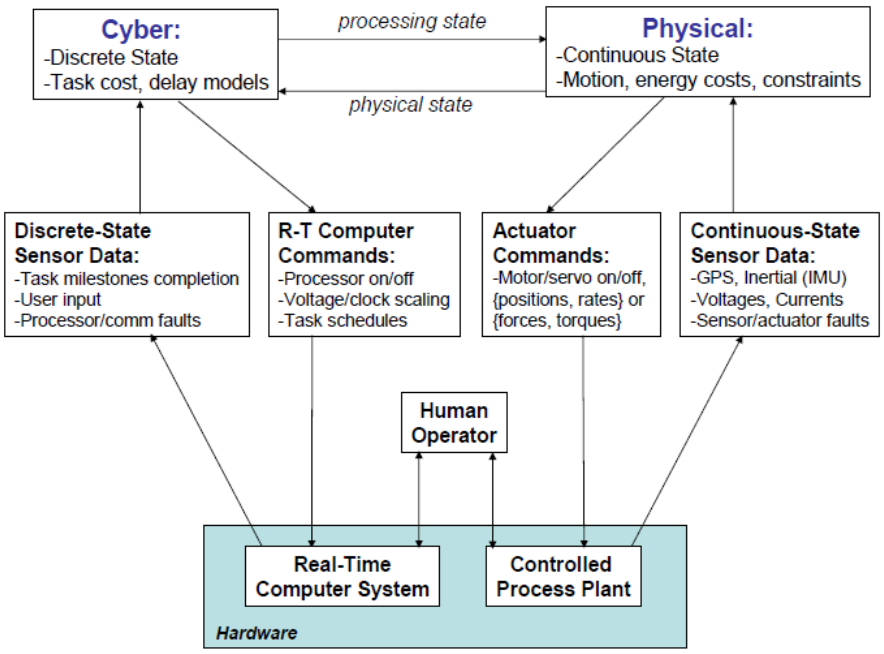


Figure 1.1: Abstract Duality of Cyber and Physical Systems

Traditional approaches to system design are specific to one of these effector classes, either physical or cyber. Both computer scientists and control engineers have developed mechanisms for dealing with the alternate cyber/physical component but rarely have those methods combined to allow for true co-design of the CPS as a whole. Poovendran points to this issue as one of the grand challenges facing the CPS community, stating "A dramatic increase in the ability to perform true cyber-physical co-design–where the physics of surface friction, moments of inertia, and computer hardware and software behavior can be simultaneously observed–is critical to advance both how CPS is engineered and how it is deployed" [16].

CPS typically require an interacting suite of communication and processing tasks. This requirement can become a limiting factor forcing RTS engineers to design inflexible schedules. RTS designers traditionally aim to provide hard timing guarantees particularly for safety-critical physical system controllers, with best-effort execution of non-critical (soft real-time) tasks. For sampled-data control systems this is done using periodic or time-triggered sampling of the system also known as Riemann sampling [17]. The effects of limited processing capacity are rarely taken into account during the design of the physical system controller, so static hard timing guarantees are expected. Without taking comput-

ing system limitations into account, the controller may ask for more resources than are needed to achieve performance objectives. As a result Riemann sampling may waste cyber resources during quiescent periods of physical system activity, in addition to providing sub-optimal system performance [17,18]. Event-triggered or Lebesgue sampling holds promise for better resource utilization and control performance at the expense of scheduling complexity for the RTS [17]. Perhaps more importantly, although there has been some recent work exploring event-based feedback control [18–22], as well as a hybrid control approach that switches between Riemann and Lebesgue sampling [23], Lebesgue sampling is still a largely unexplored area relative to Riemann sampling [17].

Following the CPS abstract duality in Figure 1.1 we propose a representation and corresponding theory to unify these disparate notions of "effector regulation" into a common framework. Figure 1.2 represents our proposed framework wherein a traditional physical system control scheme and a cyber feedback scheduling (see Section 2.3.4.2) scheme are coupled through feedback control. The cyber model (feedback scheduling scheme) is rep-

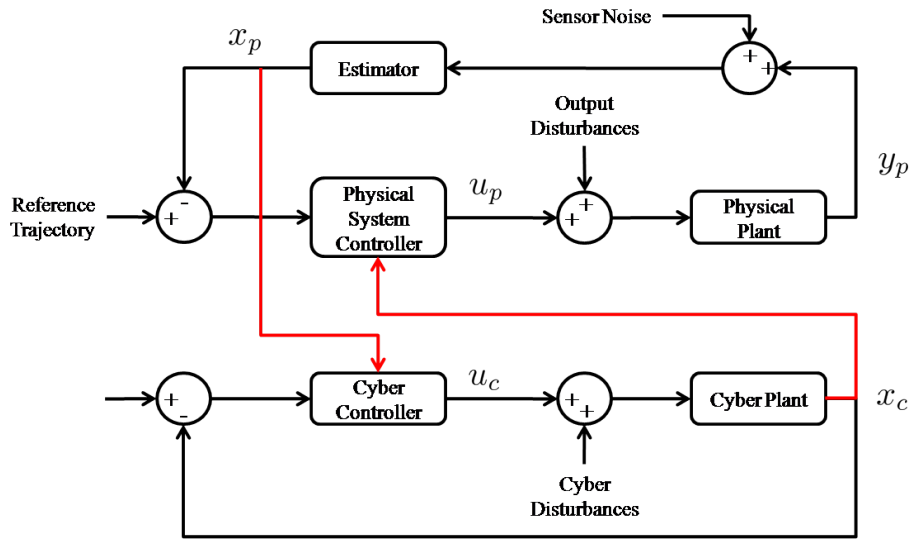


Figure 1.2: CPS Co-Regulation Block Diagram

resented by a linear model abstraction of controller task sampling rate and controllers for both the cyber and physical system are designed incorporating holistic CPS state information. Disturbances to the physical system (e.g. sensor noise, model uncertainty, etc.) and the cyber system (e.g. thermal issues, deadline uncertainty, etc.) are accounted for by simultaneously regulating both physical and cyber states.

We hope this incremental advancement toward physical-computational state co-regulation helps the CPS community meet the grand challenge of co-design. We aim to provide the benefits of Riemann sampling: ease of RTS scheduling, hard timing guarantees, and the rich

theory of digital control while also providing some of the benefits of Lebesgue sampling: on-demand cyber resource utilization. Our abstraction allows an engineer to treat scheduling of a control task as a control problem where interactions between cyber and physical states are represented in a common regulation framework. We also present two new controllers, a gain-scheduled discrete linear quadratic regulator, and a foward-propagation Riccatic-based controller for discrete-time-varying systems and through simulation demonstrate their effectiveness. This technique is applied to a spring-mass damper, inverted pendulum, and finally to attitude control of a small satellite (CubeSat). We demonstrate the effectiveness of our approach via a case study for the CubeSat representing disturbance rejection. CPS performance is analyzed using a new set of CPS metrics designed to measure both cyber and physical performance objectives.

1.3 CPS Co-Optimization

The primary goal of air vehicle design has historically been to achieve appropriate maneuverability and to overcome aerodynamic drag and the influence of gravity in a manner that maximizes range and/or endurance. To-date, the energy a powered aircraft requires to apply the necessary propulsive and control actuation forces over a flight has dominated the total energy consumed across all other vehicle subsystems, including cyber. Surveillance Unmanned Aircraft Systems (UAS) are becoming smaller and are constructed with composite materials that minimize weight [24]. They are also being equipped with increasingly sophisticated avionics and payloads. Powered glider designs in particular exhibit low drag and weight, resulting in a significantly reduced thrust requirement. For the first time, the power required by avionics and payload systems for a flight vehicle is comparable to propulsive plus control actuation requirements. We will likely see a future where avionics and payload power can even exceed power required for force application particularly during periods of demanding on-board processing and communication activity.

Control systems engineers typically optimize vehicle trajectories and thereby force application time histories over physics-based models of vehicle dynamics, including flight envelope and actuator saturation constraints. Conversely, software engineers optimize processor and communication resource use over the suite of computational and information sharing tasks, regulating energy use through the real-time regulation of variable-speed processors, activation/shutdown of cores in a multi-processor system, and regulation of communication links. While real-time task execution models are typically discrete rather than continuous-time, the methods used to optimally control physical and real-time computing systems are fundamentally the same: gradient or search-based algorithms are used to

identify minimum-cost solutions given constraints.

For emerging UAS that consume comparable power for avionics versus force application, or whose mission effectiveness is dependent on both cyber and physical resources, neither physical nor cyber system optimization is dominant. Therefore, globally-optimal (minimum-energy, minimum-time, maximum-information) performance can only be achieved if cyber and physical models can be shown independent of each other, or else if necessary cyber and physical couplings are identified and simultaneously considered during optimization. Computational resources must be utilized as a minimum to guide, navigate, and control the UAS as well as to compute or update future spatiotemporal (4-D) trajectories. Physical trajectories in turn enable the cyber system to maximize its ability to acquire information (e.g. from payload sensors) and to communicate (e.g. with ground operators). Cyber and physical resources are therefore necessarily coupled.

We present a new multi-disciplinary optimization [25] direction for which the models being integrated optimize energy consumption and mission success over both physical effectors and cyber resources. The cost function to be optimized includes weighted terms representing energy used by physical actuators, cyber resource utilization, time, and mission-critical surveillance information. We present a case study of a UAS surveillance mission to assess the potential performance improvements possible with co-optimization of cyber and physical resources.

1.4 Contributions

The contributions of this work are primarily directed toward the emerging CPS community though it makes use of and relates to the control, Artificial Intelligence (AI), Real-Time System (RTS), and Aerospace communities as well. This dissertation makes the following contributions:

- New abstractions and models for representing the cyber system in a coupled CPS. We develop continuous state variable models for the cyber system to enable cyber system integration and analysis in a linear systems framework. This formulation enables use of the rich tools from linear system theory to the holistic CPS.

- Metrics for evaluating CPS and mission performance. The development of metrics for measuring CPS performance has been one of the grand challenges in the CPS community [2]. We leverage work from the real-time systems community to develop metrics for measuring cyber performance. We also propose and demonstrate the use of specific metrics that combine both physical and cyber parameters.

- Development of two new controllers for discrete-time-varying systems.
  - Gain-Scheduled Discrete Linear Quadratic Regulator (GSDLQR) wherein DLQR gains are scheduled over time-varying sampling rates of a system. To our knowledge this is the first time feedback control gains have been scheduled over time-varying sampling rates of the controller.
  - Forward-Propagation Riccati-based (FPRB) controller. For the FPRB controller we add to the empirical evidence that forward-propagation Riccati techniques are broadly applicable if not yet fully understood.
- Potential for increased energy and mission efficiency in the operation of emerging small Aerospace vehicles. Our application domains include a fixed-wing small Unmanned Aircraft System (UAS), and a small satellite (CubeSat). The key to efficiency gains is to appropriately balance the proposed new metrics for CPS performance.
  - First application of time-varying sampling rates to a small satellite platform (CubeSat).
- Extensible software capable of simulating a system with dynamically changing sampling rates. We have built extensible software that can simulate a CPS that fits into our linear systems abstraction. Unlike traditional discrete system simulators, our implementation accurately simulates continuous-time physical system response while also simulating the time-varying sample-and-hold behavior of a digital control system.

1.5 Innovations

Innovations from our work take the form of novel methods for coupling, analyzing, optimizing, and regulating CPS. The following specific innovations are offered:

- Development of an abstract cyber model corresponding with traditional "physics-based" representations and the inclusion of this model into a linear systems framework. This innovation provides a low-level feedback co-regulation of both physical and cyber effectors.
- Design of feedback controllers for proposed cyber model enabling co-regulation of holistic CPS.

- Development of new metrics for the cyber system, as well as for a holistic CPS.
- Co-optimizing CPS trajectory over a suite of physical and cyber metrics and analyzing coupled system behavior with a Pareto front analysis that couples cyber and physical cost terms.

1.6 Outline

The dissertation is organized as follows. First, in Chapter 2 we review research related to this work both theoretically and within the aerospace community. In Chapter 3 (also [26]) we propose a new model for "abstracting" discrete task execution models into a continuous-state framework amenable to linear systems analysis and co-regulation. This cyber model is added to spring-mass-damper and inverted pendulum systems and plots are shown. In Chapter 4 (also [27]) we further refine the cyber model and apply this technique to attitude control of a small satellite (CubeSat). We discuss two new controllers for our discrete-time-varying system and simulate a disturbance response to the attitude of the CubeSat. In Chapter 5 (also [28]) we propose a co-optimization scheme and examine tradeoffs between cyber and physical resources for mission success. We then present conclusions and related future work in Chapter 6.

CHAPTER 2

Background and Related Work

CPS research encompasses a broad scope and diverse applications. Ultimately, a primary goal of developing foundations across cyber and physical system models is to appropriately balance the performance of the physical system with the performance of the cyber system. In this chapter we focus attention on computationally-based control research and aerospace applications relevant to the research in this dissertation.

We begin by discussing how "Cyber-Physical Systems" came about - the confluence of communications, control, and computing, including a brief history of those topics. We then discuss the implications of computer-controlled systems in terms of delay and sample-and-hold systems. Approaches to addressing these difficulties is then discussed including time-delay systems, network control systems, and recent advances in CPS research relevant to digital real-time control. Finally, although CPS research is still in its infancy in the Aerospace community we discuss how CPS issues have been addressed therein.

2.1 Brief History of CPS

In the last century advances in communication, control, and computing were primarily used as individual tools in their respective domains. For example, advances in communication were used strictly for communicating between humans without integration into more complex systems. However, as technology has advanced the integration of the tools and techniques within each domain into more complex systems has provided a new frontier fusing communication, control, and computing.

An excellent exposition on CPS research and its history can be found in [1] and we recapitulate the history here focusing on integration of these technologies. In Figure 2.1 is a timeline of important events and developments in communication, control, and computing that led to CPS as an important topic of research.

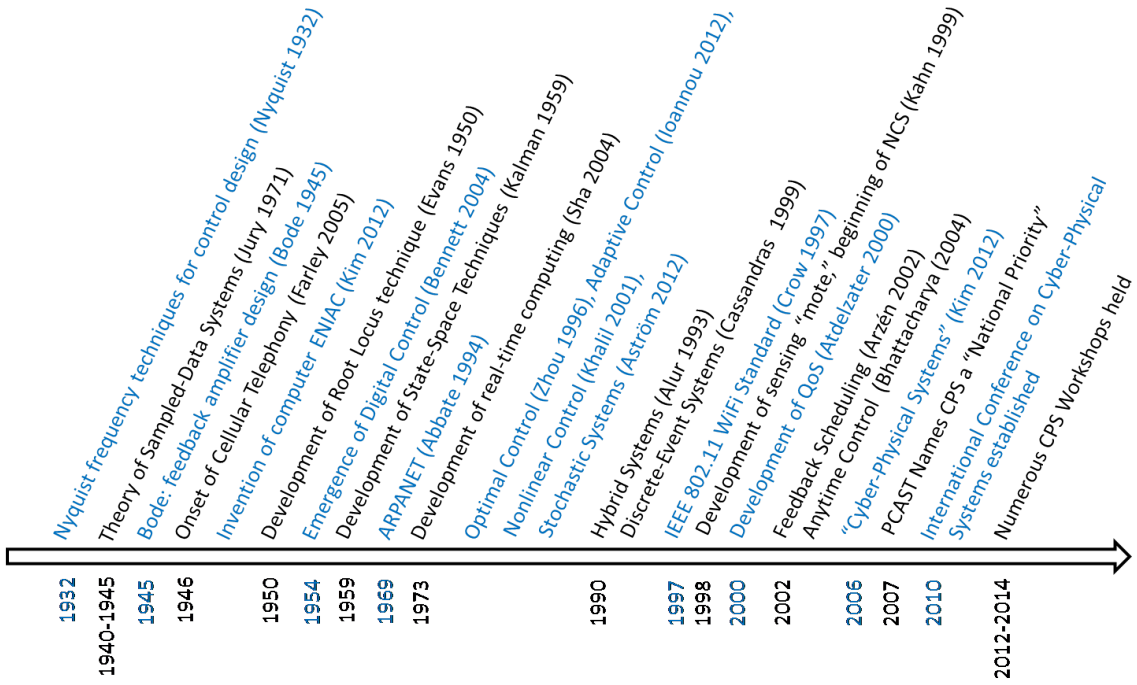


Figure 2.1: CPS History

2.1.1 "Cyber"

The term "cyber," as a prefix, stems from the field of research known as "cybernetics" - the scientific study of control and communication in the animal and the machine [29]. Cybernetics as a field of research in the modern era began in the 1940's with Norbert Wiener, Warren McCulloch, W. Ross Ashby, Alan Turing, and W. Grey Walter. Since then, however, semantically, "cyber" is usually associated with Information Technology, Computers, the Internet, or to denote control in the computer or electronic context [30]. Perhaps the most fitting definition of the word "cyber" from "cybernetics" stems from Plato's *The Alcibiades* and is "the study of self-governance" [31]. As a field of research CPS strives to improve self-governance for machines, infrastructure, and devices.

2.1.2 Communication

In 1969 the ARPANET was developed as an experiment linking computers together at major university campuses across the U.S. [32]. The development of this technology into what we now know as the Internet is perhaps one of the greatest communications innovations of humankind. We are only beginning to realize the power of this technology as the Internet of Things [33] enables communication between autonomous agents in a worldwide cyber-

physical network. Aiding this network of interconnected agents is the development of cellular communication which began modestly in the late 1940's [34]. Continuing the trend of mobile communications, in 1997 the IEEE WiFi standard (802.11b) was established and spread quickly across homes and businesses [35].

Advances in communication have enabled a revolution merging computing and communication in systems ranging from large interconnected networks to small autonomous agents leveraging local and potentially network-level (cloud) information. The end result is a high-bandwidth flow of information that will only continue to grow.

2.1.3 Control

Although control theory has roots in mechanical devices such as the steam engine governor in the early 1800's [36], very early in the 1900's a strong theory, often dubbed "classical control," based in the frequency domain was developed by Nyquist, Bode and others primarily at Bell Laboratories [37–40]. By the 1950's the onset of computing in conjunction with the "space race" spawned a new form of control - digital control - whereby computers could now perform calculations on incoming sensor signals to compute control inputs [41]. During this time "modern control" utilizing linear algebra techniques was developed and linear systems theory and state-space methods became the norm [42–45]. From such firm foundations came optimal control [46], nonlinear control [47], stochastic control [48], adaptive control [49], discrete event systems [50], and hybrid systems [51]. Perhaps most pertinent to a discussion of the integration of computing and control is modern digital control arising from the need to control inherently periodic systems. The periodic nature of sampled-data systems arises from the periodic scheduling of the control task on a real-time system (discussed further in Section 2.2). Out of this thread of research the Z-transform, "frequency" domain, state space and other digital control tools some of which were analogous to those in the continuous domain were developed [52, 53].

Control theory and applications have exploded in popularity centered on strong mathematical foundations and guarantees of important system properties (e.g. stability). The tight integration of computing and control is the future of complex systems. Advances in strong mathematical foundations for computing and control are required for efficiency and autonomy.

2.1.4 Computing

Not long after the computer was invented it was adopted for use in closed-loop control of physical systems [1]. Coupling a physical system with Analog-to-Digital (A/D) and

Digital-to-Analog (D/A) converters enabled a computer to calculate control inputs and thereby control the physical system. Prior to this computers had strictly been used to execute fixed sequences of control instructions. The timing of computations was immediately recognized as critical for success [54]. Initially to address this computational algorithms took the form of cyclic executives, *ad hoc* algorithms in which one task loops over several activities adjusting timing as it progresses through the loop [55]. One or several of these activities were designed to make computations and use them to apply control inputs to the physical system at regular timing intervals. This paradigm gave rise to many of the advancements in digital control. It wasn't long, however, until the need for a rigorous theory, tools, and curriculum for scheduling and real-time systems were needed to support more demanding applications. In the early 1980's the U.S. Office of Naval Research started the Real-Time Systems Initiative. Under this initiative many of the modern scheduling algorithms and analysis tools were developed [55]. Even so, due to success in computing and the surprising adherence to Moore's law [56] computational resources have often been assumed infinite.

Computing, control, and communication have combined to form powerful complex systems that are widely applicable. Communication between remote agents and the Internet of Things can provide a wealth of information for autonomous agents to learn quickly and leverage the experience of other robots [57]. Control enables the translation of robotic and/or human objectives into physical realities through robust, stable actions. The common thread and analogous decision-making "mind" is the computing platform governing communication and physical operation through space and time. The holistic, synergistic integration of these three areas constitutes CPS research.

2.2 Real-Time Computing Meets Control

Since computing resources are finite, a simplifying assumption of infinitely-fast sampling rate is not realizable in practice. In a Real-Time Operating System (RTOS) processor time is allocated to tasks according to a schedule. Tasks are divided into hard and soft deadline requirements; missed hard deadlines are unacceptable and may result in system failure while missed soft deadlines degrade quality of service [58]. If we use a RTOS to implement control of a system, the timing of reading sensors, calculation of control input, and output of the control signal is modeled and regulated as timing including delays and update rate have an impact on both the design of the controller and the scheduling algorithm. Whatever method is used to implement a controller task the assumption is made that the RTOS can guarantee the controller task deadlines and therefore guarantee a specific sampling rate.

Perhaps the simplest real-time scheduling algorithm is known as a *cyclic executive*. In a cyclic executive a single executive task loops over several activities, or tasks, adjusting timing as it progresses through the loop. Assume τ_1 is a control task implemented on an RTOS. That is, assume τ_1 receives sensor values from the A/D converter, obtains an updated system state estimate, $\mathbf{x}_p$, computes the control input, $\mathbf{u}_p$, and outputs the control signal to the D/A converter. Also assume that the control input is applied at the completion of the task and is held for T_{τ_1} seconds, which is the period of task τ_1. Note that T_{τ_1} is the control task period or sampling period, and that $1/T_{\tau_1} = r_{\tau_1}$ is the sampling rate, or control task execution rate. Assume that τ_2 and τ_3 are two other real-time tasks to be scheduled. In a cyclic executive the Worst-Case Execution Time (WCET) of each task is used to schedule the tasks as seen in Figure 2.2. If a given task does not require its full WCET to complete its

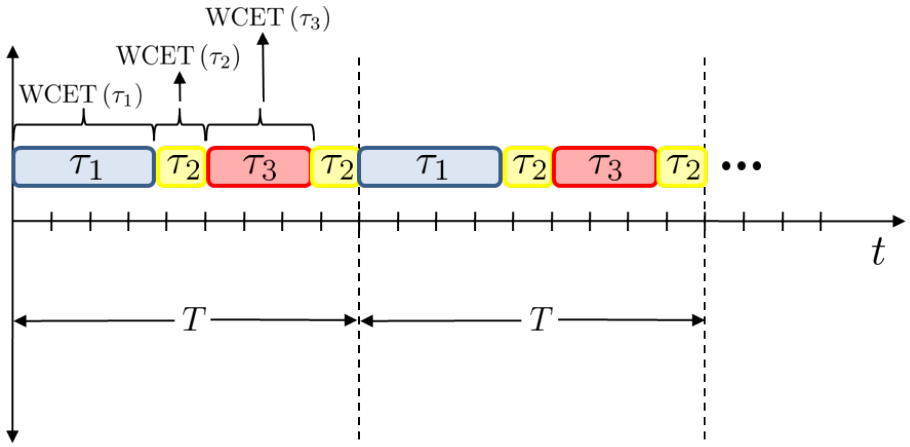


Figure 2.2: Cyclic Executive Scheduling

work the executive inserts idle time or executes other soft real-time (best-effort) tasks as it awaits arrival of the next hard real-time task. In the case of the cyclic executive this means that in the worst case the delay between reading the sensors and outputting the control signal is WCET(τ_1). If we assume a continuous-time linear state-variable model of the physical system, taking this into account would give us

$$\dot{\mathbf{x}}_p\left(t, \text{WCET}(\tau_1)\right) = \mathbf{A}_p \mathbf{x}_p\left(t, \text{WCET}(\tau_1)\right) + \mathbf{B}_p \mathbf{u}_{p,\text{ZOH}}\left(t, \text{WCET}(\tau_1)\right)$$

where $\mathbf{u}_{p,\text{ZOH}}(t, \text{WCET}(\tau_1))$ represents the zero-order held (ZOH) control input at time t which is held for time period T_{τ_1}.

Most modern real-time scheduling algorithms are preemptive, allowing tasks to interrupt each other to be serviced as long as deadlines can be met. Static or offline scheduling protocols such as Rate-Monotonic Scheduling (RMS) offer solutions that can be analyzed

prior to deployment providing a guarantee, under nominal operating conditions, that all hard deadlines will be met [59]. In contrast, dynamic, or online schedules such as Earliest Deadline First (EDF), are executed at run-time requiring the scheduling algorithm to search through a queue of jobs (instances of a task) to find the next job to be scheduled [59]. Most schedules are created to guarantee that all hard deadlines will be met and leave slack for soft real-time tasks to be inserted as they can fit (though soft deadlines may be missed without introducing potential for catastrophic failure).

In a preemptive RTOS containing multiple high-priority tasks, timing is unpredictable. We do not know precisely when the control task will be executed, or whether it will be preempted by a higher priority task. We only know that it will complete by its deadline. We demonstrate this in Figure 2.3. In this schedule each task has a periodic rate at which

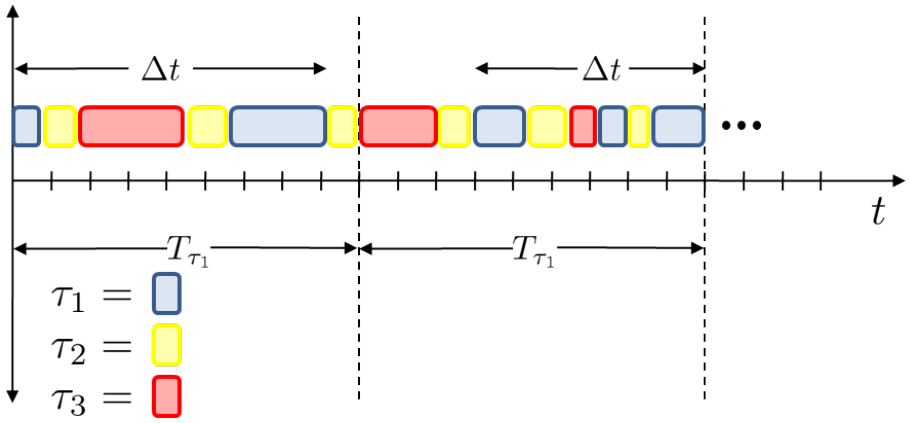


Figure 2.3: Preemptive Scheduling on a Single Processor

it must be executed, but because the tasks are preemptable higher-priority tasks may be serviced first. Schedule feasibility is determined based on the WCET of a task and total system utilization. In a preemptive scheduling paradigm the delays for the physical system being controlled are

$$\dot{\mathbf{x}}_p(t,\Delta t) = \mathbf{A}_p\mathbf{x}_p(t,\Delta t) + \mathbf{B}_p\mathbf{u}_{p,\mathrm{ZOH}}(t,\Delta t)$$

where $\Delta t \in [\mathrm{WCET}(\tau_1), T_{\tau_1}]$ and $\mathbf{u}_{p,\mathrm{ZOH}}(t,\Delta t)$ represents the Zero-Order Hold (ZOH) control input at time t which is held for task period T_{τ_1}. In the preemptive RTOS delay is dictated by context switches between tasks, the task period T_{τ_1}, any tasks that preempted τ_1, and the computation time required to complete τ_1.

Traditional digital control leverages the sampled-data system assumption that the reading of sensors, calculation of control input, and output of the control signal happens instantaneously and always with a current estimate of the physical system state. That control

input is then "held" for the entire sampling time until the next cycle. In other words, it is assumed there is no delay in the system. The problem of control under the varying delays associated with digital real-time control have been studied extensively in the Digital Control, Time-Delay Systems, Networked Control Systems, Automotive, Aerospace, and Real-Time Systems communities [60–68].

2.3 Computer-Based Control Research

The methodologies for merging discrete and physical subsystems including inherent delay and uncertainty has been addressed in several communities. Below, we first discuss approaches for modeling response delays in physical systems, where delay is one important artifact of finite control loop execution rate. This discussion centers around a review of time-delay and digital control systems. Digital control models characterize sampled system behavior but assume a predetermined sampling rate without accounting for the possibility that this rate could itself be regulated. Next real-time systems research is discussed with focus on scheduling theory and algorithms. We then discuss recent advances in CPS research related to the balancing of cyber and physical resources.

2.3.1 Time-Delay Systems

Time-delay systems research has played a prominent role in the definition, control, and stability of systems with delay. The primary difficulty in the development of appropriate tools for modeling these systems is a result of their infinite-dimensional nature [66]. Hence, traditional dynamics (using Ordinary Differential Equations) and by extension traditional continuous control are inadequate. Delay Differential Equations (DDEs), a special type of Functional Differential Equations (FDEs), with accompanying analysis, however, have provided a rich framework for investigation of such infinite dimensional systems [69–71].

The primary result of delay in a physical system is destabilization. Therefore, research into if, and when, a system becomes unstable has played a key role in this field. While some physical systems are, in fact, $\mathcal{S}_\infty$ stable (*delay-independent asymptotically stable*), most physical systems of interest are $\mathcal{S}_\delta$ stable (*delay-dependent asymptotically stable*). In $\mathcal{S}_\delta$ stability, we are interested in the δ^* (i.e. delay) that results in instability of the system, then we design the system with values of $\delta < \delta^*$ that are stable. Lyapunov stability, more specifically Lyapunov-Krasovskii and Lyapunov-Razumikhin stability, have motivated much of the stability analysis in this field. If we think of a FDE as an evolution in a Euclidean space, the application of Lyapunov's second method becomes more clear. Specifically,

Lyapunov-Krasovskii stability tells us that the derivative of the candidate Lyapunov functional $\dot{V}$ must be negative along all system trajectories. As in traditional nonlinear control theory, Lyapunov's second method is often surprisingly difficult to demonstrate. Lyapunov-Razumikhin stability relaxes the Lyapunov-Krasovskii stability theorem and seeks stability on a subset of trajectories defined by system evolution in the interval $[t-\delta, t]$ [72–74].

There are at least two main obstacles in utilizing time-delay system theory in a CPS to provide a unified framework. First, our purpose in modeling delay as a part of the CPS is to allow us to choose the optimal delay under changing conditions. While time-delay system analysis can help us analyze the range of stable delays up to δ^*, it has relatively few tools for handling time-varying delays and appropriately choosing them amidst control objectives. Second, while the delay is part of the system model, it does not function as one of the control variables. This means we cannot directly utilize the rich theory and practical tools from the control community in our design of an energy-conscious CPS through strictly time-delay analyses over physical position/velocity state.

2.3.2 Digital Control

Time-delay systems analysis considers a continuous delay term. In a CPS delay induces a ZOH effect on the physical system. This effect is better suited to a purely discrete mathematical model than a DDE model [75]. Digital control provides this discrete mathematical framework, as well as familiar control techniques couched in a "digital" representation to design, simulate, and model a system.

Two traditional techniques arise from this area of study. The first is direct digital design. Assuming a fixed sampling rate, direct digital design provides tools to derive a digital model of the system from which design, analysis, and simulation can be achieved. Utilizing the Z-transform, the left half of the S-plane is folded into the unit circle. We can guarantee asymptotic stability if the system's eigenvalues (poles) reside within the unit circle. Z-domain analysis, including root-locus, Nyquist stability criterion, etc. are equally valid in the digital domain. State-space equations using difference equations rather than differential equations can be formed, and compensator design, linear quadratic optimal controllers, Kalman filters, etc. retain their familiar form and use.

The second primary method for digital control design is emulation of the controller or compensator. In this method all design and analysis is done in the continuous domain and the assumption is made that the cyber system sampling rate and control calculations are sufficiently fast to adequately control the system. A transformation using a selected sampling rate is applied to the controller to adapt it to the digital domain [60]. This method

is often used in practice, although this assumption is not valid in cases where cyber resource utilization should be minimized rather than simply constrained to avoid over-utilization. This second method is therefore not relevant for CPS co-design.

An important consideration in digital control is the selection of the correct sampling rate. It is clear from an energy usage standpoint that lower sampling rates require less energy. However, lower control loop rates also contribute to the deterioration of system performance thereby potentially increasing energy use due to increased physical actuation. The theoretical lower bound on sampling rate is the familiar Nyquist rate, or $\omega_s/\omega_b > 2$ where ω_s is the sampling rate, and ω_b is the bandwidth of the system [60]. It is, however, well understood that a real physical system will perform poorly and will be highly sensitive to parameter variations and disturbances at sampling rates approaching the Nyquist limit. Beyond the Nyquist rate the speed of open-loop dynamics (real and imaginary poles), the spectrum of possible commanded reference trajectories, the spectrum of expected disturbances, uncertainty in plant dynamics, jitter in the RTS, aliasing, stability margins, and sensor noise are just some of the factors that should be considered when choosing a sampling rate. These are issues without clear cut answers and control engineers face tradeoffs when designing digital controllers.

There is little theoretical basis for correctly choosing sampling rate, and primarily "rules of thumb" have been the norm. *Ad hoc* approaches and trial and error techniques have been used in many cases [76]. Another practice is choosing a sampling rate based on a certain multiple of the fastest mode of the system [77]. Others suggest the safe choice of $\omega_s/\omega_b > 20$ should suffice for most systems [78, 79]. No matter what sampling rate is chosen, a limiting factor in these methods is that they assume sampling rate is fixed. Similar to time-delay system analysis, it does not appear that traditional digital control tells us how to optimally choose (under changing conditions) a sampling rate, nor does it provide us with tools to treat the sampling rate as a control parameter.

Both the time-delay systems and digital control areas approach the problem of a CPS from the perspective of the physical system, but they do not address the need for the cyber system to regulate itself *in relation to* the physical system.

2.3.3 Real-Time Systems

From the cyber perspective, RTS research focuses on task scheduling to provide guarantees of hard-deadline tasks, and the best effort and execution of soft-deadline tasks as discussed in Section 2.2. Offline static schedulers as well as online dynamic schedulers have been proposed to provide provable timing guarantees for given task sets [58]. Some RTS-centric

CPS research has attempted to redefine task execution and scheduling paradigms to accommodate and provide guarantees for classes of tasks suited for more dynamic CPS, for example, tasks with varying periodicity [22, 80].

2.3.4 Recent CPS Research

A number of results emerging from the growing CPS community have had an impact on our understanding of coupled CPS. Anytime control [81–83], feedback scheduling [84–86], Networked Control System (NCS) [61, 87, 88], hybrid systems [89–92], time-varying sampling [63, 93, 94], sensor scheduling [95–99] and optimal sampling patterns [13, 100] are particularly relevant to our work, covering a spectrum of topics related to the dynamic optimal control of the holistic CPS. Contributions in each area are highlighted below.

2.3.4.1 Anytime Control

Anytime control allows control solutions to be refined or improved as a function of available CPU time. These techniques are usually broken down into two improvement strategies: model reduction and performance reduction. In model reduction, the physical system is reduced by partial fraction expansion, modal reduction, or by weakly observable or controllable states. In this manner, we can prioritize which control inputs should be calculated given the available resources, or we can interrupt control optimization activities as required. In performance reduction the performance of the system is prioritized according to some performance index and the corresponding controls to achieve the performance indices are computed as resources become available [81].

The seminal work by Bhattacharya et al. [82] adapted anytime algorithm techniques to controller design using model reduction and a smooth switching algorithm. Most recently this type of control has been extended to utilize an optimal LQG controller to meet performance criteria when the resources are time-varying and not known *a priori* [83]. In this formulation an unconstrained and constrained formulation are developed and the latter is shown to be an adaptation of Receding Horizon Control. Our feedback control model is deterministic but in future work we anticipate the use of optimal control methods for which an anytime formulation may play an essential role.

2.3.4.2 Feedback Scheduling

In contrast to anytime control, wherein a control algorithm is designed to offer improved controller performance with increasing CPU time, feedback scheduling has become popular as a way of adjusting cyber resources based on the needs of the cyber system, including

the control algorithm [55]. Feedback scheduling adapts traditional control theory to regulate the cyber system which in turn contributes to regulating the CPS as a whole.

In this paradigm, the feedback scheduler manages resources by allocating CPU and communication resources to each control task as needed. In this scheme models are presumed to exist that relate sampling rate with controller performance [84]. Much work has been done by Cervin et al. in [85] to create a sound framework for feedback scheduling of control systems.

Such algorithms are often computationally intense. A model that directly incorporates the cost of control performance as it relates to cyber system resources would provide an excellent tool for feedback scheduling algorithms which can then utilize such information in choosing appropriate scheduling routines [86]. In our work, we devise a linear model relating sampling rate to controller performance. This effort is quite similar to past work in feedback scheduling. Our goal, however, is distinct, in that we seek to couple the resulting cyber system "dynamics" model to the physics-based model so that both can be coupled for co-regulation at each control loop cycle.

2.3.4.3 Networked Control Systems (NCS)

In a Networked Control System (NCS), feedback control loops are closed across a real-time network [61, 87]. Network communication is required to close a feedback loop whenever the sensor(s), actuator(s), and/or software are not co-located at the same physical processing unit. Typical NCS are faced with three primary problems: delay introduced due to limited bandwidth and competing control tasks, lack of synchronization between data integrated into each controller, and packet loss that may cause data to be unavailable for one or more control cycles.

Researchers have focused on a variety of issues associated with NCS, with much work focused on maintaining controller stability. Techniques from feedback scheduling (cited above) can be applied to assess the impact of delay on performance. Control scheduling techniques specifically addressing NCS issues have also been formulated. For example, [88] formally analyzes stability of a control system in the presence of delays and packet dropout, with a simulation also illustrating the effectiveness of clock synchronization compensation.

Our work is complementary to efforts in feedback scheduling and NCS. Research in feedback scheduling and NCS both offer a foundation for formally analyzing controller stability and in fact optimizing a real-time schedule over a set of controllers. This research, however, has focused on computing stability constraints through *offline* analyses, then capturing these constraints in a real-time processor and/or network scheduler. We instead focus

on a more tightly-coupled regulation process, where the rate of control loop execution is in fact regulated at each control cycle based on errors in physical state as well as an ideal "reference" control loop rate that might be computed *a priori* using methods from NCS and feedback scheduling.

2.3.4.4 Hybrid Systems

The control systems community has established a theory of hybrid systems to simultaneously capture continuous and discrete system states. In a hybrid system, a finite state machine represents discrete system modes potentially having different sets of dynamics, constraints, and controllers. This formulation has provided the ability to model systems that switch between different controllers, potentially with different task rates, and that "jump" or switch through discontinuities or nonlinearities [89, 90]. Control-theoretic analyses of hybrid systems has focused on characterizing reachability and guaranteeing stability of all reachable states. Stability has been an important topic in hybrid systems research and has followed traditional Lyapunov-based energy proofs [91]. Research in this area has primarily focused on handling the "jumps" typically representing nonlinearities in system dynamics rather than changes in control task execution rate.

Formally, a hybrid system, H, is defined by the tuple $H = \{Q, \Sigma, Inv, J, Init\}$, where set Q is the discrete state set, Σ is the collection of dynamical subsystems associated with states Q, set Inv represents invariants that must be true to remain in a particular state, mapping J represents state transition behaviors, and $Init$ represents the initial conditions, discrete and continuous. For this work, we abstract computational state to a differentiable representation in part through use of a simple hybrid systems formalism to handle a discontinuity in cyber state.

2.3.4.5 Time-Varying Sampling and Sensor Scheduling

Uncertainty in sampling rate can be caused by transmission delays in a NCS, jitter and/or missed deadlines in the RTS, etc. Research investigating the design of controllers under uncertain delays has resulted in more robust systems. Typically, as in NCS research, these approaches consider a small range of possible sampling rates and stability and robustness guarantees are given for that range under varying control schemes [63]. Successful optimal controllers under these circumstances using an Linear Matrix Inequality (LMI) approach have been designed [93, 94]. Rather than design controllers amidst uncertain sampling times we explicitly vary the sampling time and design robust controllers under that assumption. However, formal stability guarantees from research in time-varying sampling

could aid us in future work as we seek stability guarantees for our formulation.

Sensor scheduling is a technique used to determine which sensors or sensor modes should be read next to minimize error in the control system. [95, 96]. This often occurs where many sensors or sensor modes provide readings for similar phenomena. Markov Decision Process (MDP) formulations typically find an optimal policy for scheduling sensors [97–99]. In many control systems sensors are read at a higher rate than the control is output to the actuators guaranteeing up-to-date measurements of physical system state. Such a technique would benefit our work by allowing another task in the RTS to make decisions about when to execute the control task and output the next control signal even if the control task rate is very low.

2.3.4.6 Optimal Sampling Patterns

The research most related to our work has originated in research in event-triggered control and time-varying control and sampling to reduce the number of sampling instants. Bini et al. recently proposed an optimal control formulation to optimize both control inputs and sampling pattern trajectory, a computationally-feasible quanitzation-based method to estimate or approximate the optimal control solution, and proved optimality for first order systems [13]. Varying Time Control (VTC) is proposed by Kowalska et al. wherein a similar optimal control problem over control inputs and sampling instants is solved for a receding horizon with a computationally tractable algorithm [100] but loss of optimality guarantee [13]. Our work is similar, it allows for variable sampling instants, but whereas their work focuses on optimality over a planned trajectory our technique focuses on increasing robustness to system disturbances and deviations from planned trajectories through proportional feedback control which determines the sampling rate. Additionally, our feedback co-regulation scheme could be used to supplement optimal sampling pattern techniques by accepting the optimal sampling pattern as the reference trajectory and using feedback co-regulation to offer minor adjustments based on aberrant conditions.

2.4 CPS Topics in Aerospace

Safety-critical Aerospace systems require task schedules executing on real-time operating systems that have been analyzed offline to show hard deadlines are met and that soft real-time tasks will receive sufficient attention for effective mission accomplishment. In low-cost systems such as UAS or CubeSat platforms, code may be executed in a best-effort mode with non-real-time operating system such as an embedded Linux distribution. Such a

simple execution strategy can be successful so long as tasks under-utilize available cyber resources or the missing of deadlines does not place the system at risk. As more sophisticated logic such as image processing and flight plan optimization algorithms migrate on-board to improve mission data collection and robustness to scenarios such as lost link and/or loss of Global Positioning System (GPS), resources will no longer be under-utilized thus must be carefully managed in real-time.

Large spacecraft systems have typically addressed the problem of physical and cyber resource utilization through careful task scheduling. For an orbiting spacecraft, science payload data collection must often occur within a relatively short time window (e.g. a few minutes for Low Earth Orbit (LEO) [101]). During this window the system must maximize its efforts to collect science data. Traditionally such task scheduling problems have been addressed by ground operators manually constructing plans with write and check procedures [101]. EO-1 is the first of a series of NASA missions entitled "EO" (Earth Observer) targeting both science and technology demonstration goals. The Continuous Activity Scheduling, Planning, Execution, and Replanning (CASPER) planner was used onboard EO-1 to optimize science activities based on incoming data [102]. An iterative repair algorithm was used to improve task execution schedule. This science planner was highly successful, and has continued to evolve for infusion into additional missions. Other planners include the Automated Scheduling and Planning ENvironment (ASPEN) where scheduling is combined with mission planning [103], the Remote Agent [104], and the Heuristic Scheduling Testbed System (HSTS) [105]. Bataille et al. examine and design for physical constraints, fairness, and efficiency for different agents using a shared resource (an earth observing satellite) [106]. In work by Bresina et al. two techniques, GenH which generates a specialized search heuristic, and Heuristic-Biased Stochastic Sampling (HBSS) which employs the heuristic within a stochastic sampling method are combined together to automatically generate high quality schedules with respect to an objective function [107].

Agrawal, Cofer, and Samad explore some of the reasons why more advanced control algorithms are not used in modern aircraft and spacecraft avionics system [108]. They conclude that a Quality of Service (QoS) approach [109, 110] is needed to address the problem and they propose an adaptive resource management scheme for a real-time avionics system using anytime control and accompanying nontraditional task scheduling. Russ and Stütz recently proposed a higher level style of resource management that includes task based guidance, navigation, and perception plans. Their method focuses on finding algorithmic solutions adapting to perceptual demands that vary during flight as well as balancing those demands with sensory and computational resources [111]. Narayan et al. present a novel computationally adaptive trajectory decision optimization system that can dynam-

ically manage, calculate, and schedule task execution parameters [112]. An offline and online component work together to increase overall mission efficiency.

2.5 Concluding Remarks

All or most of the work presented in this chapter may be used as a baseline for developing solutions to the problem of controlling physical systems using digital devices. We present this related work primarily to create awareness for the reader that the work presented in Chapter 3 and 4 is novel in its coupling of cyber and physical systems via the *equations of motion* rather than incorporating the delays of motion into the models used for task scheduling. That is, at the feedback control level, cyber and physical resources are balanced dynamically rather than at a higher planning level presumed in [13, 100] and in traditional satellite task scheduling. Our approach does not replace traditional planning, but rather supplements it by allowing reactive reallocation of resources within the reference trajectories commanded by the planner. Our work in Chapter 5 complements existing research by providing a mechanism to optimize over cyber and physical resources while leveraging optimization theory and more particularly optimal control [113]. Existing solutions to dynamically adjust parameters can provide the tools by which a system could use our methodology to produce more efficient missions according to the individual metrics chosen in the cost function.

CHAPTER 3

Toward Continuous State-Space Regulation of Coupled Cyber-Physical Systems

Efficiency and performance are motivators for CPS co-design. Historically, the energy required to actuate physical effectors has dominated energy requirements of any cyber system. We are entering an era, however, where handheld and embedded devices are computationally advanced, with fast, distributed, networked processors demanding nontrivial energy. A physical system must therefore be optimized over its computational resources and vice versa, a synergy consistent with awareness in biologic entities. It has become imperative for any resource scheduler to be aware of if, and when, it can scale back resources devoted to reasoning about the physical system and still maintain good quality of control. In terms of performance of the physical system, an increase in resources allotted to the computation of control inputs will result in better performance of the physical system. This can occur either by scheduling additional processing time for physical system control tasks, or by increasing the processor frequency.

In this chapter we focus on augmentation of the "physical" controller depicted in Figure 1.1 to better support CPS co-design through co-regulation. We represent computational state in a continuous-time state formulation and derive an approximate or abstract continuous model that can be numerically coupled within a unified continuous-time mathematical representation. We then augment a simple physical system, a 2^{nd}-order damped oscillator, with this cyber model to illustrate how the cyber and physical systems are co-regulated. Once cast as a linear system, we then design a Linear Quadratic Regulator (LQR) law for the holistic CPS and demonstrate the closed loop co-regulation of the cyber and physical states. We evaluate results over a series of reference control loop rates, illustrating how co-regulation is particularly useful in cases where resource demands by other tasks are in direct competition with control loop execution rate to the extent of impacting system stability. We extend these results by applying this methodology to an unstable inverted pendulum system demonstrating the ability of our proposed technique to conserve resources while stabilizing

the system.

Below, we first present a simple physical system example to motivate CPS co-regulation. In Section 3.2 we describe our model of computational state as well as an approximate continuous cyber model and combine this with the physical model to form an augmented system in Section 3.3. We then design a closed-loop LQR controller and present simulation results in Section 3.4. Results compare a continuous-only simulation and the decoupled CPS model developed. Section 3.5 presents some preliminary work and results investigating one potential coupling scheme.

3.1 A Motivating Example

We present a physical system used throughout this chapter to first illustrate the impact of digital feedback control on a physical system and then to provide a baseline model for CPS co-regulation. The goal is for the reader to gain insight through a simple model into how the cyber system affects the response, stability, and control authority of the physical system and how we are able to model and leverage coupling between cyber and physical "effectors."

3.1.1 Spring-Mass-Damper System

A damped oscillator system may be represented as

$$\Sigma_p : \left\{ \begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p \,, \right. \tag{3.1}$$

where x_{p1} is the position and x_{p2} is the velocity of the mass. For this system, and throughout this chapter, we have chosen physical constants stiffness $k = 39.4784$, mass $m = 1$, and damping $c = 1.2566$. The eigenvalues for the system are $\lambda_{1,2} = -0.6283 \pm 6.2517\,j$. Note that since all the eigenvalues are in the Open Left Half Plane (OLHP) the system is stable. Our simulations in this chapter will, by default, show the response to initial conditions $\mathbf{x}_p(0) = \mathbf{x}_{p0} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. The plot of the open loop response is shown in Figure 3.1.

3.1.2 Closed-loop Controller

In designing control systems we have specifications that govern the transient and steady-state responses. Rise time, overshoot and undershoot, settling time, and steady-state value are usually designated as well as the inherent need to design a robust controller with robust

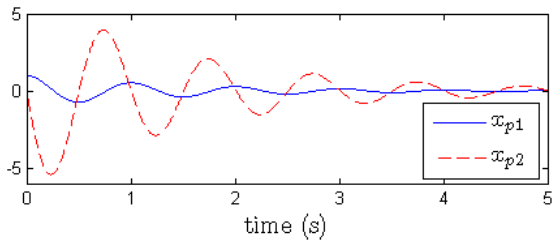


Figure 3.1: Open Loop Response

stability margins. For simplicity, and because we're interested in state regulation (driving the states to equilibrium), we design a proportional-derivative feedback controller assuming full-state feedback. Let

$$u_p = -k_{p1}x_{p1} - k_{p2}x_{p2}$$

be the control, where $k_{p1} = 3.5$, $k_{p2} = 2$ are gains chosen using pole placement techniques that give a desirable response. We can find the equivalent digital version of the system and controller assuming a sampling time $T_s = 0.1\,\text{s}$ using standard digital control techniques [60]. Simulating the continuous and digital systems and overlaying the responses, we obtain the plot in Figure 3.2. Note that the response has improved significantly relative to open-loop simulation. Also notice the zero-order hold (ZOH) nature of the system, an artifact of holding the input u_p at the last computed value throughout a control loop cycle, as well as only updating the state (output), y, once per cycle.

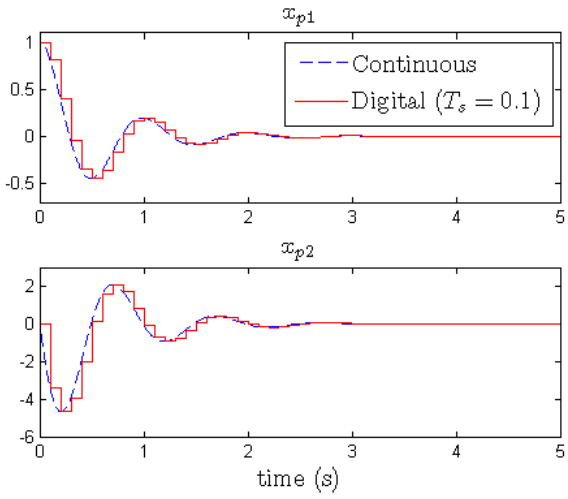


Figure 3.2: Closed-loop Response

Often, control systems can be executed at a sufficiently fast rate to ignore the effects of ZOH thereby assuming properties of a continuous system. In this chapter we provide continuous-time trends as a reference to illustrate the impact of ZOH over different intervals

on our system. However, in a co-regulated CPS we seek to minimize use of computational resource "energy" as much as we seek to minimize use of physical actuation "energy." We therefore more carefully model the time required for the control loop cycle and discuss how this might change. Let T_{si} be the time required to calculate a control input u_p, and $T_{sc} \ll T_{si}$ be the sampling time of the A/D and D/A converters [114]. Then the discrete nature of the physical system will be determined by the full sampling period

$$T_s = T_{si} + T_{sc}. \tag{3.2}$$

From the perspective of the physical system, an appropriate sampling rate is chosen such that requirements for smoothness, stability, and tracking are met. It is often recommended that the sampling rate, ω_s, be chosen such that $\omega_s/\omega_b > 20$, where ω_b is the desired bandwidth [60]. If a constant gain is held as sampling rate decreases (delay), tracking accuracy and stability decrease. From the cyber perspective, however, we would prefer the lowest sampling rate to allow the cyber system to attend to other tasks, translating to the control task occupying a lower percentage processor utilization. Alternatively, decreasing control loop rate might involve slowing a variable-speed processor or "turning off" one or more processors in a multi-core architecture to reduce energy use.

As a result of these competing objectives, a tightly coupled unified model that enables co-regulation of cyber and physical states would allow us to co-design the CPS and optimally exchange control authority of the physical system for cyber resource allocation.

3.2 Development of a Cyber Model

To augment the physical system continuous state vector with information about the cyber system, we develop a continuous abstraction representing the computational properties of the cyber system with relevance to co-regulation. For this work we assume T_s in Equation (3.2) is directly controllable by changing the sampling rate of the D/A converters, altering the fraction of CPU utilization allocated to the control task, modifying the frequency of computations (clock speed), or a combination of these. Although in any real-time operating system (RTOS) tasks will context switch, we approximate the rate of progress through the control loop, data acquisition, and computational processes to be constant, "resetting" to the beginning of the cycle each time a new control command is generated. This results in a step-wise linear relationship between progress through the control loop and elapsed time, with slope varying as a function of control loop frequency or rate.

Definition: Let y_c represent the fraction of all control loop data acquisition and calcu-

lation activities completed. Also, let $f = 1/T_s$ be the frequency at which the control loop is executed and let t be time. Then

$$\begin{aligned} y_c &= ft - n(t) \text{ for } n(t)T_s \le t \le (n(t)+1)T_s \\ n(t) &= \lfloor t/T_s \rfloor \end{aligned}$$

This relationship implies that $\dot{y}_c = f$ for $n(t)T_s \le t \le (n(t)+1)T_s$. Note that $n(t)$ represents the cycle or control loop count. In Figure 3.3 we show a plot of y_c as a function of time (with T_s units) as it would appear in a cyber system with f at a constant value.

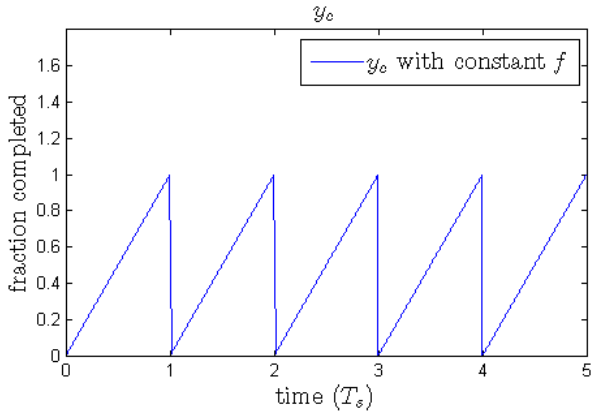


Figure 3.3: Fraction of Completed Control Task, y_c, as a Function of Time

Figure 3.4 shows y_c plotted for varied f_i to illustrate how regulation of T_s, recomputed once during each control loop cycle, impacts progress through the control loop over time.

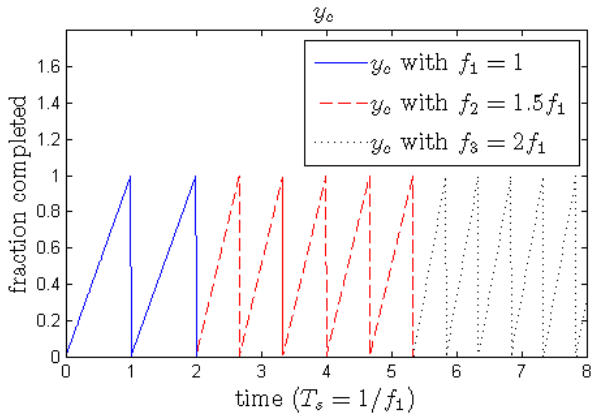


Figure 3.4: Fraction of Completed Control Task, y_c, as a Function of Time with Varying f

As a continuous abstraction to an inherently discrete model, we consider the dynamics of cyber state y_c to be modeled as an open loop double integrator subject to an impulsive

"forcing" term u_c capable of instantaneously changing $\dot{y}_c$ to a new rate at the end of each control cycle. Note that impulsive u_c is realistic because it represents an instantaneous change in task execution "velocity" or rate rather than a change in physical velocity. Once u_c updates $\dot{y}_c$ and y_c is reset to zero indicating a new controller cycle has begun, the controller process is executed under a ZOH condition with $u_c = 0$ such that y_c increases linearly over sampling period T_s.

The ideal sawtooth model described above represents impulsive u_c, possible if the real-time scheduler is indeed able to instantly reconfigure the CPU or CPUs accordingly. However, in practice, particularly when numerous real-time tasks must be managed potentially across multiple cores, the scheduler will need time to adjust tasks and resources. For this work, we presume such change to the task schedule will be incremental in nature, analogous to a physical system where an "accelerated" mass changes its rate (velocity) slowly rather than instantly. Drawing inspiration from the behavior of physics-based systems, we propose the following linear systems model of our control loop "cyber" system:

$$\Sigma_c : \left\{ \begin{bmatrix} \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_c \right. \tag{3.3}$$

where x_{c1} is the fraction of control loop calculations completed in this cycle, and x_{c2} is the frequency at which the control loop executes. As previously defined u_c represents the "force" or "acceleration" required to adjust cyber resources.

This model only applies during one sampling period T_s, requiring a discrete jump at the end of the control task. The discrete nature of the cyber model can be modeled using a hybrid automaton which we present in Section 3.3.

3.3 CPS Model

3.3.1 Augmented System

We can combine the physical model in (3.1) and the cyber model in (3.3) by combining the continuous-valued states into a single stacked vector. This gives

$$\begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & ? & ? \\ -\frac{k}{m} & -\frac{c}{m} & ? & ? \\ ? & ? & 0 & 1 \\ ? & ? & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix}. \tag{3.4}$$

where the "?" indicates terms that might couple cyber and physical system dynamics.

It is clear from the discussion in Section 3.1.2 that the physical system is affected by sampling time T_s of the cyber system. Additionally, the cyber system should respond to demands in the physical system. As a result, the coupling terms, (?), in Equation (3.4) need to be addressed. However, we initially assume a decoupled CPS model in which the only effect the cyber system has on the physical system is through the ZOH. This results in a final CPS model

$$\Sigma_{CPS}: \left\{ \begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix}. \right. \tag{3.5}$$

3.3.2 Hybrid Automaton Formulation

We employ a hybrid automaton to model the state jump (reset) of x_{c1} for the decoupled system as shown in Figure 3.5. During each sampling period, the input vector, $\mathbf{u} = \left[u_p, u_c\right]^T$,

$$J: \left[x_{p1}^+ = x_{p1}^-; x_{p2}^+ = x_{p2}^-; x_{c1}^+ = 0; x_{c2}^+ = x_{c2}^-\right]$$

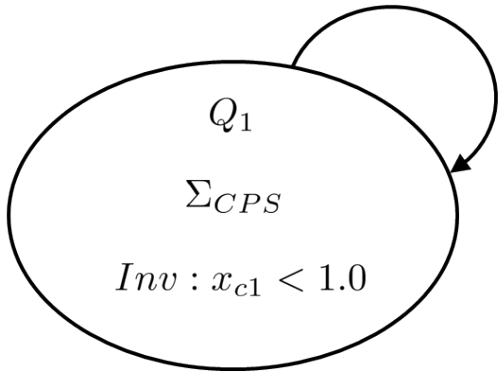


Figure 3.5: Hybrid System Model

is held constant. At the end of each sampling period, x_{c1} jumps back to zero.

3.3.3 Implementation Details

The hybrid automaton in Figure 3.5 was implemented in MATLAB wherein, at each cycle, we use a 4^{th}-order Runge-Kutta variable time step ordinary differential equation solver. As x_{c2} changes according to the CPS system dynamics and desired reference frequency, the length of a cycle, and hence range of integration, varies and appropriately captures

the changing dynamics of the cyber system. The ZOH nature of the CPS is captured by holding the input constant until the next cycle. Additionally, a cyber system cannot have negative state values even though this would be allowed by our continuous state model. To address this, we set $x_{c1} = 0$ in our simulation as it currently has no bearing on the CPS due to the ZOH. We also artificially restrict x_{c2} to a lower bound on sampling rate representing the point of instability. Although this lower limit on x_{c2} introduces a nonlinearity into the system, it is analogous to saturation in a linear system and can be analyzed similarly. In our following simulation we set this threshold to $x_{c2,min} = 3.33\,\text{Hz}$ as that is the frequency at which the spring-mass system response approaches instability in our simulations.

The following pseudocode summarizes our MATLAB-based implementation:

Algorithm 3.1 Algorithm for Simulation of a CPS

while $t < t_{max}$ **do**
  u=-K*$\mathbf{x}_{prev}$
  tspan=[t_{prev}, $t_{prev} + x_{c2}$]
  [t,x]=ode45(@CPSmodel,t_{span},$\mathbf{x}_{prev}$)
  t_{all}=[t_{all};t]
  $\mathbf{x}_{all}$=[$\mathbf{x}_{all}$;x]
end while

Note that because MATLAB's `ode45` is a one-step solver, the piecing together of multiple executions of `ode45` based only upon x_{prev} is a justifiable mechanism for obtaining the true integrated solution (to within numerical method precision).

3.4 Closed-Loop Co-Regulation of Cyber and Physical States

We employed Linear-Quadratic Regulator (LQR) controller designs in all simulation results presented below. We examined the performance of two different systems, the 2^{nd}-order damped oscillator system presented and an unstable system representing an inverted pendulum. In this initial work, we only examine decoupled systems as presented in Equation (3.5) demonstrating the power of our CPS abstraction to enable the application of traditional control theory.

3.4.1 2^{nd}-order Damped Oscillator

In this section we present results for the spring mass damper system in Equation (3.5).

3.4.1.1 LQR Design

Since we have formulated our CPS as a linear system, we can apply traditional linear feedback control design techniques. We have designed an infinite horizon LQR controller which minimizes

$$J = \int_{t_0}^{t_1} \left(\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) dt$$

We add two integrator states to the system and the LQR gain is solved for the augmented system

$$\begin{bmatrix} \dot{\mathbf{e}} \\ \dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{A}_{\Sigma_{CPS}} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_{\Sigma_{CPS}} \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix} + \begin{bmatrix} \mathbf{r} \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{r} = \begin{bmatrix} -x_{p1,r} \\ -x_{c2,r} \end{bmatrix}$$

where $x_{p1,r}$ and $x_{c2,r}$ are the reference inputs for the physical and cyber system respectively.

In choosing appropriate $\mathbf{Q}$ and $\mathbf{R}$ matrices we face a tradeoff. The double-integrator representing the cyber system can have fast rise time indicating that the cyber system is able to adjust resources very quickly. This corresponds to a large $\mathbf{Q}$ and small $\mathbf{R}$. However, with this design the double integrator will always overshoot the reference frequency, a condition that worsens as the gains are increased. If we choose small $\mathbf{Q}$ and large $\mathbf{R}$ we can make a smoother transition with little overshoot, but this is undesirable if the cyber system must rapidly respond to stabilize the physical system. Our chosen values of $\mathbf{Q}$ and $\mathbf{R}$,

$$\mathbf{Q} = \left[\begin{array}{cc|c} 150 \times 10^3 & 0 & \\ 0 & 1 \times 10^3 & \\ \hline & & \mathbf{0} \end{array}\right], \quad \mathbf{R} = \begin{bmatrix} 3 & 0 \\ 0 & 20 \end{bmatrix}.$$

in the simulations that follow place higher importance on response time since we artificially limit the lower bound on x_{c2} as described. The consequence is, however, that the system can spend a nontrivial time period at $x_{c2,min}$ causing poor quality of control of the physical system for a short time.

3.4.1.2 Results

We present results from three test cases given in Table 3.1. In these results we will compare

Table 3.1: Test Cases for 2^{nd}-Order Damped Oscillator

CASE	$x_{p1,0}$	$x_{p2,0}$	$x_{p1,r}$	$x_{p2,r}$	$x_{c2,0}$	$x_{c2,r}$
1	1 m	0 m/s	0 m	0 m/s	200 Hz	80 Hz
2	1 m	0 m/s	0 m	0 m/s	3.4 Hz	6 Hz
3	1 m	0 m/s	0 m	0 m/s	20 Hz	5 Hz

a traditional continuous-only simulation (allowing for negative values of the cyber states, and no ZOH) and our CPS simulation shown in Algorithm 3.1.

2^{nd}-Order Damped Oscillator Case 1 In the first case the initial frequency, $x_{c2,0}$ is over twice as high as $x_{c2,r}$. However, both values are sufficiently fast with respect to the destabilization frequency that the system suffers no significant loss of quality of control due to the co-regulation of physical force and control loop rate. Indeed, at these high sampling rates the CPS simulation approaches a continuous system as seen in Figure 3.6.

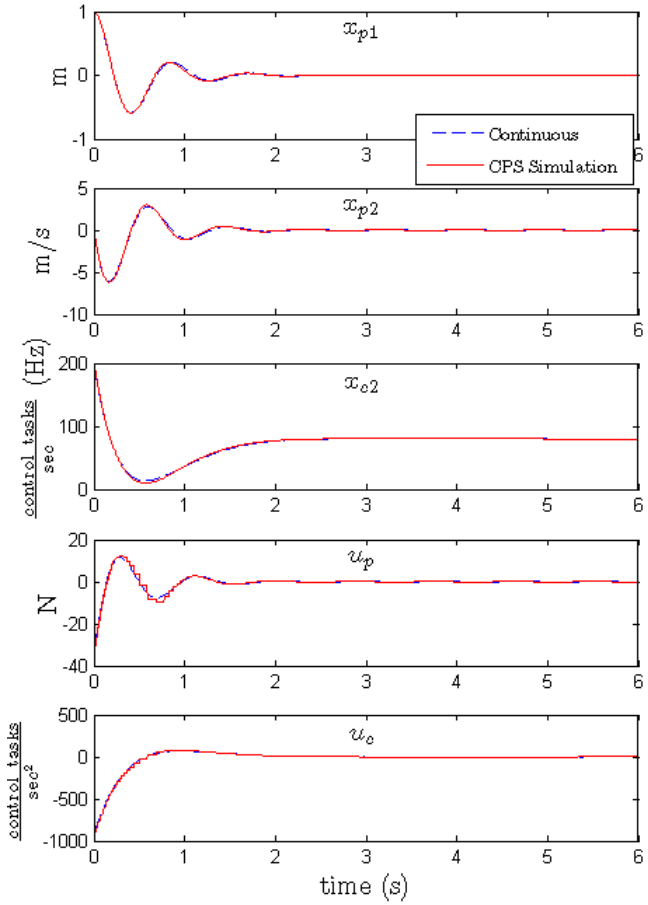


Figure 3.6: Simulation for Case 1. $x_{c2,0} = 200$ Hz, $x_{c2,r} = 80$ Hz

2^{nd}-Order Damped Oscillator Case 2 In the next case, shown in Figure 3.7, the initial frequency is very near the lower stability threshold $x_{c2,min}$, a condition that might occur in processing overload or critical energy conservation situations. In the presence of an appropriate guidance law, the cyber system would detect the impending instability and command a higher reference frequency, $x_{c2,r}$ while balancing the need to conserve cyber resources. We choose $x_{c2,r} = 6\,\text{Hz}$ to simulate this scenario.

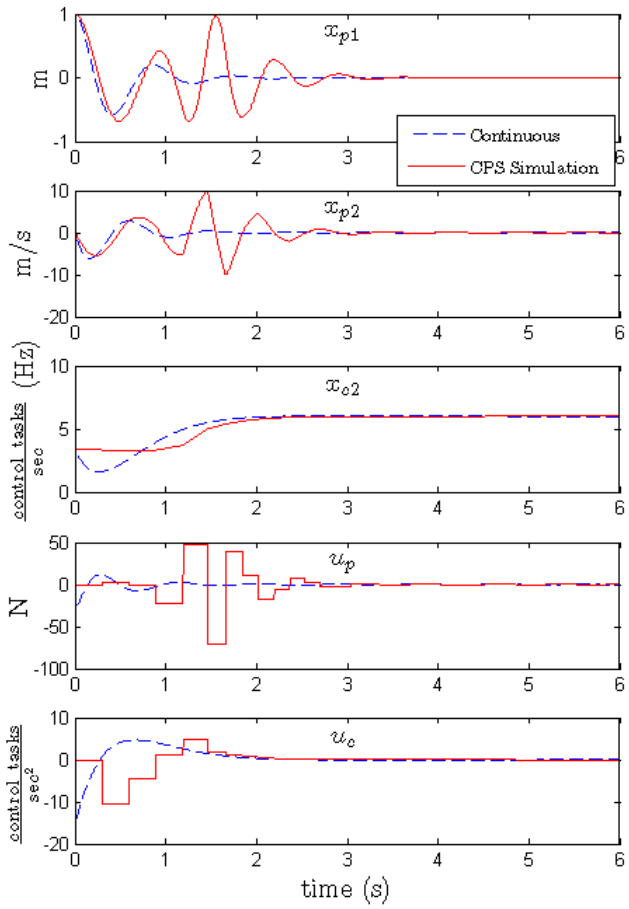


Figure 3.7: Simulation for Case 2. $x_{c2,0} = 3.4\,\text{Hz}$, $x_{c2,r} = 6\,\text{Hz}$

Note that at the extremely low sampling rate ($x_{c2} = 3.4\,\text{Hz}$) the system approaches instability due to the ZOH. As the cyber system directs more resources (higher control loop rate) to the physical system, it is able to quickly regain control.

2^{nd}-Order Damped Oscillator Case 3 In this test case we approach the energy conscious sampling rate from above. The results are shown in Figure 3.8. In this case, because we saturate at $x_{c2,min} = 3.33\,\text{Hz}$, the system nearly goes unstable, but because we have commanded $x_{c2,r} = 5\,\text{Hz}$ the CPS rate increases and the system is stabilized. Note that the

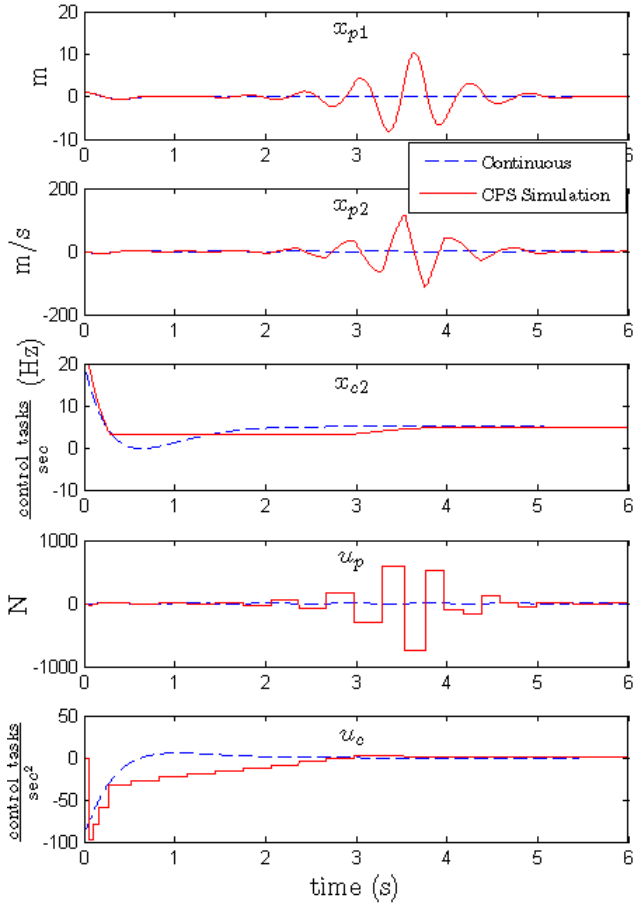


Figure 3.8: Simulation for Case 3. $x_{c2,0} = 20\,\text{Hz}$, $x_{c2,r} = 5\,\text{Hz}$

system spends a nontrivial amount of time at or near $x_{c2,min}$. As in Case 2, this is because the corresponding sampling time is 0.3 s, requiring multiple sampling periods for the integral control to respond. In this case, this is likely a limitation of using the double integrator as a cyber model abstraction. However, if the correct coupling terms were known, we have evidence suggesting that the controller would drive the system to $x_{c2,r}$ more quickly.

3.4.2 Unstable System Results

As a more challenging system that requires active control for stability, in this section present results for an inverted pendulum by applying the same techniques applied to the spring-mass system.

3.4.2.1 Inverted Pendulum

The augmented inverted pendulum system is

$$\begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g}{l} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix} \tag{3.6}$$

where $g = 9.8\,\mathrm{m/s^2}$ is gravity, and $l = 1\,\mathrm{m}$ is the length of the pendulum. The system is unstable indicated by having an eigenvalue $\lambda = 3.13$ in the Open Right Half Plane (ORHP). This means that the response to any initial condition not precisely at the equilibrium, zero in this case, requires a stabilizing controller for convergence.

3.4.2.2 LQR Design

We again use an LQR controller formulation to regulate the inverted pendulum to its equilibrium from a nearby initial condition. Again we include integrator states to ensure we reach the desired reference values and then find the LQR gain for the decoupled system using

$$\mathbf{Q} = \left[\begin{array}{cc|c} 4000 & 0 & \\ 0 & 100 & \\ \hline & & \mathbf{0} \end{array}\right], \quad \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

3.4.2.3 Results

To simulate the inverted pendulum we computed the system response to initial conditions $\left[x_{p1}, x_{p2}\right] = [0.17, 0]^T$ in radians. We determined the lower threshold for sampling rate needed to maintain stability to be $x_{c2,min} = 6.66\,\mathrm{Hz}$. We present results from the three test cases given in Table 3.2.

Table 3.2: Test Cases for Inverted Pendulum

CASE	$x_{p1,0}$	$x_{p2,0}$	$x_{p1,r}$	$x_{p2,r}$	$x_{c2,0}$	$x_{c2,r}$
1	0.17 rad	0 rad/s	0 rad	0 rad/s	200 Hz	80 Hz
2	0.17 rad	0 rad/s	0 rad	0 rad/s	6.66 Hz	13 Hz
3	0.17 rad	0 rad/s	0 rad	0 rad/s	20 Hz	7 Hz

In these results we again compare a continuous simulation with our CPS simulation of the decoupled system in Equation (3.6).

Inverted Pendulum Case 1 This test case (Figure 3.9) is the same as Case 1 for the spring-mass system above. Both the initial sampling rate and the reference sampling rate are much higher than $x_{c2,min}$. Both x_{p1} and x_{p2} closely track the continuous response. At one point, x_{c2} dips to $x_{c2,min}$ as a result of the overshoot. This causes a sampling period that would result in instability if not corrected, as well as the small deviation from the continuous response. But this low sampling rate only occurs for one period, after which x_{c2} quickly rises to $x_{c2,r}$.

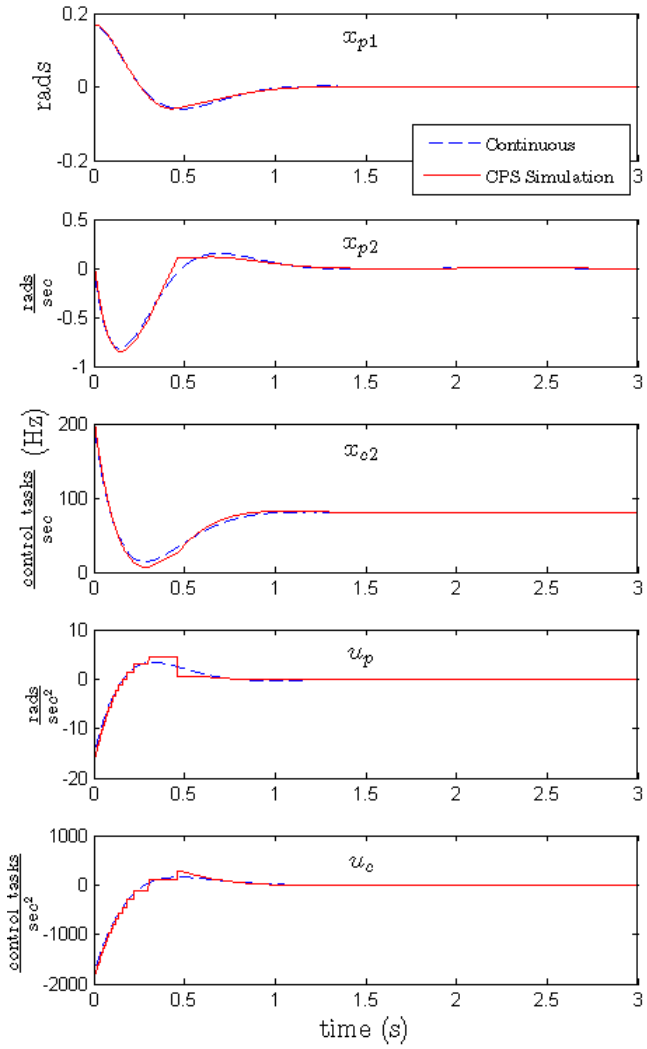


Figure 3.9: Simulation for Case 1. $x_{c2,0} = 200\,\text{Hz}$, $x_{c2,r} = 80\,\text{Hz}$

Inverted Pendulum Case 2 Similar to Case 2 for the spring-mass system, in this test we move from an energy conscious sampling rate near $x_{c2,min}$ to a faster rate in response to a disturbance in x_{p1} (represented by our initial conditions). The results in Figure 3.10 demon-

strate that the cyber system is able to adjust its resources commanding a faster sampling rate to stabilize the system.

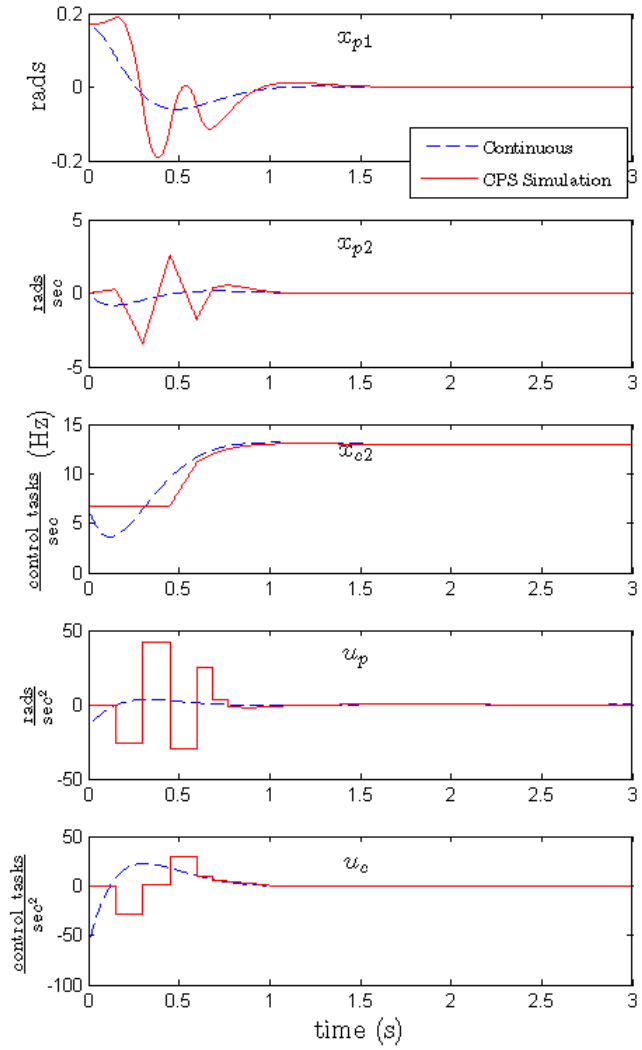


Figure 3.10: Simulation for Case 2. $x_{c2,0} = 6.66\,\text{Hz}$, $x_{c2,r} = 13\,\text{Hz}$

The initial upward trajectory of x_{p1} occurs as a result of holding $u_{p,0}$ (initial control input) at zero for one sampling period. Additionally, although the system is able to stabilize the system, the cost in control input is very high. In a real CPS it is likely our actuators would be unable to deliver such a control input. This again suggests the need for determining the correct coupling terms, "?," in Equation (3.4) so that the cyber system responds more quickly, and smaller control inputs are required.

Inverted Pendulum Case 3 In this test, similar to Case 3 for the spring-mass system, we begin with a relatively high sampling rate and command a more energy conscious rate

to allow the reallocation of resources in the cyber system. Note that $x_{c2,r}$ is very close to $x_{c2,min}$. Nevertheless, as seen in Figure 3.11 the CPS system regulates the states well and keeps it stabilized even at the low sampling rate.

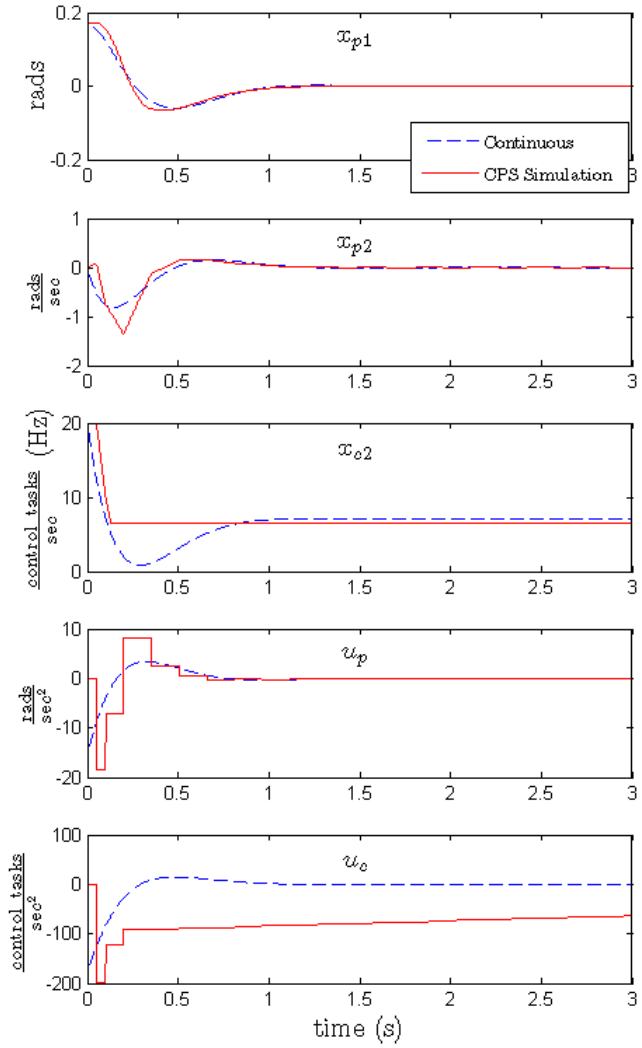


Figure 3.11: Simulation for Case 3. $x_{c2,0} = 20\,\text{Hz}$, $x_{c2,r} = 7\,\text{Hz}$

In this case, u_p is not very large, and the physical states x_{p1} and x_{p2} track the continuous response well despite a sampling rate near instability.

3.5 Toward a Coupled Model

It is apparent that the appropriate coupling terms, "?" in Equation (3.4), are needed to further improve system performance and make this CPS representation a viable asset to the

CPS community. In this section we explore some coupling options and investigate how they impact the system.

As an early investigation of how we might couple cyber state $[x_{c1}, x_{c2}]$ with physical (oscillator) state $\left[x_{p1}, x_{p2}\right]$ of the stable 2^{nd}-order oscillator system, we examine how the quality of control of the physical system is impacted by changing the frequency of the cyber system.

To do this, we propose an average error metric for quality of control. We define $\mu(f_s)$ as the average error in response of the physical system over a specified time interval for sampling frequency f_s in Hz. For our 2^{nd}-order system, we compute $\mu(f_s)$ over time interval [0.0 6.0] seconds for f_s between 2.2 Hz ($T_s = 0.45$ s) and 100 Hz ($T_s = 0.01$ s). This range in f_s is sufficient to illustrate error trends from a slow unstable rate (2.2 Hz) to a rate sufficiently fast to approximate the ideal continuous response (100 Hz).

To define $\mu(f_s)$, we performed a series of 45 simulations with equally spaced T_s. For each simulation, we computed the response of the 2^{nd}-order physical system using a 4^{th}-order Runge-Kutta method with integration step $\Delta t = 0.001$ s which yielded $n = 6/\Delta t = 6000$ data points over the six second simulation time. To simulate the particular f_s of interest, force input u_p was held constant between controller updates made every $T_s = 1/f_s$ seconds.

To compute average error for each simulation run over a particular f_s, we computed errors in the continuous physical states for each of the n simulation time points t_i:

$$e_{p1}(t_i) = x_{p1,r}(t_i) - x_{p1}(t_i)$$
$$e_{p2}(t_i) = x_{p2,r}(t_i) - x_{p2}(t_i)$$

where $x_{p1,r}$, $x_{p2,r}$ are the control system reference states. Although the reference states could change over time, they are held constant after an initial step change for our simulations. We then define average errors for physical states p_1 and p_2:

$$\mu_{p1}(f_s) = \frac{1}{n}\sum_{i=1}^{n} e_{p1}^2(t_i)$$
$$\mu_{p2}(f_s) = \frac{1}{n}\sum_{i=1}^{n} e_{p2}^2(t_i)$$

Figure 3.12 shows average errors μ_{p1} and μ_{p2} as a function of f_s. We note that these curves are consistent with results from other researchers in this area [109].

The average error of the system depends nonlinearly on the frequency as expected. However, we can approximate the curve using two linear fits by dividing the curve at a switching frequency $f_{sw} = 6.173$ Hz. As shown in Figure 3.12, velocity (p_2) is more sen-

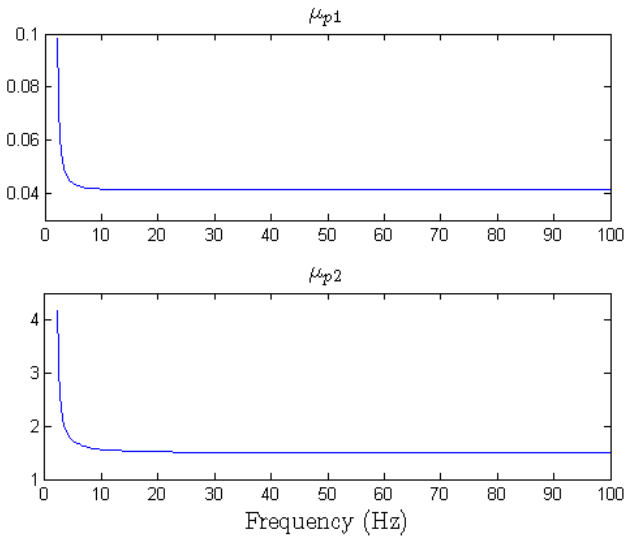


Figure 3.12: μ_{p1} and μ_{p2} as a Function of Frequency

sitive to frequency than position (p_2). The nearly horizontal portion of the curves indicate that at higher frequencies ($f > f_{sw}$) there is little effect of the frequency on the physical system, hence we use a linear fit with constant error (zero slope) for frequencies above f_{sw}. For the lower frequencies ($f \leq f_{sw}$), we apply a linear curve fit to obtain coefficients

$$c_{x_{p1},x_{c2}} = -0.01008, \quad c_{x_{p2},x_{c2}} = -0.4852 \tag{3.7}$$

where $c_{x_{p1},x_{c2}}$ is the dependence of x_{p1} on x_{c2} (cyber or control loop frequency), and $c_{x_{p2},x_{c2}}$ is the dependence of x_{p2} on x_{c2}.

3.5.1 Early Coupling Results

We now combine the augmented CPS model in Equation (3.4) with our linear fit from Equation (3.7). Because we used a two segment linear fit to characterize the dependence of the physical system on the cyber system, we have two sets of continuous dynamics

$$\Sigma_{CPS,1} = \Sigma_{CPS} \text{ in Equation (3.5)} \tag{3.8a}$$

$$\Sigma_{CPS,2} = \Sigma_{CPS} \text{ where } \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & c_{x_{p1},x_{c2}} \\ -\frac{k}{m} & -\frac{c}{m} & 0 & c_{x_{p2},x_{c2}} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.8b}$$

We employ a hybrid automaton, seen in Figure 3.13, to switch between $\Sigma_{CPS,1}$ and $\Sigma_{CPS,2}$ in Equation (3.8). In this formulation two discrete modes represent the full CPS, $\Sigma_{CPS,1}$ applied when $x_{c2} > f_{sw}$ and $\Sigma_{CPS,2}$ applied when $x_{c2} \leq f_{sw}$. We tested this new hybrid

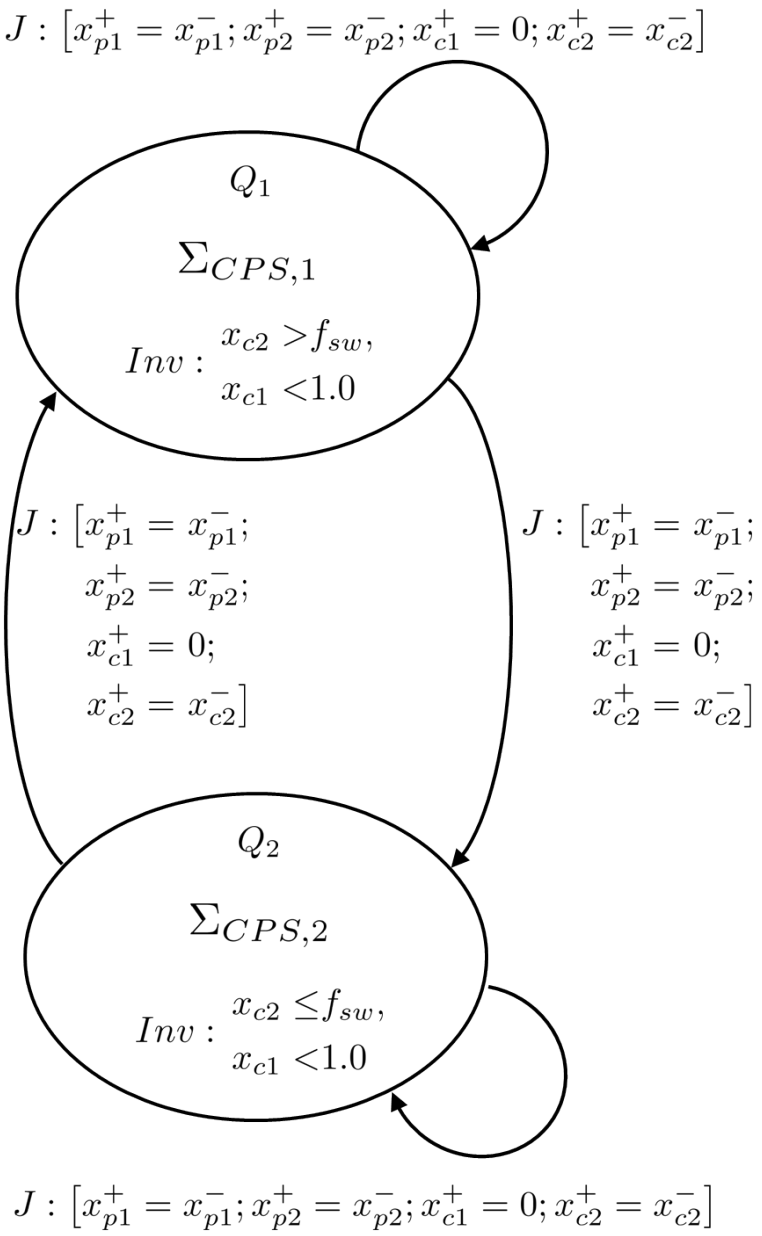


Figure 3.13: Hybrid System for Coupled CPS Model

automaton formulation using a MATLAB implementation similar to Algorithm 3.1 but allowing for switching discrete modes at f_{sw}.

3.5.1.1 2^{nd}-Order Oscillator

For the 2^{nd}-order oscillator we demonstrate the results shown in Figure 3.14 by simulating the third test case from Table 3.1. We compare the decoupled formulation simulated earlier with this preliminary hybrid switching formulation presented in this section.

In our testing, we used the values in Equation (3.7) as initial starting points for the coupling terms, but learned that tuned values $c_{x_{p1},x_{c2}} = -0.04$ and $c_{x_{p2},x_{c2}} = -0.6$ yielded improved results. In this simulation with the hybrid switching formulation x_{p1} and x_{p2}

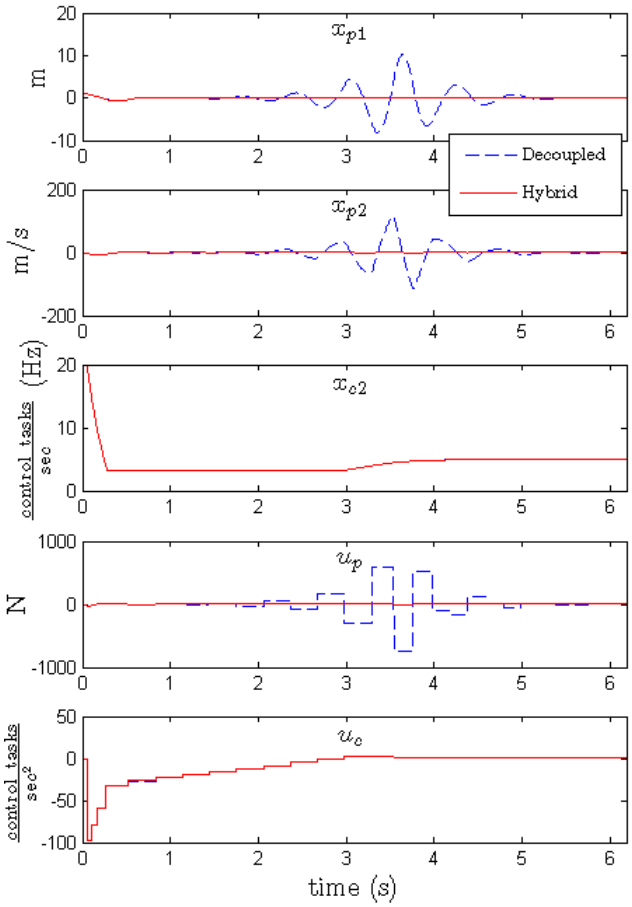


Figure 3.14: Simulation for 2^{nd}-order Oscillator with Coupling Terms. $x_{c2,0} = 20\,\text{Hz}$, $x_{c2,r} = 5\,\text{Hz}$

have much smaller error as they are regulated. Even more compelling is the very low control effort, u_p, required to obtain these results. This shows that at low frequencies using these coupling terms, we can control the system with smaller control efforts as the cyber system directs more resources to the control loop.

3.5.1.2 Inverted Pendulum

Despite the strong results just given, there is still work to examine how and when the cyber states should affect the physical system as evidenced by the results shown in Figure 3.15 for the inverted pendulum. In these results we used the same techniques described for the 2^{nd}-order oscillator system and tuned the coupling values to $c_{x_{p1},x_{c2}} = -0.03$ and $c_{x_{p2},x_{c2}} = -1.1$. The hybrid switching formulation for the inverted pendulum regulates the physical system

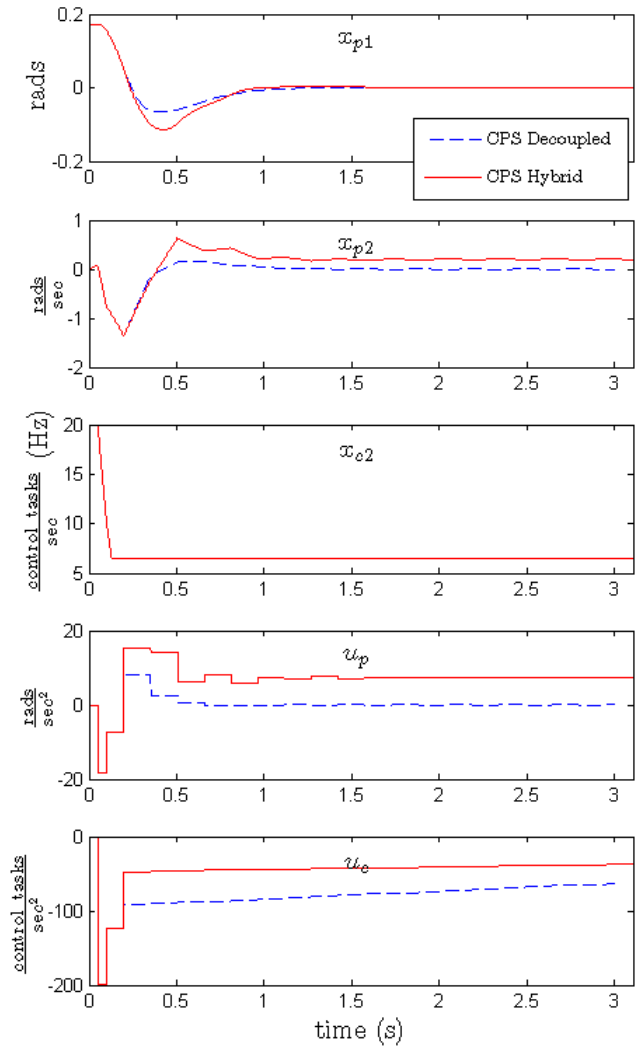


Figure 3.15: Simulation For Inverted Pendulum with Coupling Terms. $x_{c2,0} = 20\,\text{Hz}$, $x_{c2,r} = 7\,\text{Hz}$

states sufficiently (albeit with undesirable oscillations), but more concerning is that the control effort expenditure is significantly higher.

3.5.2 Future Coupling Objectives

Based on this preliminary investigation, we believe the coupling terms discussed in this section could be treated as a disturbance rather than coupling terms in the system matrix. Nevertheless, this coupling itself seems less compelling than the coupling that models the cyber system state dependency on the physical system states $\left(\text{e.g. } c_{x_{c2},x_{p1}},\ c_{x_{c2},x_{p2}}\right)$. Because the cyber states are an abstraction and don't adhere to the same laws as the physical system, and because we have more direct control over the cyber system, we believe the strongest results will lie in the development of these latter coupling terms.

3.6 Conclusions

We have presented a novel CPS representation in which physical and computational systems are represented as a single continuous multi-variable linear system. This representation enables co-regulation of physical and computational state to optimally balance computational load with physical system stability and disturbance rejection at each control loop cycle. A 2^{nd}-order oscillator system was used as an illustrative physical model, and a continuous cyber model abstraction of the associated single-thread control task executing on a CPU was developed. We developed a simulation environment for the CPS modeling technique presented and have shown results both for the 2^{nd}-order oscillator developed as well as for an unstable inverted pendulum. These baseline models were augmented further with coupling terms derived from curves representing quality of control as a function of sampling rate. Two contrasting results were given demonstrating that appropriate coupling can provide robustness to the physical system response in situations where computational resources must be conserved, but that more work is needed to understand how to appropriately select those terms.

Ideally, in a more complex CPS a similar coupled continuous representation could be developed containing both the dynamics of the physical system and the abstraction of the cyber system. This would allow the use of the powerful set of techniques, theory, and results from modern control theory to be applied to the CPS holistically. Perhaps most attractive would be the ability to develop optimal control strategies to balance goals for both the physical and cyber states of the full system much like that done in [13, 100]. However, much work remains before the proposed model will be of practical use as a complement to existing scheduling and modeling strategies. First, although we have introduced terms for the dependency of the physical system on the cyber system, this coupling is incomplete. Exploration of the dependency of the cyber system on the physical system is necessary

to properly couple the CPS. Further, the proposed continuous cyber model will always overshoot the reference command, which may not be desirable in a real system. Cyber models need to be explored that can possibly reduce this overshoot or prevent it entirely. Finally, this work should be applied to more complex real-world systems.

CHAPTER 4

Cyber-Physical System Modeling and Co-Regulation of a CubeSat

In small robotic platforms such as miniature rovers, Micro-Air Vehicles (MAVs) or small satellites (CubeSats), power required for computing and communication can rival or exceed propulsion/locomotion system power. For such systems, it is therefore inefficient and potentially insufficient to regulate motion (e.g. over propulsive/locomotion commands) first and computational/communication resources independently or in response. It is equally ineffective in platforms where computational and locomotion resource requirements are comparable to optimize the real-time computing/communication task schedule first then "move as needed" to support the cyber mission.

This chapter builds on Chapter 3 and our previous work in [26, 115] by proposing the application of state-variable techniques to the real-time feedback co-regulation of physical actuation and real-time controller task rate of execution (or sampling rate) for attitude control of a small spacecraft (CubeSat). With this scheme, computational resources devoted to attitude control during quiescent periods can be directed to other tasks such as communication, data gathering/processing, or mission planning. Although resource reclamation of this type has already been identified as valuable to increase mission productivity, spacecraft missions to-date have yet to run guidance, navigation, and control (GNC) tasks less frequently than would be required in worst-case disturbance and maneuver scenarios [102]. To our knowledge this is the first time a dynamic sampling rate scheme has been proposed for any spacecraft.

We conduct a CubeSat case study simulating disturbance rejection to the 3-DOF attitude of the CubeSat which uses reaction microwheels as physical actuators for attitude control. The CubeSat has an onboard computer and RTOS with presumed schedulability constraints representing the cyber system. A modeling abstraction of control task execution rate is coupled to the state-space model for attitude control allowing the dynamic adjustment of that rate and forming a discrete-time varying Cyber-Physical System model. We apply two

new controllers to handle the discrete-time-varying system: a feedback controller where the gains are scheduled over the time-varying sampling rate of the system, and a forward-propagation Riccati-based controller. We also believe this to be the first time controller gains have been scheduled over a dynamically changing control task execution rate. We further hope to add empirical evidence of the utility of forward-propagation (and forward-integration) Riccati-based controllers, the full understanding of which remains an open question in control theory [116–119]. Finally, we evaluate coupled CPS performance in terms of physical tracking error, control effort, and CPU resource requirements for the control task.

In this chapter we first summarize CubeSat attitude dynamics and formulate a continuous-time linearized model followed by the discrete-time-varying CubeSat model that utilizes our CPS control strategy. A cyber model representing control task execution rate is coupled to the CubeSat model forming a closed-loop holistic CPS model. Evaluation metrics for coupled CPS performance are then presented. The CubeSat case study is then discussed and results are given focusing on an evaluation of physical state error, control effort (energy utilization), and cyber control task resource utilization relative to fixed-rate controllers.

4.1 CubeSat Equations of Motion

Attitude control of a class of picosatellites called "CubeSat" [120] is a compelling cyber-physical system (CPS) challenge because of the unstable system dynamics and widely-varying pointing accuracy requirements for data collection and communication versus quiescent drift periods. Typically science data can be collected much faster than it can be communicated, a problem confounded by constraints on orbital windows in which a ground station is accessible. This requires the CubeSat to devote substantial effort to manipulating data onboard, as was done with EO-1 [102], to improve science output. CubeSats, therefore, usually contain substantial computing power for their size. At any given time computational activities on a CubeSat can easily consume 10%-50%[1] of available energy resources, motivating the need for CPS co-design techniques that co-regulate both cyber and physical resources.

CubeSat missions are accomplished with a $1-3\,\text{kg}$ satellite containing major onboard subsystems such as attitude control, communication, power distribution, generation, and storage, command and data handling, and payload. Pointing may require rotational movements once or more per orbit depending upon the mission. A spacecraft in a $500\,\text{km}$ cir-

[1] Personal communication with Dr. James W. Cutler from the Michigan Exploration Laboratory at The University of Michigan.

cular orbit spends 38% of its time in eclipse meaning that energy can be generated during the other 62% of the orbital period. Since a typical time period for a 500 km altitude orbit is about 95 min, this poses a challenge for energy utilization. Data transmission requires energy that depends on multiple factors such as data rate, signal strength, antenna size and type, etc. These factors provide motivation for communication and position-aware computing. In this work we focus on making the cyber system (i.e. real-time system) able to regulate the attitude control sampling rate so that it can achieve appropriate balance between that and resource availability for other tasks such as science data handling.

4.1.1 Equations of Motion

The equations of motion for attitude control of a CubeSat can be developed using Euler equations for rigid body kinematics and dynamics with a diagonal inertia matrix **J**. The equations used in this chapter assume a circular orbit and small perturbations about the equilibrium point about which the equations of motion are linearized. The dynamics about the pitch axis are represented as

$$\begin{aligned} \dot{\theta}_2 &= \omega_2 \\ \dot{\omega}_2 &= \frac{3\omega_o^2(J_3 - J_1)}{J_2}\theta_2 + \frac{M_2}{J_2} \end{aligned} \tag{4.1}$$

where the body-fixed pitch axis is assumed to be aligned with one of the principal axes of the spacecraft. The torque applied (M_2) is equal to and opposite in direction to the rate of change of angular momentum of the microwheel $\left(\text{i.e. } \dot{H}_2^w = -M_2\right)$. The angular velocity for a circular orbit is $\omega_o = \sqrt{\mu/R^3}$ where μ is the gravitational constant and R is the radius of the orbit.

The dynamics about roll (1) axis and yaw (3) axis are represented by

$$\begin{aligned} \dot{\theta}_1 &= \omega_1 - \omega_o\theta_3 \\ \dot{\theta}_3 &= \omega_3 - \omega_o\theta_1 \\ \dot{\omega}_1 &= \frac{\omega_o(J_2 - J_3)}{J_1}\omega_3 + \frac{3\omega_o^2(J_3 - J_2)}{J_1}\theta_1 - \frac{\bar{H}_2^w}{J_1}\omega_3 + \frac{M_1}{J_1} \\ \dot{\omega}_3 &= \frac{\omega_o(J_1 - J_2)}{J_3}\omega_1 + \frac{\bar{H}_2^w}{J_3}\omega_1 + \frac{M_3}{J_3} \end{aligned} \tag{4.2}$$

where roll and yaw axes are assumed to be aligned with the principal axes of the spacecraft perpendicular to each other and perpendicular to the pitch axis. Note that the equations of motion are linearized about an equilibrium point where the body-fixed axes of the spacecraft are aligned with a Local Vertical Local Horizontal (LVLH) reference frame. Hence

$(\omega_1, \omega_2, \omega_3)$ are components of the perturbation about the equilibrium point in the angular velocity vector with respect to an inertial frame expressed in the body-fixed frame of reference. θ_1, θ_2, and θ_3 are perturbations of the 3-2-1 Euler angles that define the spacecraft attitude with respect to the LVLH coordinate frame. $\bar{H}_2^w$ represents bias in the pitch reaction microwheel. The torque applied (M_1, M_3) is equal to and opposite in direction to the rate of change of angular momentum of the microwheel $\left(\text{i.e. } \dot{H}_1^w = -M_1 \text{ and } \dot{H}_3^w = -M_3\right)$.

We can rewrite the open-loop equations in state-space form

$$\dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p \mathbf{u}_p$$

where the states and controls are

$$\begin{aligned} \mathbf{x}_p &= \left(\theta_1, \theta_2, \theta_3, \omega_1, \omega_2, \omega_3, H_1^w, H_2^w, H_3^w\right) \\ \mathbf{u}_p &= (M_1, M_2, M_3) \end{aligned} \tag{4.3}$$

and matrices $\mathbf{A}_p$ and $\mathbf{B}_p$ are taken from Equations (4.1) and (4.2). The CubeSat considered has mass of 3 kg with dimensions of $30\,\text{cm} \times 10\,\text{cm} \times 10\,\text{cm}$. The CubeSat has inertia matrix $\mathbf{J} = \text{diag}\,(0.005, 0.025, 0.025)\ \text{kg} \cdot \text{m}^2$. The altitude of the spacecraft is assumed to be 500 km above Earth's surface which results in an orbital angular velocity $\omega_o = 0.0011\,\text{rad/s}$. $\bar{H}_2^w = 0.01\,\text{kg} \cdot \text{m}^2$. Because this work also introduces a cyber system model we use the subscript "p" to indicate that these equations depict the physical system.

Depending on the configuration of the spacecraft the linearized system can either be stable or unstable [121]. For our CubeSat, the system matrix $\mathbf{A}_p$ has unstable poles thus it requires active control to stabilize.

4.2 Discrete CubeSat Model

As discussed in Section 2.2 there are several sources for uncertain delays when implementing a controller on an RTOS. Nevertheless, the traditional sampled-data assumption of no delay is reasonable to make under most scenarios. In a modern digital control system it is likely that dedicated A/D and D/A converters minimize conversion delays and we assume that a predictive algorithm can always provide the current physical system state at the moment the control output is calculated thereby eliminating appreciable delays in state estimation. This assumption allows us to leverage digital control theory to discretize the CubeSat model and design digital controllers.

4.2.1 Discrete CubeSat Model

If we assume the control task, τ_1, is a hard-deadline task and that execution deadlines are always satisfied by the real-time system, we can discretize the system for a given sampling period. In the most general case the discrete system matrices may vary due to parameter changes, uncertainty in dynamics, or in our case, a time-varying sampling rate. We reflect the discrete-time-varying nature of the system using the variable k, representing an execution cycle of the control task. Assuming a ZOH we can write the physical system as

$$\mathbf{x}_p(k+1) = \mathbf{\Phi}_p(k)\mathbf{x}_p(k) + \mathbf{\Gamma}_p(k)\mathbf{u}_p(k)$$

where

$$\begin{aligned} \mathbf{\Phi}_p(k) &= e^{\mathbf{A}_p T_{\tau_1}(k)} \\ \mathbf{\Gamma}_p(k) &= \int_0^{T_{\tau_1}(k)} e^{\mathbf{A}_p \eta} d\eta \mathbf{B}_p. \end{aligned} \tag{4.4}$$

We note that in traditional digital control theory a constant sampling period is assumed and the resulting system would be

$$\mathbf{x}_p(k+1) = \mathbf{\Phi}_p\mathbf{x}_p(k) + \mathbf{\Gamma}_p\mathbf{u}_p(k)$$

in which system matrices $\mathbf{\Phi}_p$, and $\mathbf{\Gamma}_p$ are constant over each cycle [60].

4.2.2 Physical System Control Laws

The design of feedback controllers for a system that can dynamically adjust its own sampling rate is a relatively new area for research [13, 100]. As a result we borrow from strong foundations in digital, optimal, and nonlinear control and seek to apply them to discrete-time-varying systems. We propose two controllers: a Gain-Scheduled Discrete Linear Quadratic Regulator (GSDLQR), and a Forward-Propagation Riccati-based (FPRB) controller.

4.2.2.1 Gain Scheduled DLQR Control

Infinite horizon Discrete Linear Quadratic Regulator (DLQR) controllers are designed assuming a fixed sampling rate and constant system matrices. For a given stabilizing sampling rate, because our system is completely controllable it is possible to compute an infinite horizon DLQR controller with a finite cost where the cost function is given by:

$$J = \frac{1}{2}\sum_{k=0}^{\infty} \mathbf{x}_p^{\mathrm{T}}(k)\,\mathbf{Q}\mathbf{x}_p(k) + \mathbf{u}_p^{\mathrm{T}}(k)\,\mathbf{R}\mathbf{u}_p(k). \tag{4.5}$$

The resulting optimal control law is given by:

$$\mathbf{u}_p(k) = -\mathbf{K}_p\mathbf{x}_p(k)$$

where

$$\mathbf{K}_p = \left(\mathbf{R} + \boldsymbol{\Gamma}_p^{\mathrm{T}}\mathbf{P}\boldsymbol{\Gamma}_p\right)^{-1}\boldsymbol{\Gamma}_p^{\mathrm{T}}\mathbf{P}\boldsymbol{\Phi}_p$$

and $\mathbf{P}$ is the positive definite solution to the Discrete-time Algebraic Riccati Equation (DARE)

$$\mathbf{P} = \mathbf{Q} + \boldsymbol{\Phi}_p^{\mathrm{T}}\left(\mathbf{P} - \mathbf{P}\boldsymbol{\Gamma}_p\left(\mathbf{R} + \boldsymbol{\Gamma}_p^{\mathrm{T}}\mathbf{P}\boldsymbol{\Gamma}_p\right)^{-1}\boldsymbol{\Gamma}_p^{\mathrm{T}}\mathbf{P}\right)\boldsymbol{\Phi}_p. \tag{4.6}$$

In the simulations carried out for our work, $\mathbf{Q} = 100\mathbf{I}_9$ and $\mathbf{R} = 10^5\mathbf{I}_3$ where $\mathbf{I}_n$ is the $n \times n$ identity matrix.

The choice of sampling rate is a difficult one and many of the factors affecting that choice are mentioned in Section 2.3.2. Research into the "Quality of Service" a RTS can provide offers insight into the tradeoffs between computational resources and system performance. Metrics quantifying control system performance under various sampling rates can be used to optimize computational resources under various conditions [109, 110]. Trends observed in QoS research suggest that increasing sampling rate yields improved control system performance [109]. However, this generality doesn't always hold as system characteristics such as harmonics and exponentials impact the response [122].

Consider the effect of sampling rate on the DLQR gains. Higher sampling rates generally result in larger gains while lower sampling rates generally result in lower gains [123]. While lower sampling rates may conserve energy, most often system robustness suffers as a result. For our CubeSat we specify upper and lower bounds for sampling rate. We choose a maximum sampling rate $r_{\tau,\max}$ for which we can guarantee that the control task is schedulable and a minimum sampling rate $r_{\tau,\min}$ for which we can still guarantee physical system stability:

$$r_{\tau_1,\max} = 10\,\mathrm{Hz}$$
$$r_{\tau_1,\min} = 0.1\,\mathrm{Hz}.$$

To illustrate the relationship between sampling rate and gain, we computed the matrix norm of DLQR gains for the CubeSat discretized at $r_{\tau_1,\max}$, $r_{\tau_1,\min}$, and an intermediate rate, $r_{\tau_1} = 1.0\,\mathrm{Hz}$ (see [124]). These are listed in Table 4.1.

Table 4.1: Scaling Factor Comparison for Normalized DLQR CubeSat Gains

SAMPLING RATE	$\|\mathbf{K}_p\|_2$
$r_{\tau_1,\max}$	0.0626
$r_{\tau_1} = 1\,\text{Hz}$	0.0325
$r_{\tau_1,\min}$	0.0050

We make the general assumption that sampling faster yields improved control system performance except for sampling rates where the H_2 cost is infinite which we disallow. In light of the widely varying gains between $r_{\tau_1,\min}$ and $r_{\tau_1,\max}$, and considering that some sampling rates will cause system instability, we followed the procedure in [122] to examine the relationship between sampling rate and control system performance. We swept the sampling rate from $r_{\tau_1,\min}$ to $r_{\tau_1,\max}$ and calculated the H_2 cost for each discrete closed-loop system using DLQR feedback control at each sampling rate. Results are shown in Figure 4.1. Triangles in the plot indicate points where $H_2 = \infty$ and those corresponding sampling rates are avoided in our simulation. Circles indicate sampling rates where a DLQR gain could not be found. For our CubeSat QoS increases with sampling rate commensurate with H_2 cost.

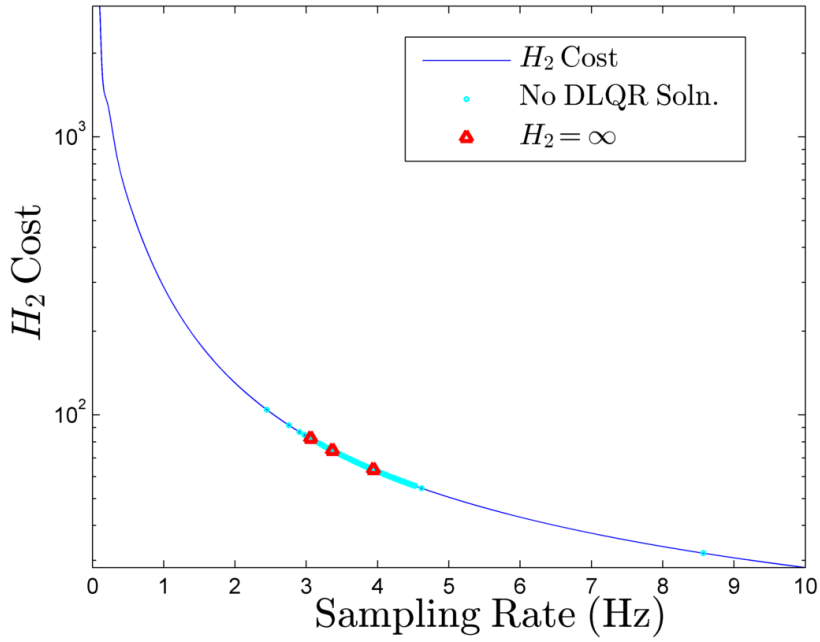


Figure 4.1: H_2 Performance Cost with Changing Sampling Rate

Because this work focuses on the dynamic adjustment of sampling rate, and since DLQR gains vary significantly over the range of possible rates, a constant DLQR gain will

yield suboptimal results. From the plot in Figure 4.1 it is also clear that we cannot compute a DLQR gain at all possible sampling rates between $r_{\tau_1,\min}$ and $r_{\tau_1,\max}$. Gain scheduling is a technique traditionally applied to nonlinear systems where the complexity of the nonlinear system prevents or greatly complicates the design of feasible controllers. In this paradigm, a nonlinear system is linearized about operating points or equilibrium points for which linear system control techniques can be applied. The effects of nonlinearities in the system are then mitigated by "scheduling"[2] the designed gains via an interpolating scheme to compute gains at intermediate operating points [125, 126].

We use this strategy as inspiration for developing a gain scheduling scheme over operating points of the cyber system (i.e. sampling rates). Where possible we design DLQR controllers for the CubeSat at discrete sampling rates between $r_{\tau_1,\min}$ and $r_{\tau_1,\max}$ where each sampling rate is an operating point of the CPS. We then "schedule" the appropriate DLQR gains for the CubeSat corresponding to the commanded sampling rate, $r_{\tau_1}(k)$, as illustrated in Figure 4.2. This paradigm ensures that an appropriate gain is used to compute

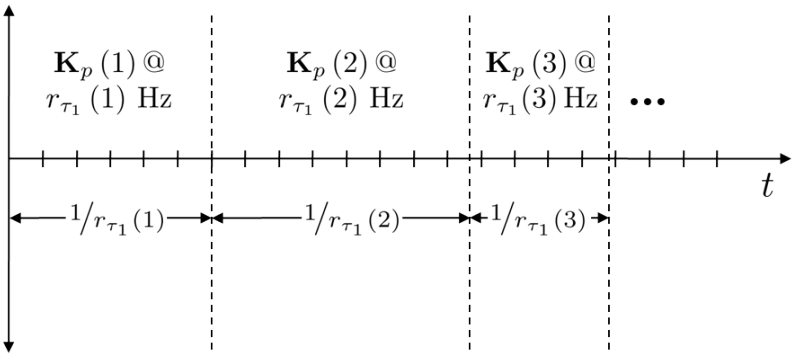


Figure 4.2: Gain Scheduling Over $r_{\tau_1}(k)$ (Sampling Rate)

the next control input for the newly-commanded control task sampling rate.

4.2.2.2 Forward-Propagation Riccati-Based (FPRB) Control

The optimal DLQR control is found by either propagating the DARE in Equation (4.6) backward from a final condition for finite-horizon control, or by finding the steady-state positive definite solution to the DARE for infinite-horizon control. Suppose we know system matrices $\mathbf{\Phi}_p(k)$, and $\mathbf{\Gamma}_p(k)$ $k = 1, 2, 3, \ldots, N$. We could then propagate the DARE in Equation (4.6) backward from a final condition to obtain the optimal discrete-time-varying

[2]We note that this form of "scheduling" is not the same as the scheduling discussed in Section 2.2 in the context of RTS.

control [127]. Since we don't know how the sampling rate will evolve (i.e. it is dynamically adjusted based on error in the physical system trajectory as described in Section 4.3) we do not know the system matrices in advance.

Forward-Integration Riccati-Based control is an emerging control design method wherein the solution to the forward-in-time control Riccati equation is used to compute the control gain. While research is still investigating the stability and performance guarantees of this method it has empirically shown to be effective in controlling a wide array of systems [116–119]. We apply this strategy to our discrete-time-varying CubeSat attitude control problem by computing

$$\mathbf{u}_p(k) = -\mathbf{K}_p(k)\,\mathbf{x}_p(k)$$

where

$$\mathbf{K}_p(k) = \left(\mathbf{R} + \boldsymbol{\Gamma}_p^{\mathrm{T}}(k)\,\mathbf{P}(k)\,\boldsymbol{\Gamma}_p(k)\right)^{-1}\boldsymbol{\Gamma}_p^{\mathrm{T}}(k)\,\mathbf{P}(k)\,\boldsymbol{\Phi}_p(k)$$

and $\mathbf{P}(k)$ is found iteratively using the *forward-in-discrete-time* algebraic Riccati equation

$$\begin{aligned}\mathbf{P}(k) =& \mathbf{Q} + \boldsymbol{\Phi}_p^{\mathrm{T}}(k)\Big(\mathbf{P}(k-1) - \mathbf{P}(k-1)\,\boldsymbol{\Gamma}_p(k)\\ &\left(\mathbf{R} + \boldsymbol{\Gamma}_p^{\mathrm{T}}(k)\,\mathbf{P}(k-1)\,\boldsymbol{\Gamma}_p(k)\right)^{-1}\boldsymbol{\Gamma}_p^{\mathrm{T}}(k)\,\mathbf{P}(k-1)\Big)\boldsymbol{\Phi}_p(k)\end{aligned}$$

with initial-time boundary condition $\mathbf{P}(0) \geq \mathbf{0}$. As before, in the simulations carried out for this work, $\mathbf{Q} = 100\mathbf{I}_9$ and $\mathbf{R} = 10^5\mathbf{I}_3$. As will be shown in Section 4.6 this controller is effective and only requires the forward-propagation of the DARE.

4.3 Cyber-Physical System Model

Having designed controllers for a discrete-time-varying CubeSat model we now present our state-variable cyber model, two cyber controllers, and couple this model to the state-variable CubeSat model via feedback control.

4.3.1 State-Variable Cyber Model

The RTOS on a CubeSat potentially executes many tasks related to guidance, navigation, control, science data collection, communication, etc. The ability to dynamically modify the schedule to allocate cyber resources is useful if some tasks can benefit from increased CPU time while others are scaled back. We assume that in our CubeSat study science data collection tasks or other tasks related to guidance, navigation, and CubeSat maintenance can always benefit from increased cyber resources. While it is also possible that the control task has multiple operating modes, branches, or algorithms we assume our control task

consists of linear feedback only and hence execution time from job to job remains approximately constant. With these assumptions in mind, wherein Chapter 3 we modeled two cyber states, one representing progress through the control task, and the other representing sampling rate, for our CubeSat case progress through the control task has no impact on design of a cyber or physical controller and we restrict our problem to modeling and regulating sampling rate or control task execution rate.

In a RTOS the scheduler runs periodically to order jobs in a priority queue. As such it also has the capability to nearly instantaneously (ignoring context switch time) modify the priority and sampling rate of the control task. For this work we assume that the sampling rate can be regulated any time the control task is not running or in an interrupted state (i.e. it has completed a cycle and has not started a new one). To apply state feedback we require a cyber model represented by an ordinary differential equation. This has the added benefit of providing "memory" or filtering. The cyber model of sampling rate is

$$\dot{x}_c = u_c$$

where x_c is the cyber state representing the frequency of the control task τ_1 $\left(\text{i.e. } x_c = r_{\tau_1} = 1/T_{\tau_1}\right)$, and u_c a forcing term adjusting the rate of change of the sampling rate. This implies that x_c has units $1/\text{s}$, or Hz, and u_c has units $1/\text{s}^2$.

4.3.2 Open-Loop Cyber-Physical System Model

We augment the continous-time physical system with our proposed cyber model forming the open-loop CPS equations

$$\begin{bmatrix} \dot{\mathbf{x}}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} \mathbf{A}_p & 0 \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ x_c \end{bmatrix} + \begin{bmatrix} \mathbf{B}_p & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_p \\ u_c \end{bmatrix}.$$

Since the cyber model will also be implemented on a digital computer we can apply the formula in Equation (4.4) to specify the CPS model as a set of difference equations:

$$\begin{bmatrix} \mathbf{x}_p(k+1) \\ x_c(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_p(k) & 0 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ x_c(k) \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma}_p(k) & 0 \\ 0 & T_{\tau_1}(k) \end{bmatrix} \begin{bmatrix} \mathbf{u}_p(k) \\ u_c(k) \end{bmatrix} \tag{4.7}$$

and note again that $x_c(k) = r_{\tau_1}(k) = 1/T_{\tau_1}(k)$. Because $T_{\tau_1}(k) = 1/x_c(k)$ and $\mathbf{\Phi}_p(k)$ and $\mathbf{\Gamma}_p(k)$ are functions of x_c (as per Equation (4.4)) the system is now nonlinear.

4.3.3 Cyber System Control Law

To design a control law for the new cyber model we must examine dependencies between the cyber and physical systems. In the closed-loop system, performance is directly dependent on the execution rate of the control task due to the ZOH nature of the RTOS implementation. System state, $\mathbf{x}_p$, is fed back into the cyber system from which we can compute the performance metric $\mathbf{x}_p - \mathbf{x}_{p,r}$ where $\mathbf{x}_{p,r}$ is the physical state reference trajectory. We want the cyber system to in turn adjust sampling rate based on the performance of the physical system.

As a result we design a two-part control law for the cyber system. One part reacts to off-nominal disturbance conditions in the physical system, and the other drives the task execution rate to a reference rate. We introduce two versions of the cyber control law for comparison in our results. In Version One, $u_{c,1}(k)$, we allow the control input to be scaled by the sampling period as indicated in Equation (4.7):

$$u_{c,1}(k) = \mathbf{K}_{cp}(k)\left(\mathbf{x}_p(k) - \mathbf{x}_{p,r}\right) - k_c\left(x_c(k) - x_{c,r}\right) \tag{4.8}$$

where $x_{c,r}$ is the cyber system reference trajectory (i.e. a desired sampling rate for τ_1), and k_c is a gain. For $u_{c,1}$, $\mathbf{K}_{cp}$ has units necessary to cancel physical state units

$$\mathbf{K}_{cp} = \left[\begin{array}{ccccccccc} 1/\mathrm{s}^2 & 1/\mathrm{s}^2 & 1/\mathrm{s}^2 & 1/\mathrm{s} & 1/\mathrm{s} & 1/\mathrm{s} & 1/\mathrm{N{\cdot}m{\cdot}s}^3 & 1/\mathrm{N{\cdot}m{\cdot}s}^3 & 1/\mathrm{N{\cdot}m{\cdot}s}^3 \end{array}\right]$$

and k_c has units $1/\mathrm{s}$. In Version Two, $u_{c,2}(k)$, we eliminate the scaling by $T_{\tau_1}(k)$ and the nonlinearity in the cyber system so that the cyber controller is unaffected by the current sampling rate. Therefore

$$u_{c,2}(k) = \frac{1}{T_{\tau_1}(k)}\left(\mathbf{K}_{cp}(k)\left(\mathbf{x}_p(k) - \mathbf{x}_{p,r}\right) - k_c\left(x_c(k) - x_{c,r}\right)\right). \tag{4.9}$$

For $u_{c,2}$ $\mathbf{K}_{cp}$ and k_c now have units

$$\mathbf{K}_{cp} = \left[\begin{array}{ccccccccc} 1/\mathrm{s} & 1/\mathrm{s} & 1/\mathrm{s} & \dim & \dim & \dim & 1/\mathrm{N{\cdot}m{\cdot}s}^2 & 1/\mathrm{N{\cdot}m{\cdot}s}^2 & 1/\mathrm{N{\cdot}m{\cdot}s}^2 \end{array}\right]$$
$$k_c = \dim$$

where dim indicates the quantity is dimensionless. Note that if there is nonzero error in the physical system the cyber system should *increase* the sampling rate. Therefore, $\mathbf{K}_{cp}$ is

specified as a gain vector with

$$\mathbf{K}_{cp}(k) = \begin{cases} k_{cp,i} & \text{if } x_{p,i}(k) - x_{p,i,r} \geq 0 \\ -k_{cp,i} & \text{if } x_{p,i}(k) - x_{p,i,r} < 0 \end{cases} \quad \forall k_{cp,i} \in \mathbf{K}_{cp},\, x_{p,i} \in \mathbf{x}_p,\, x_{p,i,r} \in \mathbf{x}_{p,r}.$$

This control law allows the cyber system to adjust its resources in accordance with the performance of the physical system as it simultaneously targets a reference execution rate. In practice it is likely a trajectory planner would update reference trajectories for both the physical and cyber system to meet mission and performance requirements.

4.3.4 Closed-Loop CPS Model

Now that we have discrete controllers for both the physical and cyber system we can write the closed-loop equations of the full CPS model using Equations (4.7), (4.8), and (4.9). Since we are regulating $\mathbf{x}_p$ to zero, $\mathbf{x}_{p,r} = \mathbf{0}$ and for $u_{c,1}$ we have

$$\begin{bmatrix} \mathbf{x}_p(k+1) \\ x_c(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_p(k) - \mathbf{\Gamma}_p(k)\mathbf{K}_p(k) & 0 \\ T_{\tau_1}(k)\mathbf{K}_{cp} & 1 - T_{\tau_1}(k)k_c \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ x_c(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ T_{\tau_1}(k)k_c x_{c,r} \end{bmatrix}.$$

For $u_{c,2}$ we have

$$\begin{bmatrix} \mathbf{x}_p(k+1) \\ x_c(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_p(k) - \mathbf{\Gamma}_p(k)\mathbf{K}_p(k) & 0 \\ \mathbf{K}_{cp} & 1 - k_c \end{bmatrix} \begin{bmatrix} \mathbf{x}_p(k) \\ x_c(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ k_c x_{c,r} \end{bmatrix}.$$

4.4 CPS Metrics

We demonstrate the effectiveness of our proposed methodology by analyzing and comparing simulation results against fixed-rate optimal control strategies. Measuring holistic CPS performance requires the development of additional metrics to evaluate more than traditional control performance indicators (e.g. rise time, settling time, etc.). In this section we describe our CPS evaluation metrics.

4.4.1 Performance Metrics

To appropriately compare results we utilize three metrics that collectively account for both physical and cyber performance. Each metric is described below.

4.4.1.1 Physical State Metric

To gauge the effectiveness of the control and rate of the control task on the physical system we examine the time-average squared error of physical state $\mathbf{x}_p$. Let $\mathbf{m}_p$ represent the metric for physical state, and let subscript j indicate the j^{th} entry in the state vector. Also let $x_{pj,r}$ be the reference trajectory for the j^{th} physical state. We then compute time-averaged physical state error as

$$\mathbf{m}_p = \begin{bmatrix} \frac{1}{t_\text{f}} \int_0^{t_\text{f}} \left(x_{p1}(t) - x_{p1,r}(t) \right)^2 dt \\ \vdots \\ \frac{1}{t_\text{f}} \int_0^{t_\text{f}} \left(x_{pj}(t) - x_{pj,r}(t) \right)^2 dt \end{bmatrix} \tag{4.10}$$

where t_f is the final time. This metric provides an assessment of how well CubeSat attitude and angular velocities are being regulated by the real-time control system. To facilitate comparison we also make use of a normalized physical state metric wherein we leverage the inherent discrete nature of the simulation to normalize the metric for each physical state

$$\mathbf{m}_{p,n} = \begin{bmatrix} \frac{1}{t_\text{f} x_{p1,\max}^2} \sum_{i=1}^{n} t_i \left(x_{p1,i} - x_{p1,r} \right)^2 \\ \vdots \\ \frac{1}{t_\text{f} x_{pj,\max}^2} \sum_{i=1}^{n} t_i \left(x_{pj,i} - x_{pj,r} \right)^2 \end{bmatrix} \tag{4.11}$$

where j is the j^{th} state and there are n discrete samples of the state.

4.4.1.2 Cyber Rate Metric

To measure computing system utilization/performance we focus attention on our regulation of the sampling rate. Although in a RTS many tasks would consume resources we assume that utilization of the control task is proportional to utilization of the total RTS. Lower utilization could result in reduced energy requirements for the RTS (e.g. with a voltage scaling CPU) or the liberation of resources that can be devoted to other tasks. For this metric we select a maximum sampling rate, $x_{c,\max} = r_{\tau_1,\max}$, under which the complete RTS remains schedulable (i.e. can meet all hard real-time task deadlines). We define our metric to be the time-averaged percent of maximum sampling rate:

$$m_c = \frac{1}{t_\text{f} x_{c,\max}} \sum_{i=1}^{n} t_i x_{c,i} \tag{4.12}$$

where n is the number of time slices from time $t \in [0, t_\text{f}]$. This metric was chosen over the traditional RTS utilization definition (as described in Section 4.5) because it allows

us to easily compare and analyze different controller designs independent of the RTOS implementation.

4.4.1.3 Control Effort Metric

An important measure of system performance is how much physical control effort is expended to meet performance requirements. This effort, a function of both sampling rate and control gain, requires energy expenditure for the CPS and therefore minimizing control effort can improve endurance and mission performance. An important consideration in the design of an energy-efficient control law is the sampling rate. Generally as sampling rate increases higher gain values can be tolerated while the system remains stable, while slower sampling rates require lower gains [68].

We are interested in minimizing control effort while maintaining closed-loop stability and trajectory tracking, captured in physical metric (4.11). It is common in optimal control to minimize $\mathbf{u}^{\mathrm{T}}\mathbf{u}$ as in the DLQR cost function in Equation (4.5). Because energy expenditure is generally a monotonically increasing function of control, minimizing control effort reduces energy expenditure. Our metric for control effort in this context only includes effort for the physical system, $\mathbf{u}_p$, given that we don't throttle CPU clock rate or turn cores on/off. Taking the DLQR cost term as a cue and due to the discrete nature of the control input caused by the ZOH, we define a control effort metric as the discrete time squared average of physical control effort

$$\mathbf{m}_{up} = \begin{bmatrix} \frac{1}{t_{\mathrm{f}}} \sum_{i=1}^{n} t_i u_{p1,i}^2 \\ \vdots \\ \frac{1}{t_{\mathrm{f}}} \sum_{i=1}^{n} t_i u_{pj,i}^2 \end{bmatrix} \tag{4.13}$$

where j is the j^{th} control input.

4.5 CubeSat Case Study

To develop a realistic case study of attitude control of a CubeSat we summarize the CubeSat literature with focus on simulating responses to disturbances. We then describe our CubeSat cyber model.

4.5.1 Physical Characteristics and Setup

Low-earth orbit presents a challenging environment due to the potential for plasma-induced and magnetic disturbances, high velocity debris and meteoroids, atmospheric drag, radiation, solar wind, and dust [128–132]. All are sources of disturbance on attitude and orbit of a CubeSat. Generally a CubeSat has three reasons to adjust its attitude: scientific data acquisition, communication with a ground station, or to maximize solar energy harvesting. Pointing activities must be planned and carried out within narrow time constraints and it is critical that controllers be capable of rejecting disturbances to achieve these goals. As discussed in Section 2.3.4.6, optimal control input and sampling pattern algorithms [13, 100] have been proposed to schedule controller sampling rate and conserve computing resources; however, these algorithms do not attempt to deal with disturbances which are more effectively handled by feedback control [13]. In this chapter we have proposed such a CPS feedback control formulation and therefore focus on highlighting its ability to deal with disturbances.

Our tests generate system responses to initial conditions representing an impulsive disturbance due to an impact or other transient event that perturbs the attitude and corresponding angular rates of the CubeSat. The controller objective is then to restore both attitude and angular rates to a zero reference state. The initial conditions on the physical state representing this disturbance are defined:

$$\mathbf{x}_{p0} = \begin{bmatrix} 0.1 & 0.5 & 0.2 & 0.02 & 0.01 & 0.005 & 0 & 0 & 0 \end{bmatrix}$$

where states (1, 2, 3) are roll, pitch, and yaw in the LVLH reference frame, states (4, 5, 6) are elements of the angular velocity vector, and states (7, 8, 9) represent angular momentum of each of three reaction microwheels used in control. Because we are regulating states to zero the reference trajectory is

$$\mathbf{x}_{p,r} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In a 500 km orbit altitude ([133, 134]) above Earth's surface our simulated CubeSat spends roughly 62% of its orbit (59 min) in sunlight during which energy is collected via solar panels, producing about 7 W of power, and stored in a 7.4 V, 4.4 Ah LiOn battery [135, 136]. While it is possible to store energy in the microwheels ([137, 138]) we assume they are used strictly for attitude control and that energy for control of the microwheels is only delivered from the battery system [136]. We also assume that the solar energy harvesting is sufficient during each orbital period to replenish the energy expended during

eclipse. We use one reaction wheel for each axis of rotation which has characteristics (similar to [139–141]) shown in Table 4.2.

Table 4.2: Reaction Microwheel Characteristics

CHARACTERISTIC	VALUE
Max Torque	$30\,\mathrm{mN \cdot m}$
Supply Power	$7.0\,\mathrm{W}$ @ $6500\,\mathrm{RPM}$, $5\,\mathrm{mN \cdot m}$
Wheel Inertia	$0.001766969\,\mathrm{kg \cdot m^2}$
Mass	$500\,\mathrm{g}$

4.5.2 Cyber Characteristics and Setup

Current trajectories of CubeSat development suggest that the time will come when the majority of computationally-intense tasks onboard a CubeSat will be those associated with autonomous decision making and science data handling [142–144]. However, at present, guidance, navigation, and control (GNC) tasks still consume a nontrivial portion of cyber resources. With this in mind we posit that significant savings can be realized by adjusting GNC tasks in accordance with pointing performance.

We assume the computing platform onboard the CubeSat is running a RTOS capable of dynamically adjusting the period of the control task as long as the control task isn't running or in an interrupted state. As discussed in Section 4.2.2 we set hard limits on the cyber rate based on the maximum schedulability for the control task and the performance requirements of the CubeSat. For our particular system we choose

$$\begin{aligned} x_{c,\max} &= r_{\tau_1,\max} = 10\,\mathrm{Hz} \\ x_{c,\min} &= r_{\tau_1,\min} = 0.1\,\mathrm{Hz}. \end{aligned}$$

Such hard limits are similar to saturation limits on actuators and are treated as such. RTS utilization is defined as

$$U_{\mathrm{RTS}} = \sum_{i=1}^{n} \mathrm{WCET}(\tau_i)\, r(\tau_i)$$

where $\mathrm{WCET}(\tau_i)$ is the worst-case execution time of τ_i, $r(\tau_i)$ is the rate of task τ_i, and n is the number of tasks [58]. In popular preemptive scheduling paradigms such as Rate-Monotonic (RMS) or Earliest Deadline First (EDF) scheduling, $U_{\mathrm{RTS}} \leq 1$ implies a valid schedule such that all deadlines will be met [58]. Recalling that τ_1 is the attitude control

task, we assume that without τ_1, $U_{\text{RTS}} = 0.70$ and that $\text{WCET}(\tau_1) = 0.03\,\text{s}$. Therefore

$$U_{\text{RTS}}(x_{c,\max}\,\text{Hz}) = U_{\text{RTS}} + 0.03x_{c,\max} = 1$$
$$U_{\text{RTS}}(x_{c,\min}\,\text{Hz}) = U_{\text{RTS}} + 0.03x_{c,\min} = 0.703$$

which implies a significant reduction in cyber resource utilization when we reduce the sampling rate. The selection of the cyber reference rate could be based on physical system characteristics such as the Nyquist rate, disturbance characteristics, harmonics, exponentials as well as cyber characteristics such as maximum schedulable rate, QoS tradeoffs, etc. We assume that a higher level trajectory planner (such as that described in Chapter 5) would provide the cyber reference rate. In this work through testing it was determined that

$$x_{c0} = 0.3\,\text{Hz}$$
$$x_{c,r} = 0.3\,\text{Hz}$$

clearly illustrate the differences in controller behavior and demonstrate good cyber resource reclamation. With the 0.3 Hz reference rate, $U_{\text{RTS}}(0.3\,\text{Hz}) = 0.709$, resulting in a 29.1% cyber resource utilization savings relative to the maximum rate.

$\mathbf{K}_{cp}$ was determined by manual tuning as

$$\mathbf{K}_{cp} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Similarly, the control gain of the cyber system was tuned to

$$k_c = 0.5.$$

Our simulation is executed over a 20 second interval which is sufficient in our case to observe disturbance rejection behavior.

4.6 CubeSat CPS Simulation Results

We illustrate the utility of our variable-rate control laws by comparing them with fixed-rate DLQR controllers and with each other in our CubeSat case study. We first offer some specifics of our MATLAB simulation. In the results we use as baseline designs DLQR controllers designed at fixed sampling rates $r_{\tau_1,\max}$, $r_{\tau_1,\min}$, and $r_{\tau_1} = 1\,\text{Hz}$. We first compare time response plots of GSDLQR control against a fixed 1 Hz DLQR control design. We then compare time response plots of FPRB control against GSDLQR control and fixed 1 Hz

DLQR control. Finally, to compare all designs we use the evaluation metrics presented in Section (4.4.1) and tabulate the results.

4.6.1 Simulation

MATLAB offers two primary methods of control system simulation, continuous and discrete. In the case of continuous time systems, ordinary differential equation solvers such as `ode45` can be used to simulate linear and nonlinear system response to initial values. Specifically aimed at control design for both discrete and continuous linear systems `lsim` provides the system response to a user defined control input. All of MATLAB's simulation techniques assume either a purely continuous system or a discrete system executed at a single sampling rate. Our proposed technique, however, requires a mechanism for simulating a system with a time-varying sampling rate.

To manage this difficulty we use a fourth-order Runge-Kutta variable time step ordinary differential equations solver, namely MATLAB's `ode45`, to solve each time-varying discrete step of the simulation. At each discrete step (integration cycle) of the simulation the "initial condition" is the final state from the previous integration cycle, and the control input is held constant during that cycle. As the control loop execution rate, x_c, changes according to the cyber system dynamics the length of an integration cycle changes. Because MATLAB's `ode45` is a one-step solver, we can piece together the output from multiple executions of `ode45` based only upon the "initial conditions," $\mathbf{x}_{p,prev}$, as shown in Algorithm 4.1. We've chosen a highly-accurate integrator to enable us to look into the true system response including "ripple" or transients between discrete (sample-and-hold) cycles [60, 145].

4.6.2 GSDLQR CPS Designs

GSDLQR control was applied to the CubeSat CPS as discussed in Section 4.2.2.1 and simulated with initial state disturbance-induced error specified in Section4.5. In Figure 4.3 we show the response of states $\theta_{p,1}$ (roll angle), and $\omega_{p,1}$ (angular velocity in roll direction), the physical control for roll, $u_{p,1}$, and the cyber state x_c. In Figure 4.3a cyber controller $u_{c,1}$ (Equation (4.8)) is used and in Figure 4.3b $u_{c,2}$ (Equation (4.9)) is used.[3]

Recall that the state x_c is the sampling rate of the system for the next time step. Because $x_{c0} = 0.3\,\text{Hz}$ the system does nothing for $T_{\tau_1}(0) = 3.\bar{3}\,\text{s}$ while waiting for the next update to observe the error in the physical states. At time $t = 3.\bar{3}\,\text{s}$ the controller executes and computes a new sampling rate that is higher due to the large physical state error. As $\mathbf{x}_p$

[3] Comprehensive plots for all CubeSat simulations can be found in Appendix B

Algorithm 4.1 Algorithm for Simulation of CPS

Initialize variables
while $t < t_{\text{final}}$ **do**
  % Propagate the cyber system
  $x_c = x_c + T_{\tau_1} u_c$
  tspan = $[t, t + 1/x_c]$

  % Propagate the physical system
  $\mathbf{K}_p$ = computeKp$\left(t, \mathbf{x}_{p,prev}, x_c\right)$ % either Gain scheduled or FPRB control
  [t,x_p] = ode45$\left(\text{@CPSmodel(),tspan}, \mathbf{x}_{p,prev}\right)$
  $\mathbf{x}_{p,prev}$ = x_p(end,:)

  % Collect the states and inputs
end while

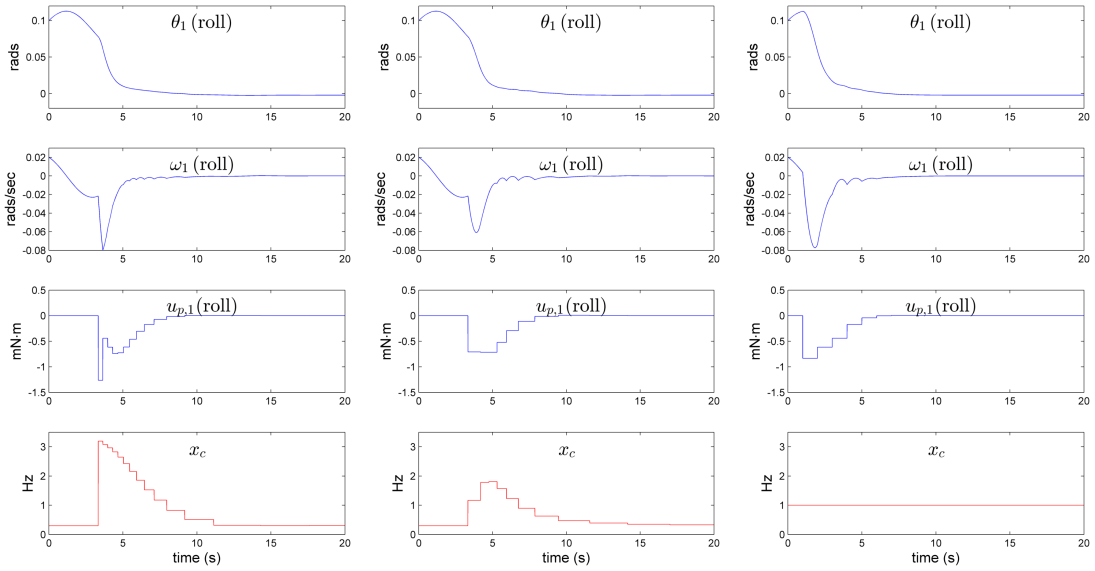
approaches zero, the reference value, the cyber controller begins to push the sampling rate down to $x_{c,r}$.

There are minor differences between using cyber controllers $u_{c,1}$ and $u_{c,2}$ as seen in Figure 4.3. In the equations for u_c (Equations (4.8) and (4.9)) there is balance between the errors in the physical states forcing x_c high and the error in the cyber state forcing it low. That balance is scaled by $T_{\tau_1} = 1/x_c$ as seen in Equation (4.7). Hence when x_c is high, u_c is less forceful thereby attenuating that balance, and when x_c is low (e.g. < 1) that balance is magnified. This effect is seen in the more gradual slopes of x_c both ramping up and ramping down in Figure 4.3b which has the added benefit of resulting in lower control effort and cyber resource utilization while providing similar physical system performance.

4.6.3 FPRB CPS Designs

We now select $u_{c,1}$ as the controller for the cyber system and show comparisons of our FPRB design from Section 4.2.2.2 with the the GSDLQR controller also using $u_{c,1}$. In Figure 4.4 we show time response plots for the same states and control $\left(\theta_1, \omega_1, u_{p,1}, x_c\right)$. In Figure 4.4a we show FPRB control using $u_{c,1}$ and in Figure 4.4b GSDLQR control using $u_{c,1}$. We then show the fixed-rate DLQR at 1 Hz in Subfigure 4.4c for reference.

Consider the physical control effort $\left(u_{p,1}\right)$ applied by FPRB and GSDLQR control. Despite having nearly identical physical and cyber state trajectories the control effort for GSDLQR spikes very low initially, and only subsequently follows a trajectory similar to that of FPRB. The FPRB controllers generally exert much less control effort on the physical system for nearly identical responses in physical and cyber states than the DLQR controllers,



(a) GSDLQR using $u_{c,1}$ (b) GSDLQR using $u_{c,2}$ (c) Traditional DLQR Control at 1 Hz

Figure 4.3: Gain Scheduled DLQR CPS Comparisons

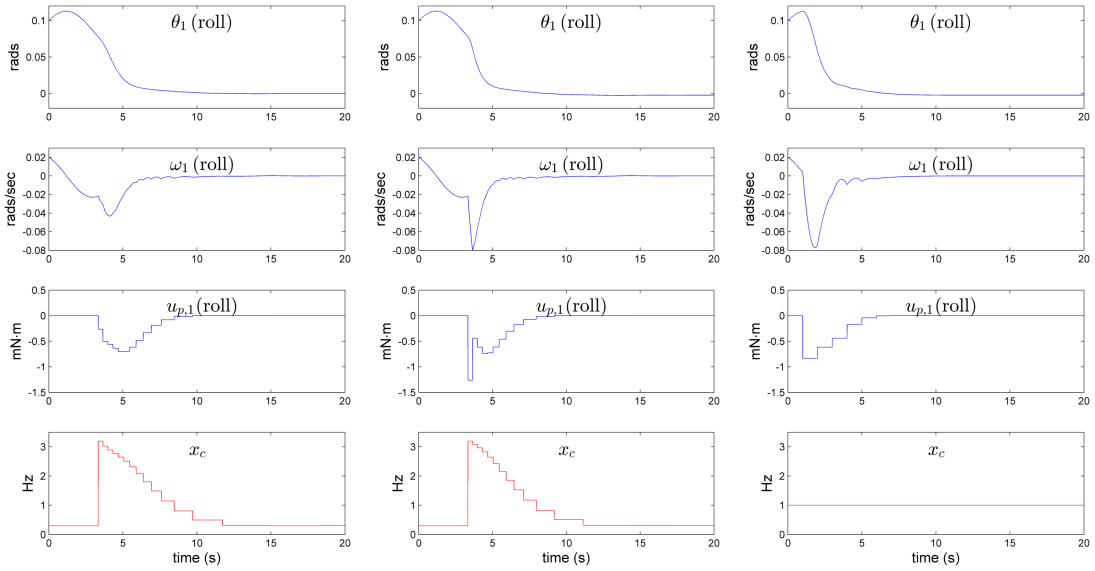
suggesting FPRB out-performs GSDLQR and fixed-rate (1 Hz) DLQR control.

4.6.4 Design Comparisons

In this section the metrics presented in Section 4.4.1 are used to evaluate the effectiveness of all presented controller designs. We investigate three baseline DLQR controllers at $r_{\tau_1,\max}$, $r_{\tau_1} = 1$ Hz, and $r_{\tau_1,\min}$ and simulate them in the traditional manner using the chosen sampling rate. The first baseline design, $r_{\tau_1,\max}$, represents a system design wherein CubeSat pointing performance is most valued and real-time system bandwidth is plentiful. The design assuming $r_{\tau_1,\min}$ represents the opposite extreme where cyber resources are scarce and more highly valued than attitude pointing accuracy. This may be appropriate where cyber resources are prioritized to favor tasks such as communication, or science data collection. Finally, we choose $r_{\tau_1} = 1$ Hz as a compromise between these two extremes.

In Table 4.3 we show a comparison of the different designs using our metrics. Table 4.3 reveals some important tradeoffs between control strategies. The DLQR fixed-rate controller at 1 Hz controls the physical states very well while using reasonable physical and cyber control effort. GSDLQR controllers offer a significant savings in cyber effort but result in higher error in physical state trajectories and a very large amount of physical control effort cost (see column 4) even exceeding the fixed rate 10 Hz controller.

The FPRB controllers show promise in balancing cyber and physical cost metrics via



(a) FPRB using $u_{c,1}$ (b) GSDLQR using $u_{c,1}$ (c) Traditional DLQR Control at 1 Hz

Figure 4.4: FPRB CPS Comparisons

online rather than *a priori* specification. On the cyber side, FPRB CPS using $u_{c,2}$ (i.e. the last row in Table 4.3), when compared with the maximum fixed-rate 10 Hz controller, achieves slightly poorer physical control, most of the error of which occurs in the transient portion during time $t = \left[0, 3.3\bar{3}\right]$ before the controller responds. However, at that expense it achieves significantly lower cyber resource utilization. In fact, RTS utilization goes from $U_{\text{RTS}}(10\,\text{Hz}) = 1$ to $U_{\text{RTS}}(0.613\,\text{Hz}) = 0.718$, a 28.2% savings in RTS cyber resource utilization.

On the physical side, as seen in column four of Table 4.3, the FPRB controllers use significantly less control effort over our 20 s simulation than all but the lowest effort controller (DLQR@0.1 Hz). If we assume a constant power bias to operate the electronics, the mechanical power of each wheel is

$$P_i = \Omega_i u_{p,i}$$

where Ω_i is the angular speed of the i^{th} wheel [146]. The total mechanical power for all wheels is [146]

$$P_{\text{total}} = |P_1| + |P_2| + |P_3|.$$

FPRB CPS using $u_{c,1}$ (i.e. the sixth row in Table 4.3) gives us 12.2% savings in average total power, and a 44.1% savings in peak power compared with the fixed-rate DLQR 10 Hz

Table 4.3: Comparison of CPS Control Designs

DESIGN	$\frac{\|\|\mathbf{m}_{p,n}\|\|}{\|\|\mathbf{m}_{p,n}\|\|_{\min}}$	$\frac{\|\|\mathbf{m}_{p}\|\|_{\infty}}{\|\|\mathbf{m}_{p}\|\|_{\infty,\min}}$	$\frac{\|\|\mathbf{m}_{up}\|\|}{\|\|\mathbf{m}_{up}\|\|_{\min}}$	m_c
DLQR@10Hz	1.0000	1.0000	71.9176	1.0000
DLQR@1Hz	1.4253	1.9743	58.0076	0.1000
DLQR@0.1Hz	7.9076	18.1157	1.0000	0.0100
GSDLQR CPS using $u_{c,1}$	2.4131	4.0907	78.8988	0.0786
GSDLQR CPS using $u_{c,2}$	2.4563	4.1937	67.2715	0.0599
FPRB CPS using $u_{c,1}$	2.7085	4.5546	32.7810	0.0824
FPRB CPS using $u_{c,2}$	2.5723	4.3232	47.3717	0.0613

controller.[4]

4.7 Conclusions

Research in cyber-physical systems (CPS) demands creative approaches to develop new models and abstractions to couple interacting cyber and physical control strategies. To this end we propose an abstraction to couple CPS control that builds upon linear state-space feedback control. The physical dynamics state-space model is augmented with an abstracted model of the cyber system, and a control formulation is proposed to dynamically regulate cyber resources based on physical state error. We have applied our co-regulation approach to attitude control of a small satellite system (CubeSat) and conducted a disturbance-rejection case study based on that platform.

Our CPS controller enables the cyber system, specifically the attitude controller, to operate at a lower sampling rate than might otherwise be chosen based on a single worst-case condition yet still retaining robustness to disturbances. This strategy can free cyber resources thereby allowing the cyber system to reallocate resources to other tasks, or to conserve energy by reducing processor clock speed or turning off cores. We have also devised baseline GSDLQR and FPRB control law formulations, proposed evaluation metrics, and investigated the performance of the controllers in simulation. Results indicate that FPRB formulations can indeed dynamically balance cyber and physical resource use via our co-regulation scheme.

While this representation makes progress toward a holistic CPS representation for co-regulation, there are important issues requiring further investigation. In this work we did not provide a formal optimization scheme to determine the best values for the gains $\mathbf{K}_{cp}$ or

[4]Comprehensive plots for all CubeSat simulations can be found in Appendix B

k_c. Future work is also needed to explore alternative performance metrics, domain models, and disturbances to provide additional insight into the tradeoffs between GSDLQR, FPRB, and fixed-rate digital control. Additionally, a critical component for future use of this proposed system will be establishing formal stability guarantees for the CPS. Finally, our results and proposed system would be strengthened by experimental verification in a real CubeSat or similarly complex robotic platform.

CHAPTER 5

Cyber-Physical Optimization for Unmanned Aircraft Systems

CPS co-design requires us to consider computation, communication, and control at all layers of system integration. In Chapters 3 and 4 we looked at computational control co-design at the feedback control level. We now move to a higher level of reasoning - planning - and propose a new co-optimization scheme of computational and physical resources to maximize mission success for an Unmanned Aircraft System (UAS).

In this chapter, using UAS pipeline inspection as an example, a mission-appropriate analytical cost function is developed to provide a minimal-cost trajectory over the mission. We simplify the cost function by allowing design variables to remain static throughout the mission, consistent with a steady flight scenario, thereby reducing the complexity of the cost function and optimization process. We then examine Pareto fronts for combinations of cost function objectives to demonstrate the important tradeoffs between physical and cyber resources and to give insight into the interdependence between them. We use a numerical solver to find physical-subsystem optimal, cyber-subsystem optimal, and holistic-system optimal solutions and compare them with solutions selected from Pareto front analysis. We demonstrate that only via a total Cyber-Physical System optimization can one achieve efficiency throughout the total system.

For our case study, we adopt a solar-supplemented powered-glider small UAS currently flown by a University of Michigan student team (SolarDrones) for which steady flight performance parameters are available. The small UAS payload is a downward-facing video camera that can provide frames at a variable rate. The simulated avionics allows direct regulation of computational power requirements in a manner that trades energy use with camera data acquisition bandwidth. A one-dimensional pipeline inspection case study is investigated, focusing attention on physical and computational energy use tradeoffs without the additional complexity of optimizing an accelerated path through three-dimensional space.

5.1 Cost Functions

We seek optimization over both physical and cyber characteristics of the UAS and its mission to more holistically optimize system performance. This means developing cost terms for task performance and energy required for cyber activities as well as for control actuation effort and propulsion. Moreover, we want to maximize efficiency of our designated mission which will include goals for both the physical and cyber components of the UAS. These mission-dependent goals may include maximizing coverage area or amount of information acquired for a given area, along with collection, processing and transmission of data.

In this work we develop cost terms representing both mission goals and efficiency for a UAS whose mission is surveillance of a straight section of pipeline with a small, lightweight downward-facing gimballed camera. We focus on movement in one dimension only and assume flight dynamics are governed by steady flight assumptions. The metrics proposed in this chapter complement the metrics in Section 4.4.1 but are specific to our proposed UAS mission.

Our objective for the physical system will be to determine the optimal velocity (airspeed in one dimension) of the UAS for the mission. Owing to the assumptions of the mission and steady flight we rely on a gimbal to consistently adjust the camera to point directly toward the ground (optical axis perpendicular to ground plane) which compensates for changes in pitch of the aircraft needed to accommodate various speeds of flight.

We model a single real-time task to accomplish the primary goals of the mission related to pipeline surveillance. This task performs image acquisition, processing, and communication/storage of image. Our design objective for the cyber system will be to determine the optimal execution rate of this task. While there are other system-critical tasks on the cyber system including the control task, we assume these require a fixed amount of resources. We instead focus on optimizing over the remaining non-critical bandwidth available in the cyber system.

We divide the cost terms into physical and cyber goals for clarity, and emphasize the assimilation of each into a system-wide cost function. “Physical” in the context of a UAS includes items related to flight, for example, the airframe, propulsion system, and control surfaces. “Cyber” relates to items required for data processing, communication, image collection, computation of control inputs, etc. In this work we endeavor to focus clearly on the idea of combining physical and cyber cost terms into a holistic cyber-physical system cost function.

5.1.1 Physical System Terms

Small UAS typically have modest energy reserves, most often consisting of small battery packs or a small fuel tank. In non-energy-harvesting applications under normal conditions such energy supplies can provide a small fixed-wing UAS between thirty minutes to a few hours of flight time. These flight times can be reduced when cyber-intensive activities such as image processing and communication are involved. Minimizing energy consumption over the mission is an important consideration in the design and control of the UAS.

5.1.1.1 Physical System Energy

In most aircraft applications, propulsion will consume the majority of the energy required for flight, surpassing actuation effort required by control surfaces. For simplicity, in this work we assume propulsion is the only drain on energy supplies by the physical system, that net energy is always a loss, and we model steady flight in which power used by control surface servos would be constant or near-constant. We therefore seek to minimize energy of the physical system over the entire mission

$$E = \int_0^T P(v(t))\,dt \tag{5.1}$$

where T is the duration of the mission and $P(v(t))$ is a traditional model for power as a function of velocity [147]

$$P(v(t)) = \frac{1}{2} S C_{D_0} \rho v(t)^3 + \frac{2KW^2}{\rho v(t) S}. \tag{5.2}$$

In steady level flight, power of the aircraft, and therefore its velocity, is manipulated by a throttle setting that maps nonlinearly to power as

$$P = \eta \delta_t \left(\frac{\rho}{\rho^s}\right)^m P^s_{\max} \tag{5.3}$$

where $P^s_{\max}$ is the maximum power of the engine/motor at sea level, $m > 0$ is a characteristic of the engine/motor, $0 \leq \eta \leq 1$ is a propeller efficiency factor, ρ^s is the air density at sea level, and $0 \leq \delta_t \leq 1$ is the throttle setting [147]. The power curve for our UAS (described in Section 5.2.4.1 on page 83) can be seen in Figure 5.1.

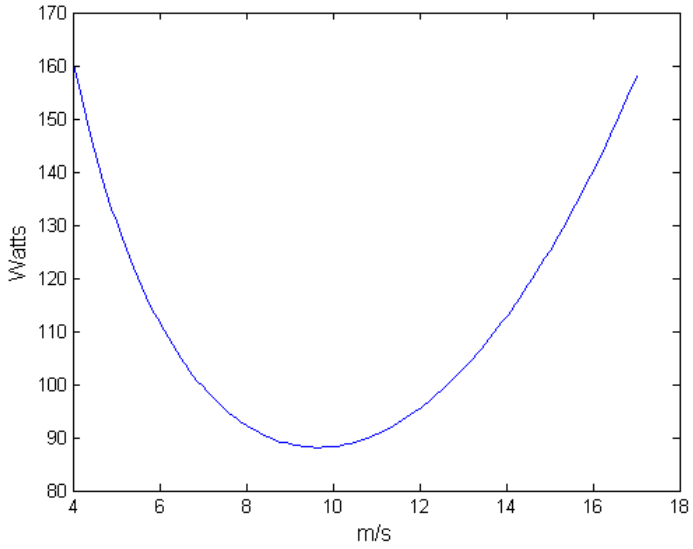


Figure 5.1: Power Curve for SolarDrones UAS

5.1.1.2 Time

In addition to minimizing energy, ideally we would like to efficiently accomplish our mission by minimizing the time required to complete it. Such minimum-time optimization cost terms appear frequently in traditional optimization schemes and are given by

$$T = \int_{t_0}^{t_1} dt \tag{5.4}$$

where $[t_0, t_1]$ =the total mission surveillance time.

5.1.1.3 Cost Function for Physical System

These two competing objectives, E_p and T, comprise the cost function for the overall physical system

$$J_p = \beta_{p1} \int_0^T P(v(t))\,dt + \beta_{p2} T \tag{5.5}$$

where β_{p1}, β_{p2} are weighting terms. Optimizing J_p alone is what a traditional trajectory or path planner might do if no costs are attributed to the cyber system. While some UAS researchers have added tracking information, target acquisition, and other mission objectives to their control and optimization algorithms [148, 149], to our knowledge this has historically been done from the physical perspective without attempting to optimize over cyber system performance and requirements.

5.1.2 Cyber System Terms

In a modern fully-autonomous UAS the cyber system becomes the gateway for virtually all aspects of the system. Control actuation inputs, data collection, communication, throttle setting, path and mission planning are potentially all being done simultaneously on-board. While real-time system researchers have advanced scheduling techniques for prioritizing each of these critical tasks, the correlation between physical performance, mission objectives, and computational efficiency has remained largely unexplored [16].

In many cyber-physical systems (CPS) task execution rates are selected *a priori* based on requirements of the system. For example, the sampling rate of the control task may be selected based on digital control analysis thereby ensuring robustness and stability margins. While it is unreasonable to interfere with such high priority tasks, lower priority tasks may still have some flexibility in task execution rate allowing us to optimize over mission and cyber parameters without interfering with mission critical tasks.

In our previous work we explored the tradeoff of mission critical task execution rates and physical performance [26, 115]. For this work we assume that hard real-time feedback control tasks are appropriately scheduled and executed while we focus on the rest of the available cyber resources for soft real-time tasks. More specifically, we assume that we can not only conserve energy by optimally selecting execution rates of lower priority tasks, but we can also increase mission effectiveness by developing costs that relate task execution rates to mission efficiency.

5.1.2.1 Cyber Utilization

In real-time system scheduling theory online schedules can be created by examining relative deadlines of independent periodic tasks as in the earliest deadline first scheduling algorithm. Such optimal scheduling algorithms are dynamic in that they can assign task priority as jobs are released to the operating system for scheduling [58]. They therefore have the ability to respond to changing deadlines and periodic rates.

In this work we assume that at least part of the cyber utilization is fixed based on selected and scheduled periodicity of mission critical tasks, consistent with current practices in the Aerospace community. We then focus on maximizing use of the remaining resources. The innovation of this work relative to applications studied by others (e.g. CASPER for EO-1 [102]) is that we purposely co-optimize the speed of the aircraft and the speed of payload (image) data acquisition/processing. Because aircraft motion changes what is observed in the acquired image sequence, and because energy is consumed by flight control and payload systems, this co-optimization is essential for a more "globally-optimal" solu-

tion than is possible when separately optimizing cyber and physical resource usage.

In this initial co-optimization work, we assume a single task, τ_2, is repeatedly executed to achieve the mission goal of capturing and processing images of a pipeline to be inspected. The task runs at execution rate $r_{\tau_2}(k)$ Hz and has a maximum execution rate of $r_{\tau_2,max}$ Hz stemming from restrictions based on available cyber resources and the worst case execution time (WCET) of task τ_2. That is, we ensure schedulability of the task based on $r_{\tau_2,\max}$ but allow that rate to slow down resulting in freed cyber resources to be devoted to increased service of other processing tasks or to conserving energy through fewer memory cycles, reduced processor clock rate when possible, and/or shutdown of some system cores when possible. Let k represent the execution cycle of τ_2, incrementing each time task τ_2 is run. The execution rate of τ_2 at cycle k is then $r_{\tau_2}(k)$. Figure 5.2 demonstrates an example processor utilization timeline depicting $r_{\tau_2}(k)$. We then introduce the cyber utilization term

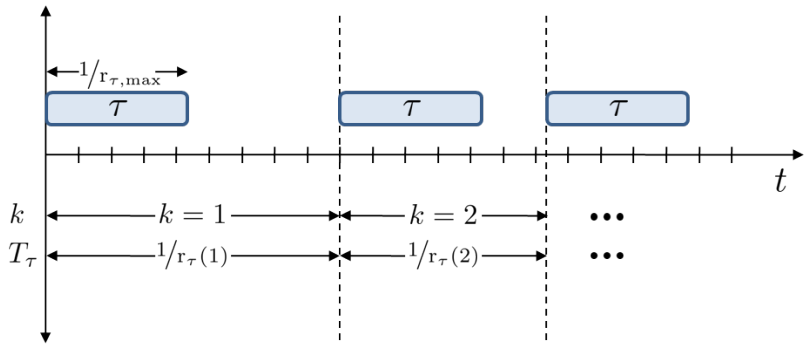


Figure 5.2: Processor Utilization Timeline for Task τ

as the weighted average of the ratio of task rate to max task rate

$$U_{\tau_2} = \frac{1}{T r_{\tau_2,\max}} \sum_k t_k r_{\tau_2}(k). \tag{5.6}$$

Note that the rate of execution cannot change during a particular execution cycle of that task. We assume that cyber utilization is proportional to energy consumed by the cyber system, and as a result, minimizing it is the cyber equivalent to the energy minimization term of the physical system in Equation (5.1). This metric is identical to the metric used to quantify RTS performance in our co-regulation work in Equation (4.12).

5.1.2.2 Mission Information

We seek to relate mission efficiency to cyber and physical parameters. For our specified mission, we assume that detailed imagery of the pipeline is critical for detecting aberra-

tions and problems. Collecting an appropriate amount of imagery, and more specifically appropriate imagery of any pipeline cracks, is critical for detection of problems. We approach this goal from an information theory viewpoint and desire to create a cost term in that context.

Information theory was originally designed to be applied to source coding where the limits of data compression are lower bounded by the entropy of the source code [150, 151]. However, it has since been applied to numerous fields and in numerous creative ways including image processing, object detection, and surveillance. In image processing and computer vision, mutual information can be used to provide image registration in medical imaging [152], while maximum entropy is regularly used in image reconstruction particularly in astronomy [153]. In the context of UAS surveillance systems information theory has been used for dim target detection in sense-and-avoid applications [154].

The most common quantity of information is entropy of a random variable. Let X be an ensemble $\{x, \mathcal{A}_x, P_x\}$ where x are the outcomes, $\mathcal{A}_x$ is the domain or sample space, and P_x the set of outcome probabilities. Then the entropy for a discrete ensemble is defined as

$$H(X) = -\sum_{x \in \mathcal{A}_x} p(x) \log_2 p(x). \tag{5.7}$$

Intuitively, entropy measures the average unpredictability of a random variable.

Information theory is used in this work to develop an appropriate mission information cost term. To ensure a well-behaved total system cost function each individual cost function must be continuous, convex, and non-negative such that, when minimized, greater benefits are realized. This means we need a metric that, when minimized, produces increased information about the pipeline being imaged.

For surveillance missions in which detection of aberrations or events are important, multiple observations of any single point in the area of interest are valuable. Acquiring multiple images of the same ground points has the advantage of providing additional viewpoints and redundant data, and may allow for super-resolved imagery thereby increasing our ability to detect pipeline anomalies [155]. Keeping this in mind, we propose a cost term based on overlap between successive images where increasing overlap is rewarded. Increased overlap between images is equivalent to lower entropy as the scene in each successive image changes little. Additionally, if there is underlap each successive image is completely new information about the pipeline and therefore provides the maximum entropy. From this perspective, contrary to traditional applications of information theory, we seek to minimize entropy as this strategy provides the most redundant information.

From an information theory perspective we view information cost as an exponential dis-

tribution of redundancy of acquired information of the scene or overflown region (pipeline). In this sense, minimizing the entropy has the effect of maximizing the total information acquired. This term has the effect of requiring a combination of slow aircraft speed and/or increased task frequency and depends on both velocity of the aircraft, $v(t)$, and rate of image acquisition and processing $r_{\tau_2}(k)$

$$H = \sum_k \int_{t_k} e^{-\alpha\Omega(t,k)} dt. \tag{5.8}$$

Let α be a tuning parameter. The overlap between successive image footprints is then

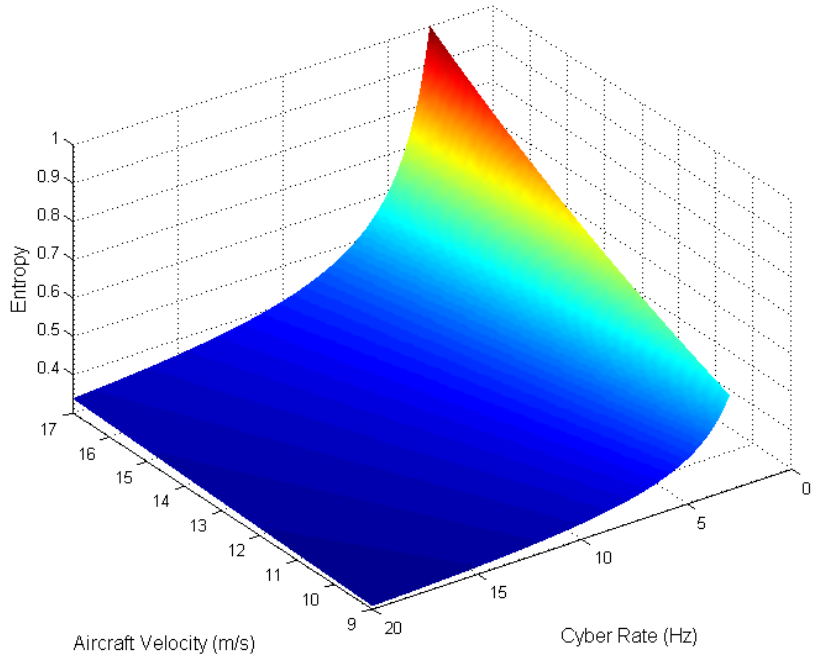


Figure 5.3: Entropy Cost H

$$\Omega(t,k) = \frac{1}{A}\left(A - w\int_{t-T_{\tau_2}(k)}^{t} v(\gamma)\,d\gamma\right) \tag{5.9}$$

where A is the total area of an image, w is the width of an image, and $T_{\tau_2}(k) = {}^{1}/_{r_{\tau_2}(k)}$ is the period of task τ_2. An exponential distribution was used as it provides a rapidly increasing penalty for flying too fast, and appropriate diminishing returns for flying slowly. For simplicity, we assume that the aircraft flies at approximately the same height above ground for the duration of the mission, and therefore A and w remain constant. In Figure 5.3 we show a plot of H to demonstrate how the entropy changes with both cyber rate (r_{τ_2}) and velocity

(v). The dependence of entropy on cyber rate falls off as a steep exponential, and falls off more gradually with aircraft velocity. This means we expect our Pareto front analysis in Section 5.3.1 on page 86 to indicate that lower entropy will be achieved primarily by increasing cyber rate rather than flying at a slower velocity.

5.1.2.3 Cost Function for Cyber System

The expressions in Equations (5.6) and (5.8) comprise the cost function for the cyber system

$$J_c = \frac{\beta_{c1}}{T r_{\tau_2,\max}} \sum_k t_k r_{\tau_2}(k) + \beta_{c2} \sum_k \int_{t_k} e^{-\alpha\Omega(t,k)} dt \tag{5.10}$$

where we have weighting terms β_{c1}, and β_{c2}. Such a cost function might be used if we were only interested in trading cyber resource utilization cost against reward for accomplishing mission objectives, which in the pipeline inspection case study maps to minimizing entropy (increasing information redundancy) that could be obtained through overlapping image data acquisition and processing.

5.1.3 CPS Cost Function

We combine J_p and J_c to obtain a holistic CPS cost function

$$J = \beta_{p1} \int_0^T P(v(t))\, dt + \beta_{p2} T + \frac{\beta_{c1}}{T r_{\tau_2,\max}} \sum_k t_k r_{\tau_2}(k) + \beta_{c2} \sum_k \int_{t_k} e^{-\alpha\Omega(t,k)} dt \tag{5.11}$$

In Section 5.3 on page 85 we select appropriate weighting terms to compare physical-only optimization, cyber-only optimization, and total system optimization to demonstrate how increased efficiency and conservation of energy can be achieved by including both physical and cyber objectives.

5.2 Setup and Solution

Our mission objective is to survey a straight segment of pipeline by flying a small, high aspect ratio UAS with a downward facing gimballed camera directly overhead. We have created a simulation in MATLAB to compare various solutions to the optimization problem posed. We first list the assumptions we've made to simplify the problem and demonstrate why an analytical solution is not possible. We then describe the numerical methods chosen to solve our optimization problem, and discuss the models we have adopted.

5.2.1 Assumptions

Equation (5.11) is non-trivial to solve in part due to the need to find the time-varying solution $v(t)$ and $r_{\tau_2}(k)$. It is further complicated by the discrete nature of the cyber system design variable $r_{\tau_2}(k)$ making this a mixed discrete-continuous equation. We make the following assumptions in order to simplify the problem:

1. The segment of pipeline is straight.
2. We assume aircraft performance is consistent with the principles of steady level flight.
3. The mission takes place close to sea level, with a relatively low altitude allowing use of standard sea level air density.
4. Altitude remains approximately constant through the mission.
5. Due to assumptions 1 and 2, the on-board gimballed camera always points straight down toward the ground. Specifically, this implies the optical axis of the camera is always perpendicular to the ground plane.
6. The scene, ground, and accompanying pipeline are approximately flat compared with the camera's height.
7. We restrict our problem to finding the optimal static v and r_{τ_2} that minimize the cost of the mission assuming $v(t)$ and $r_{\tau_2}(k)$ remain constant throughout.

An interesting addition to this work to be made in the future will be to model certain places on the pipeline as "high interest," either *a priori* or through real-time image processing, and therefore solve for the optimal trajectory with dynamically changing velocity and task execution rate.

5.2.2 Simplified Cost Function

Let $r_{\tau_2,\max}$ be the WCET of task τ_2, and D be the (constant) total straight-line distance of the entire mission. Based on the assumptions made and knowing the total distance, D, of the mission we can rewrite the overlap term, $\Omega(t,k)$, as

$$\Omega(v, r_{\tau_2}) = 1 - \frac{wv}{r_{\tau_2} A}$$

where w and A denote the constant width and area of an image footprint. Because we limit our problem to finding the optimal static v and r_{τ_2}, and owing to the fixed distance of the mission, we can replace the integrals in Equation (5.11) with the total corresponding quantities as a function of v and r_{τ_2}. This yields

$$J(v, r_{\tau_2}) = \beta_{p1}\frac{DP(v)}{v} + \beta_{p2}\frac{D}{v} + \beta_{c1}\frac{r_{\tau_2}}{r_{\tau_2,\max}} + \beta_{c2}De^{-\alpha\Omega\left(v, r_{\tau_2}\right)}. \tag{5.12}$$

In Figure 5.4 is a plot of $J(v, r_{\tau_2})$ in Equation (5.12). We note the convex shape and un-

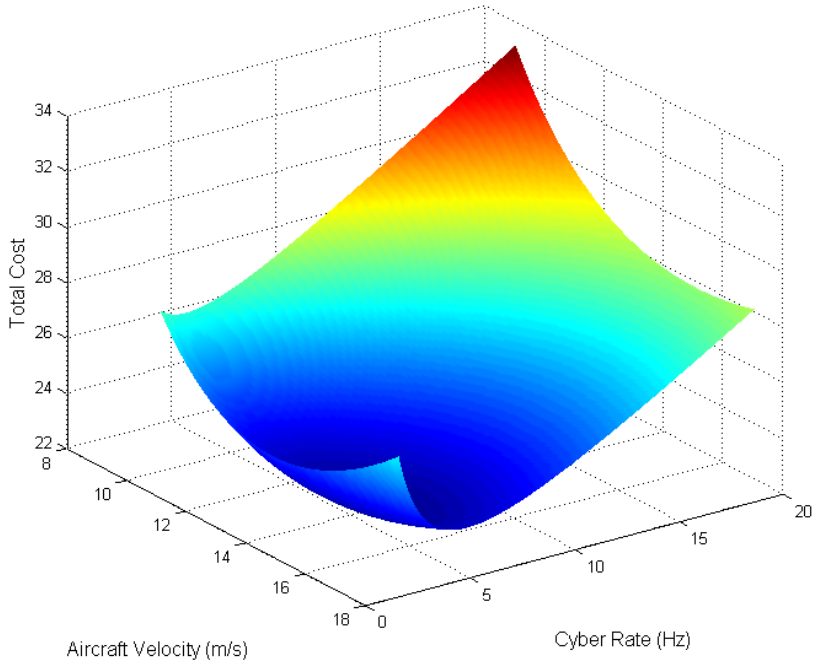


Figure 5.4: $J(v, r_\tau)$

constrained nature of the minimum, implying we should obtain a robust solution with an appropriate numerical optimization method.

A difficulty in any optimization scheme is the correct selection of weights for each cost metric. To equalize the contribution from each term, we normalize each thereby giving us more intuition later on as we choose our weighting terms $\beta_{p1}, \beta_{p2}, \beta_{c1}$, and β_{c2}. Because we have constrained our problem to realistic parameters for velocity, v, and mission task rate, r_{τ_2}, we can compute $\max\{E\}$ and $\max\{T\}$, occurring at the slowest velocity ($v_{\min}$), and $\max\{H\}$ occurring under conditions giving rise to lowest amount of overlap between images $\left(v_{\max}, r_{\tau_2,\min}\right)$. Combining these we obtain a new normalized cost function

$$J(v, r_{\tau_2}) = \beta_{p1} \frac{DP(v)}{v \max\{E\}} + \beta_{p2} \frac{D}{v \max\{T\}} + \beta_{c1} \frac{r_{\tau_2}}{r_{\tau_2,\max}} + \beta_{c2} \frac{De^{-\alpha\Omega\left(v, r_{\tau_2}\right)}}{\max\{H\}}. \tag{5.13}$$

After substitutions and some algebra we can rewrite Equation (5.13) as

$$J(v, r_{\tau_2}) = \beta_{p1}\gamma_1 v^2 + \frac{\beta_{p1}\gamma_2}{v^2} + \frac{\beta_{p2} v_{\min}}{v} + \frac{\beta_{c1} r_{\tau_2}}{r_{\tau_2,\max}} + \beta_{c2}\gamma_3 e^{\frac{wv}{r_{\tau_2} A}} \tag{5.14}$$

where

$$\gamma_1 = \frac{v_{\min} S C_{D_0} \rho}{2P(v_{\min})} \tag{5.15a}$$

$$\gamma_2 = \frac{2 v_{\min} K W^2}{\rho S P(v_{\min})} \tag{5.15b}$$

$$\gamma_3 = \frac{e^{-\alpha}}{e^{-\alpha\Omega\left(v_{\max}, r_{\tau_2,\min}\right)}}. \tag{5.15c}$$

5.2.3 Analytical Solution and Feasibility

We investigated the possibility of identifying a minimum for Equation (5.14) through analytical computation. Due to the constraints of the flight envelope and of the real-time computing system, let domain $\mathcal{D} \subset \mathbb{R}^2$ be

$$\mathcal{D} = \left\{ \begin{bmatrix} v \\ r_{\tau_2} \end{bmatrix} \in \mathbb{R}^2 \middle| \begin{bmatrix} v_{\min} \\ r_{\tau_2,\min} \end{bmatrix} \leq \begin{bmatrix} v \\ r_{\tau_2} \end{bmatrix} \leq \begin{bmatrix} v_{\max} \\ r_{\tau_2,\max} \end{bmatrix} \right\} \tag{5.16}$$

which is a compact set. From the Weierstrass theorem, $J(v, r_{\tau_2})$ in Equation (5.14), has a global minimizer [156].

In attempting to find an analytical solution we formulate the constrained optimization problem

$$\begin{aligned} \text{Minimize} \quad & J(v, r_{\tau_2}) \\ \text{subject to} \quad & v \leq v_{\max} \\ & v \geq v_{\min} \\ & r_{\tau_2} \leq r_{\tau_2,\max} \\ & r_{\tau_2} \geq r_{\tau_2,\min} \end{aligned} \tag{5.17}$$

To solve the constrained optimization problem we form the Lagrangian

$$L(v, r_{\tau_2}, \lambda_{1\ldots4}) = J(v, r_{\tau_2}) + \lambda_1 (v - v_{\max}) + \lambda_2 (-v + v_{\min}) + \lambda_3 (r - r_{\tau_2,\max}) + \lambda_4 \left(-r + r_{\tau_2,\min}\right)$$

where $J(v, r_{\tau_2})$ is from Equation 5.14 and $\lambda_{1\ldots4}$ are Lagrange multipliers. However, in applying the Karush-Kuhn-Tucker (KKT) necessary conditions to the Lagrangian we encounter a transcendental function

$$\nabla_v L = 2\beta_{p1}\gamma_1 v - \frac{2\beta_{p1}\gamma_2}{v^3} - \frac{\beta_2 v_{\min}}{v^2} + \frac{w\beta_{c2}\gamma_3}{r_{\tau_2} A} e^{\frac{wv}{r_{\tau_2} A}} + \lambda_1 - \lambda_2 \tag{5.18a}$$

$$\nabla_{r_{\tau_2}} L = \frac{\beta_{c1}}{r_{\tau_2,\max}} - \frac{wv\beta_{c2}\gamma_3}{A r_{\tau_2}^2} e^{\frac{wv}{r_{\tau_2} A}} + \lambda_3 - \lambda_4. \tag{5.18b}$$

We have attempted to solve this set of equations analytically by hand and by using mathematical software solvers and were unable to do so. This requires us to resort to numerical solutions.

5.2.4 Experimental Models and Setup

5.2.4.1 Aircraft

Table 5.1: UAS Model Parameters

PARAMETER	DESCRIPTION	VALUE
e_0	Oswald Factor	0.95
b	Wingspan	3.3 m
S	Surface Area	1.0 m^2
m	Mass	11.5 kg
C_{D_0}	Zero-lift Drag	0.04

The SolarDrones student team in the Aerospace Engineering Dept. at the University of Michigan has designed, built, and tested the solar-supplemented powered glider UAS Solar Sight illustrated without solar cells in Figure 5.5. In our simulation we presume the aerodynamic model parameters given in Table 5.1. This model leverages the well known power/velocity relationship of a single-engine propeller-driven aircraft as was described in Equation (5.2) [147]. From these model parameters we compute the remaining necessary parameters for the power equation which are shown in Table 5.2. While the proposed aircraft is solar-supplemented, the details of optimizing trajectories taking into account solar-generated power is not considered here. For this work we assume that the aircraft never generates more power than it can use and hence it is always advantageous to minimize energy over the mission[1].

[1] In some applications (e.g. CubeSats) it is sometimes necessary to frivolously expend energy through propulsion, thrust, or other mechanical means to avoid thermal issues associated with excess energy.

Figure 5.5: SolarSight Solar-powered UAS

Table 5.2: Additional Parameters for Power Equation

PARAMETER	DESCRIPTION	VALUE
$K = \frac{1}{\pi e_0 AR}$	Aerodynamic Parameter	0.0347
$AR = \frac{b^2}{S}$	Aspect Ratio	9.6628
$W = gm$	Weight	112.7 N
$\rho = \rho^s$	Air Density	$1.225\,\mathrm{kg/m^3}$

5.2.4.2 Camera

Table 5.3: Camera Model Parameters

PARAMETER	DESCRIPTION	VALUE
f	Focal Length	0.0046 m
H_{dist}	Horizontal Distance of Image Plane	0.00361 m
V_{dist}	Vertical Distance of Image Plane	0.00272 m

For the camera we use a standard pin-hole model consistent with specifications of the Panasonic GP-CX161/45P/E [157]. The key parameters for this camera are shown in Table 5.3. Given the pinhole assumption for simplicity, we *do not* model lens distortions and other effects. We also assume that the ground and pipeline are approximately flat compared to the much larger camera height above the ground. Because we know the camera height above ground at all times, presumed constant in steady level flight conditions, we can directly calculate the image footprint on the ground as a function of height above the ground. This is done by projecting the four corners of the image plane onto the (presumed) flat

ground forming the image footprint [158, 159]. We can then easily compute overlapping area between acquired images.

5.2.4.3 Experimental Setup

For the SolarSight aircraft, the stall speed (around $9\,\mathrm{m/s}$) and maximum power output of the engine/motor determine the bounds of v. For the cyber rate, r_{τ_2}, the lower bound was chosen based primarily on tuning the information cost in Equation 5.8. That is, at rates lower than 3 Hz there was no overlap between images resulting in coverage gaps thus maximum entropy ($H = 1$) presumed for the entire mission. The maximum cyber rate was chosen based on diminishing returns from cyber rates higher than 20 Hz. Bounds are therefore

$$9\,\mathrm{m/s} \leq v \leq 17\,\mathrm{m/s}$$
$$3\,\mathrm{Hz} \leq r_{\tau_2} \leq 20\,\mathrm{Hz}.$$

Additionally, via tuning, we chose the parameter $\alpha = 4$ in Equation (5.13), and chose the height above ground (from which we are able to derive A_i and w_i) to be 30 m.

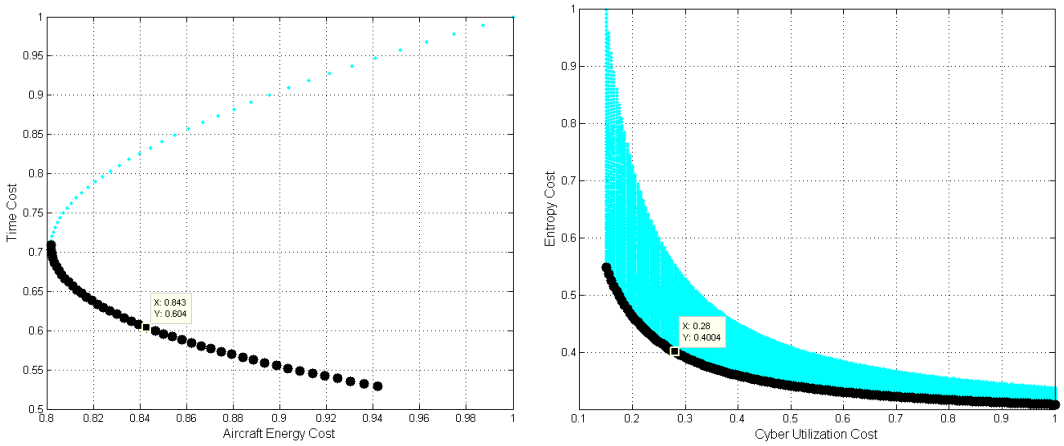
In our simulation we used MATLAB's `fmincon` function to solve the optimization problem. There are a variety of available algorithms, and we obtain equally good results with `fmincon`'s implementation of the Active-Set and Sequential Quadratic Programming (SQP) algorithms [160].

5.3 Results

We investigated the impact and tradeoffs between objectives from both the cyber and physical systems with the goal of minimizing energy use and time while maximizing information (minimizing entropy). Our goal is to show that simultaneous consideration of cyber and physical cost terms can yield more capable missions than what would be possible from designing these two parts of the CPS individually. We first examine and analyze Pareto fronts of the cost function in Equation (5.13) to gain insight into the tradeoffs from competing objectives. We select candidate points along the Pareto front of several plots representing true multi-objective optimization and use the corresponding v and r_{τ_2} to compute associated costs of the mission. We then select weights $\beta_{p1}, \beta_{p2}, \beta_{c1}$, and β_{c2}, optimize the CPS single objective cost function in Equation (5.13) and compare results with solution points selected from the Pareto fronts.

5.3.1 Pareto Fronts

Pareto front examination and analysis gives insight into the tradeoffs between competing objectives. Pareto front plots of J_p (Equation (5.5)) and J_c (Equation (5.10)) can be seen in Figure 5.6 where the black (darker) data points represent the Pareto front.



(a) Pareto Front for J_p (b) Pareto Front for J_c

Figure 5.6: Pareto Front for J_p and J_c

These curves show how the objectives for the physical and cyber systems, individually, trade off respective costs. The plots in Figure 5.6 follow their respective governing dynamical equations to produce the curves shown. For Figure 5.6a the plot is dominated by the power curve indicating we could expend similar amounts of energy, accomplishing our mission in very different lengths of time. Clearly to achieve our minimum time objective, the front side of the power curve is more optimal as indicated by the Pareto front. Because our entropy cost is a function of both v and r_{τ_2} we have multiple points corresponding to a single cyber rate r_{τ_2}. As a result, the velocities resulting in a higher entropy cost are dominated by those producing lower entropy.

If we choose a solution along one of these Pareto fronts we will optimize for either the physical or cyber portion of the system. Using these plots, we select the velocity corresponding with the data point highlighted in Figure 5.6a, $v = 14.9\,\mathrm{m/s}$. From the Pareto front for J_c we choose the cyber rate corresponding with the data point highlighted in Figure 5.6b, or $r_{\tau_2} = 5.6\,\mathrm{Hz}$. In Table 5.4 we show the costs associated with a mission using these parameters.

We can gain more insight into the tradeoffs of the entire cost function by also examining the tradeoffs between the CPS as a whole. We show these Pareto fronts in Figure 5.7 where in each subfigure we examine the tradeoff between three of the four objectives. In

Table 5.4: Costs For $v = 14.9\,\mathrm{m/s}$ and $r_{\tau_2} = 5.6\,\mathrm{Hz}$

COSTS	VALUES
E	16626.2 J
T	134.2 s
U_{τ_2}	0.28
H	66.7

Figure 5.7a we again observe the presence of the power curve governing the relationship between aircraft energy (E) and the other objectives. The curve folds over onto itself and we choose the point indicated in that plot which is in the crease of the function while also balancing cyber utilization (U_{τ_2}) and entropy (H) costs.

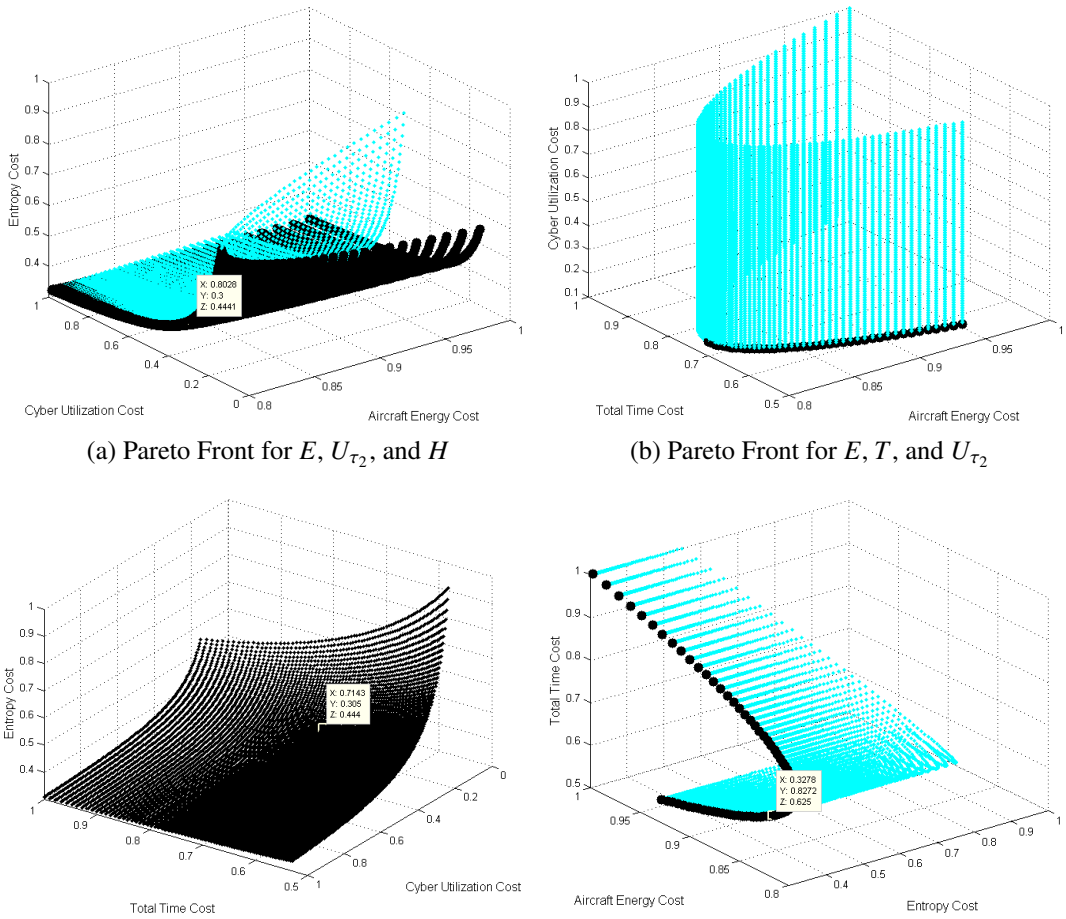
In Figure 5.7b no new insight or information is gained since the Aircraft Energy cost and Total Time cost (T) are independent of Cyber Utilization cost. Additionally, we note the similarity of this plot with the Pareto front for J_p in Figure 5.6a. In the Pareto front plot in Figure 5.7c there are no dominated points making the entire surface a Pareto front. We select the solution point indicated on this Pareto front that we determined from inspection of the plots which provides appropriate balance between the competing objectives.

Figure 5.7d shows the tradeoffs between entropy, aircraft energy, and total time costs. In this Pareto front we call attention to the normal tradeoff between total time and aircraft energy costs, but more interestingly the tradeoff with entropy cost. This shows the coupling between cyber and physical cost terms and gives insight into how they compete in the total cost. We follow our previous reasoning in choosing a point that compromises total time and aircraft energy but augmented by an attempt to minimize entropy as well.

We list the velocities, cyber rates, and corresponding costs for each of these three selected points in Table 5.5.

Table 5.5: Parameters and Costs for Data Points Selected from Pareto Fronts

PARAMETERS	E	T	U_{τ_2}	H
$v = 14.4\,\mathrm{m/s}$, $r_{\tau_2} = 15.4\,\mathrm{Hz}$	16314.5 J	138.9 s	0.77	45.2
$v = 12.4\,\mathrm{m/s}$, $r_{\tau_2} = 6\,\mathrm{Hz}$	15833.3 J	161.3 s	0.30	58.4
$v = 12.6\,\mathrm{m/s}$, $r_{\tau_2} = 6.1\,\mathrm{Hz}$	15816.9 J	158.7 s	0.31	58.4



(a) Pareto Front for E, U_{τ_2}, and H

(b) Pareto Front for E, T, and U_{τ_2}

(c) Pareto Front for H, T, and U_{τ_2}

(d) Pareto Front for H, T, and E

Figure 5.7: Pareto Fronts for J

5.3.2 Optimization over Total Cost Function $J(v, r_{\tau_2})$

In addition to examining Pareto fronts, using numerical methods, we can solve the single objective cost function $J(v, r_{\tau_2})$ in Equation (5.13), examine the resulting costs, and compare them with those found from the Pareto front analysis. This requires we select the weights for each cost term. Often there are practical reasons for favoring one cost term over another such as length of time since the last mission, or a cloudy day with less direct sunshine which might result in a tighter energy budget for our solar powered glider. Since we wish to compare holistic CPS optimization to independent physical and cyber system optimization we allow corresponding weights to go to zero as indicated in the 5^{th} and 6^{th} rows of Table 5.6. In each case, however, in the absence of any compelling reasons to favor one term over another we equalize all non-zero cost terms as shown. We compare the previous results from our Pareto analysis with our numerical solutions and show the

Table 5.6: Comparison of All Solutions

PARAMETERS	SOLUTION TYPE	E	T	U_{τ_2}	H	Total	RATIO TO BEST
$v = 14.9\,\mathrm{m/s}$, $r_{\tau_2} = 5.6\,\mathrm{Hz}$	Pareto from Table 5.4	16626.2 J	134.2 s	0.28	66.7	0.5587	0.25%
$v = 14.4\,\mathrm{m/s}$, $r_{\tau_2} = 15.4\,\mathrm{Hz}$	Pareto from Table 5.5	16314.5 J	138.9 s	0.77	45.2	0.6416	15.13%
$v = 12.4\,\mathrm{m/s}$, $r_{\tau_2} = 6\,\mathrm{Hz}$	Pareto from Table 5.5	15833.3 J	161.3 s	0.30	58.4	0.5682	1.96%
$v = 12.6\,\mathrm{m/s}$, $r_{\tau_2} = 6.1\,\mathrm{Hz}$	Pareto from Table 5.5	15816.9 J	158.7 s	0.31	58.4	0.5663	1.61%
$v = 15.2\,\mathrm{m/s}$, $r_{\tau_2} = 18\,\mathrm{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.5, \beta_{c1} = \beta_{c2} = 0.0$	16844.1 J	131.6 s	0.90	44.3	0.6708	20.37%
$v = 9.0\,\mathrm{m/s}$, $r_{\tau_2} = 4.3\,\mathrm{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.0, \beta_{c1} = \beta_{c2} = 0.5$	19722.8 J	222.2 s	0.22	58.7	0.6654	19.40%
$v = 14.2\,\mathrm{m/s}$, $r_{\tau_2} = 5.6\,\mathrm{Hz}$	J with $\beta_{p1} = \beta_{p2} = \beta_{c1} = \beta_{c2} = 0.25$	16208.8 J	140.8 s	0.28	64.9	0.5573	N/A

individual costs, as well as the scaled, and normalized total cost in Table 5.6. The lowest total cost solution is the last entry in the table wherein each individual cost term was given equal weight, and we compare its total cost with the other solutions as a percentage.

5.4 Conclusions

As technology allows us to shrink physical platform size, resource use by a cyber-system (e.g. the computational and communication components) begins to rival actuation effort of the physical system required for propulsion and servo actuation. This chapter investigates holistic optimization over both, demonstrating by example that co-design of the cyber–physical system results in a net savings of energy for given mission time and information gain by efficiently allocating cyber–physical system (CPS) resources.

Such a coupled co-design has been demonstrated in the form of optimization over cost functions, describing competing cyber and physical objectives. Unmanned aircraft system surveillance of a straight segment of pipeline was proposed as a baseline candidate mission, and simulation results were obtained. Pareto fronts of these results were analyzed, illustrating important tradeoffs between aircraft airspeed and task-execution rate for the candidate mission task. Optimal solutions were found to the combined CPS cost function, and were compared to results with independently optimized cyber and physical cost functions, demonstrating that large efficiency improvements can be realized by such an approach.

This chapter has focused on a simulation-based analysis of CPS optimization for a UAS. An actual flight test demonstrating results was beyond the scope of this project. We have, however, been able to validate the basic CPS cost tradeoffs introduced in this chapter using a single degree-of-freedom satellite simulator (TableSat) in a laboratory environment [161]. This paper is included as Appendix A.

An important enhancement will be to demonstrate a dynamic real-time planner that can appropriately adapt cost function parameters that, in turn, modify the flight plan to maintain optimal conditions based on feedback from sensors, onboard data-processing elements, and mission operators. Examination of more complex missions will require additional CPS cost metrics; it is anticipated that this will promote additional innovations in measuring task, cyber, and overall mission success. Complex missions could involve overflight of rugged terrain, target tracking, and interaction with other cooperative, non-cooperative, or hostile entities. Results from real-world flight operations would also bolster this work, enabling evaluation of static or ultimately dynamic optimization to improve overall CPS metrics during representative missions such as area coverage or inspection.

CHAPTER 6

Conclusion

Cyber-Physical Systems require new models, abstractions, and co-design techniques that account for the interdependency between components of the system. While CPS research is very broad in scope in this dissertation we have focused on the interactions between computing and control of a mobile physical system and have introduced methods to co-regulate and co-optimize a CPS as summarized below.

6.1 Co-Regulation of CPS

We have presented and evolved an abstraction of the task execution rate of the controller (i.e. sampling rate) as a continuous-valued dynamics system to couple with a traditional linear system model of a physical system. This CPS model will allow an engineer to design a feedback controller for both the cyber and physical system together, exploiting the coupling between them. The result is a CPS model lying between the extremes of Riemann sampling on the one hand, and Lebesgue sampling on the other [17]. The sampling rate of the CPS is regulated in discrete time according to a feedback controller and thereby reaps many of the benefits of Riemann scheduling theory and digital control while also offering much of the cyber resource savings implicit in "as-needed" Lebesgue sampling. In a broader context this co-design technique gives the cyber system awareness of the needs and performance of the physical system and allows the cyber system to adjust controller resources appropriately.

Along with this abstraction we have proposed two new controllers for our discrete-time-varying CPS:

- **GSLDQR**. A Gain-Scheduled Discrete Linear Quadratic Regulator controller employs DLQR gains designed at various operating points, or fixed sampling rates, of the CPS. Those gains are scheduled over the range of possible sampling rates so that at each sampling time the corresponding DLQR gain is used to compute the next

control input for the physical system. To our knowledge this is the first time gains have been scheduled over sampling rates of a system. While this controller provided excellent cyber resource savings it did not compare favorably with traditional fixed-rate DLQR controller for reference tracking or control effort. It is possible that an investigation of the values of the **Q** and **R** weighting matrices could yield improved performance.

- **FPRB**. We designed a Forward-Propagation Riccati-based controller wherein the gain is computed by propagating the discrete Riccati equation forward in discrete time (as opposed to backward in time which is traditionally done) according to the discrete system dynamics at each time step. We have added important empirical evidence that these techniques can be broadly applied and provide stable control with good performance for our class of systems.

We have applied our CPS co-regulation technique to three systems and used our metrics developed for CPS to compare performance. First, our co-design methodology was applied to a spring-mass-damper system to illustrate proof of concept. A more challenging unstable system, an inverted pendulum, was then used and several test case scenarios for both systems were explored. Scenarios explored the increase or decrease of sampling rate as a result of a new reference command scenario representing the new goals of a higher-level planner to increase performance or scale back cyber-resource use respectively. Results demonstrated that cyber resources could be dynamically adjusted to meet new planning goals.

Second, we have applied our co-regulation methodology to attitude control of a CubeSat using reaction microwheels where active control is required due to unstable system dynamics. This is the first time variable sampling rate control has been applied to spacecraft. In this case study we focus on disturbance rejection thereby highlighting the effectiveness of feedback control over optimal sampling pattern techniques [13] in regulating sampling rate. Results indicate that significant cyber resource savings (28% for our sample CubeSat) as well as significant control effort savings (12% for our sample CubeSat) can be achieved while maintaining near optimal tracking performance.

6.1.1 CPS Co-Regulation Future Work

This work would greatly benefit from future development of discrete-time-system theory and proof of stability guarantees for both the GSDLQR and FPRB control law formulations. Similarly, the design of feedback control laws for both the physical and cyber system needs further analysis to develop an optimal methodology for choosing coupled CPS gains for

the time-varying sampling rate system. Finally, the application of our technique to real hardware, software, and mission conditions is vital for gaining insight into further tradeoffs between cyber and physical resource use at the feedback control level.

6.2 Co-Optimization of CPS

New metrics for CPS design and performance evaluation is one of the grand challenges of CPS research [2]. At the planning level, we have devised new CPS metrics and presented a novel method for trading off cyber and physical resources to obtain improved mission performance. Using a UAS surveillance case study we developed metrics representing energy, time, cyber utilization, and surveillance information and formulated a cost function to improve mission performance. The cost is a function of UAS velocity and rate of execution of a mission-critical surveillance task which collects, processes, and stores imagery of a pipeline. We show and analyze Pareto fronts of the competing objectives to gain insight into the tradeoffs between cyber and physical resources for mission success. The cost function is then optimized numerically and results are tabulated comparing physical-only cost, cyber-only cost, and joint cyber-physical cost. We have demonstrated that increased mission success can be achieved by considering cyber and physical resources together rather than independently. We envision such a planning scheme could serve to provide reference commands to the lower-level co-regulation system also presented in this dissertation.

This dissertation presents an approach to co-optimization that is innovative with respect multi-disciplinary CPS models, metrics, and Pareto analysis. As proof of concept we limited our optimization to finding constant values for velocity and task rate. However, a dynamic planner that updates reference commands based on *in situ* measurements and feedback of information for both the cyber and physical system remains an important topic for future work and will likely prove quite valuable for improved mission success. While in Appendix A we have provided limited hardware verification of our proposed technique, our work would be further strengthened by real vehicle-based demonstration of its effectiveness in more complex scenarios, missions, terrain, etc.

The work in this dissertation represents progress toward co-design of the computational control systems for CPS. We hope that future work building on these ideas will spawn the next generation of more capable, aware, reliable, and secure Cyber-Physical Systems.

APPENDIX A

Mission-Aware Cyber-Physical Optimization on a Tabletop Satellite [161]

A.1 Introduction

Modern systems require sensors, actuators, algorithms, and real-time digital systems to coordinate their activities with a physical system to achieve designated goals. Often each of these individual subsystems are designed independently to meet performance objectives. Important system design properties such as compositionality and composability can suffer without co-design techniques that account for limitations and strengths of each subsystem as well as the physical objects with which the system interacts. Additionally, as systems become smaller, requiring less energy for actuation and sensing, computational (cyber) resources begin to demand energy comparable to that of the physical system.

Control systems engineers attempt to optimize physical system trajectories by the proper application of force over time. Physics-based models of system dynamics including saturation constraints and other nonlinearities are used to design control laws that achieve designed trajectories - most often in the continuous time domain. On the other hand, real-time systems engineers, using discrete mathematical tools, optimize task allocation and scheduling over processor, communication, and Input/Output (I/O) resources to guarantee performance deadlines for reliability and robustness. Good design of the task schedule may provide enough slack so that energy can be conserved through variable speed processors. Alternatively, slack in the task schedule may allow for aperiodic and sporadic task guarantees thereby providing event-driven capabilities or simply just increased service of individual tasks.

For systems which must more carefully manage all their physical and cyber resources together to achieve their objectives, globally-optimal (minimum-energy, minimum-time,

maximum-information) performance can only be achieved by identifying and exploiting coupling between cyber and physical resources during the design process. Cyber resources provide the means for guidance, navigation, and control of the physical system, as well as the estimation of states, communication, and processing of information. The physical system, in turn, provides the ability to acquire information, survey an area, or take important measurements.

We have been conducting research to try and accomplish such optimal cyber-physical co-design. In recent work we developed a multidisciplinary approach for optimizing over both cyber and physical resources, including mission goals and objectives [28, 162]. Using metrics encompassing physical system energy, time, surveillance information, and cyber utilization we showed that we can more appropriately balance overall system performance. We also used Pareto front analysis to examine some of the coupling between cyber and physical resource use.

In this paper we take this research a step further by trying to experimentally validate this technique. To this end we have adopted TableSat [163], a one-degree of freedom rotating platform emulating a small satellite to demonstrate this method of design. TableSat, shown in Figure A.1, uses computer fan actuators, rate gyros, and accelerometers controlled by an on-board Gumstix computing platform to control its rotation. We have fitted TableSat with

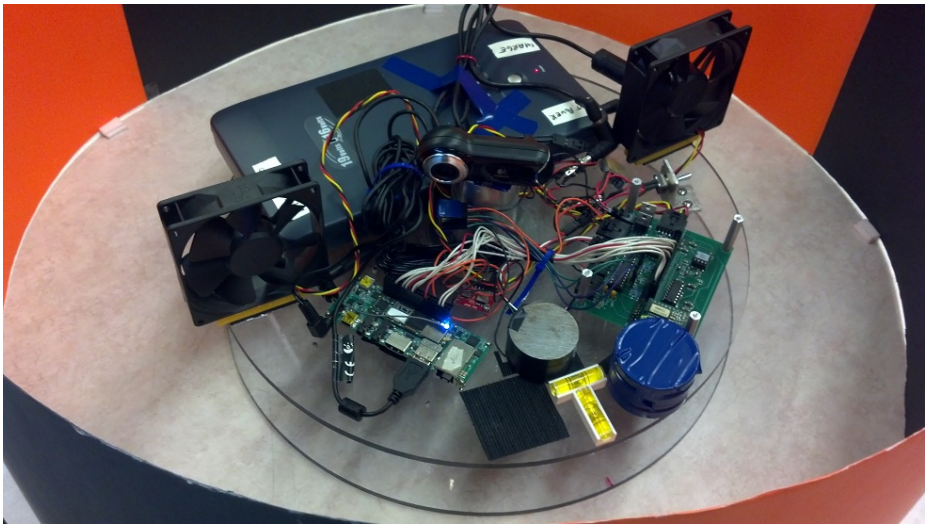


Figure A.1: TableSat

a camera and propose an example mission to emulate a satellite gathering important data. The example mission is to collect the maximum amount of "information" which we define to be number of orange pixels viewed by the camera. The colored paper setup can also be seen in Figure A.1.

We first introduce some related work in the fields of real-time systems and aerospace systems. We then describe the cost function including the metrics describing the competing mission objectives. A description of our experimental setup follows, after which we present analytical and subsequently experimental results.

A.2 Related Work

The design of control systems under the constraints of cyber resources is not a new area of research. Anytime control [81–83] tries to improve control accuracy as cyber resources become available. Feedback scheduling [55, 84, 86] is a technique wherein cyber resources are modified according to the needs of the cyber system. However, only recently in the context of aerospace, satellites, avionics systems, surveillance, and UASs has there been increasing work attempting to address this important issue.

Agrawal et. al. explore some of the reasons why more advanced control algorithms are not used in modern avionics systems. They conclude that a Quality of Service (QoS) approach is needed to address the problem and they propose an adaptive resource management scheme for a real-time avionics system using anytime control and accompanying nontraditional task scheduling [108]. Russ and Sttz proposed a higher-level style of resource management that includes task-based guidance and navigation and perception plans. Their method focuses on finding algorithmic solutions adapting to perceptual demands that vary during flight as well as balancing those demands with sensory and computational resources [111]. Narayan et al. present a novel computationally adaptive trajectory decision optimization system that can dynamically manage, calculate, and schedule task execution parameters [112]. An offline and online component work together to increase overall mission efficiency.

For satellite systems information gathering by imaging systems generally happens in a relatively short time window (in low earth orbits around three min) during which the system must maximize its efforts to collect the data. There is generally a 10-15 minute window during which the system can prepare resources for intense data collection. Traditionally, such task scheduling problems have been addressed by teams of planners on the ground using write and check procedures [101]. However, automated methods have been proposed and used with success. Bataille et. al. examine and design for physical constraints, fairness, and efficiency for different agents using a shared resource (an earth observing satellite) [106]. Bresina et. al. combine together two techniques - GenH which generates a specialized search heuristic, and HBSS which employs the heuristic within a stochastic sampling method - to automatically generate high-quality schedules with respect

to an objective function [107].

Our work complements existing research by providing a mechanism to optimize over cyber-physical resources as well as mission objectives for the holistic system while leveraging optimization theory, and more particularly, optimal control. Existing solutions to dynamically adjust parameters can provide the tools by which a system could use our methodology to produce more efficient missions according to the individual metrics chosen in the cost function.

A.3 Cost Functions

We desire to optimize over both physical and cyber characteristics to improve efficiency and performance for our designated mission. We do this by developing *cost* terms for physical system energy, cyber utilization, as well as mission-critical task performance. This means developing cost terms for the energy consumed by the fan actuators as function of desired angular velocity, a term representing the energy consumed by the computing system, and a metric measuring the amount of information we gather from the on-board camera.

For the physical system, we wish to determine the optimal angular velocity of TableSat for our proposed mission. Because the on-board camera is fixed and rotating with TableSat, and since we only allow for rotation in one direction, we need only consider simplified motion about a single axis. For the cyber system, our objective is to determine the optimal execution rate of a mission-specific task. This task is a real-time surveillance task in which we wish to collect data on orange colored segments of construction paper. A single complete job of this surveillance task consists of acquisition, processing, and storage of an image. We assume that other system-critical cyber tasks, including the control task, have been allocated a fixed amount of resources. Our surveillance task therefore operates within the remaining non-critical bandwidth available in the cyber system.

We build our cost function from the ground-up, integrating physical and cyber cost terms and associated functions. "Physical" in the context of a TableSat includes items related to rotation of the system, for example, the rotating table, fan actuators, and sensors. "Cyber" relates to items required for image collection, data processing, computation of control inputs, etc.

A.3.1 Physical System Terms

To balance the goals of the physical system we seek to minimize total energy consumption by the fan actuators while also minimizing time required for mission completion.

A.3.1.1 Physical System Energy

In TableSat all the energy required to rotate the table is consumed by the fan actuators. The metric then for physical system energy becomes

$$E_p = \int P(\omega)\, dt. \tag{A.1}$$

Where P is power as a function of the angular velocity, ω. Because we use simple computer fans controlled by a Pulse Width Modulation (PWM) signal we assume power consumed is simply

$$P(V) = IV \tag{A.2}$$

where I is the constant current draw (a parameter of the fan), and V the time averaged voltage "seen" by the fan via the PWM duty cycle. As voltage is a function of PWM signal which is our control input, we can rewrite Equation A.2 as

$$P(d) = IV_{\max} \frac{d - d_{\min}}{d_{\max} - d_{\min}}$$

where d is the PWM duty cycle and $d_{\min}$ and $d_{\max}$ are the constant minimum and maximum duty cycles respectively as experimentally determined. $V_{\max}$ is a constant parameter of the fan and is the maximum voltage the fan can accept.

Because we require a power function that takes angular velocity as a parameter for optimization we conducted a series of experiments commanding a PWM duty cycle, awaiting approximately steady state angular velocity, and subsequently timing the rotational speed. We used MATLAB's `cftool` to determine a linear curve fit as seen in Figure A.2. We note that higher order polynomials would provide a better fit. We explicitly chose a linear fit due to the uncertainty surrounding our friction coefficients between experiments.

Functionally, we have power as a function of angular velocity given by

$$P(\omega) = IV_{\max} \left(\frac{p_1 \omega + p_2 - d_{\min}}{d_{\max} - d_{\min}} \right) \tag{A.3}$$

Using the coefficients experimentally determined we can reduce Equation A.3 to

$$P(\omega) = 0.514\omega - 0.292 \tag{A.4}$$

The experimentally determined power curve and associated energy curve can be seen in Figure A.3. We note the flattening out of the energy curve as angular velocity increases. This suggests that at some point it does not cost us significantly more to rotate much faster.

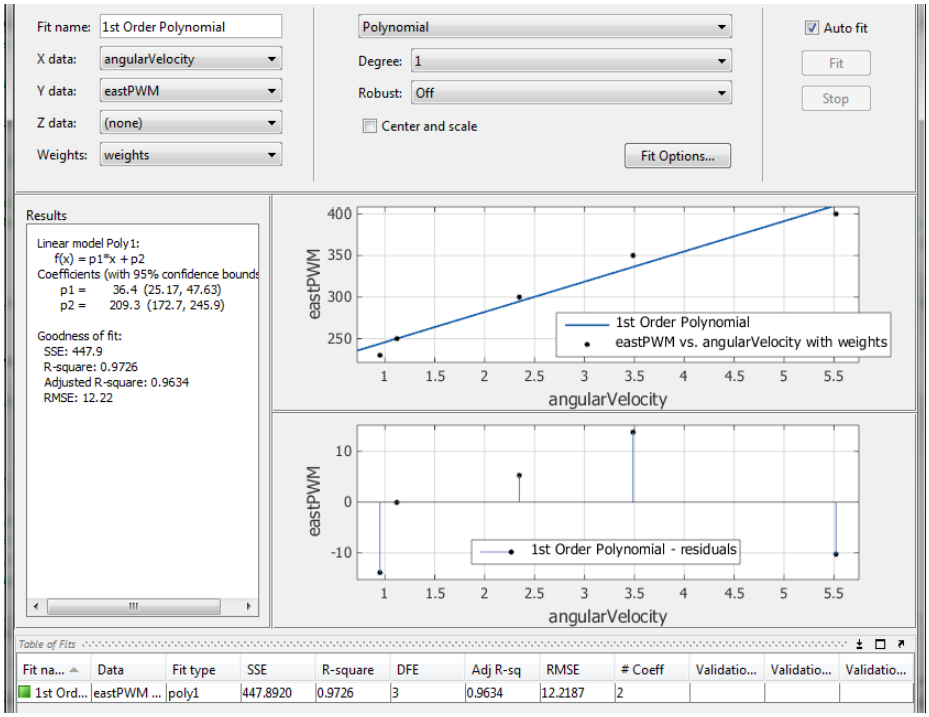


Figure A.2: Linear Curve Fit Using MATLAB's `cftool`

This observation is seen later on in the final cost function.

A.3.1.2 Time

In addition to minimizing power consumption, we would also like to minimize the amount of time required to accomplish our mission. Such time-minimal optimization cost terms are common, and are simply given by

$$T = \int dt. \tag{A.5}$$

A.3.1.3 Cost Function for Physical System

The two competing cost metrics, E_p and T, comprise the cost function for the physical system

$$J_p(\omega) = \beta_{p1} \int P(\omega)\, dt + \beta_{p2} \int dt$$

where β_{p1}, β_{p2} are weighting terms. Optimizing J_p alone is what a traditional trajectory or path planner would do if no costs are attributed to the cyber system.

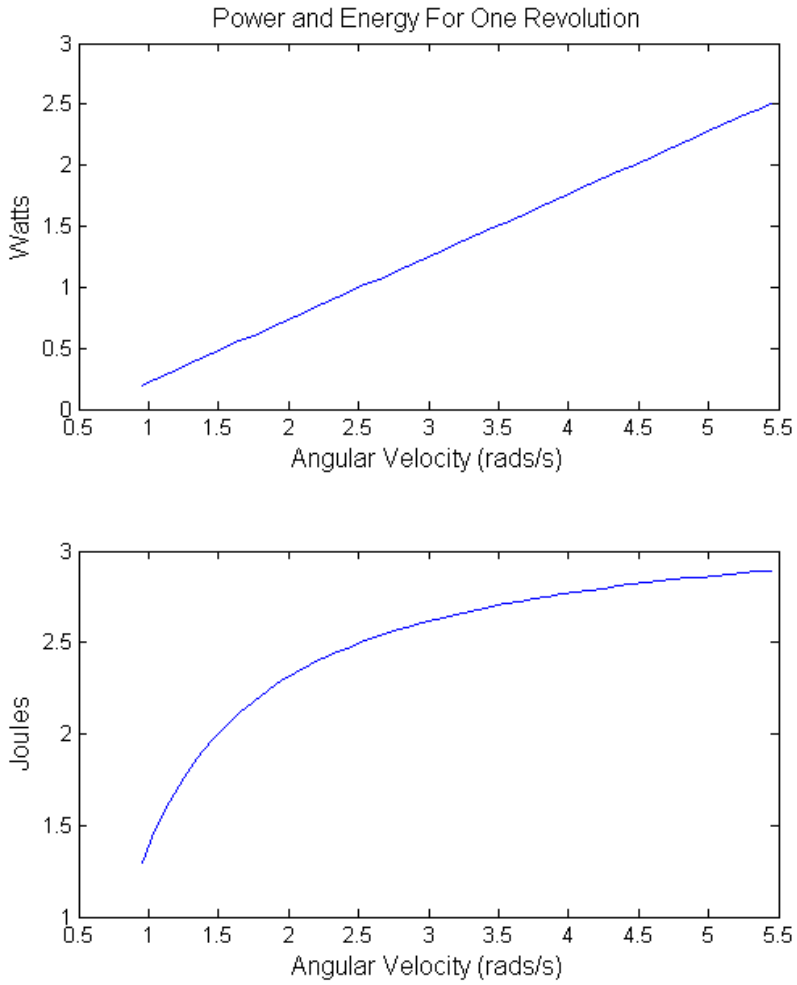


Figure A.3: Power and Energy Curves for TableSat Physical System

A.3.2 Cyber System Terms

For a modern autonomous system the cyber portion becomes the center point for data collection, actuation inputs, communication, I/O, path planning, control input calculation, etc. In many real-time systems task execution rates are determined *a priori* based on requirements of the system and/or mission. For example, the sampling rate of the control task may be selected based on digital control analysis. In this work we do not interfere with such high-priority task assignments and focus instead on the tuning of a lower-priority mission-critical task giving us the ability to optimize over mission and cyber parameters without interfering with safety-critical tasks. We assume that we can not only conserve energy by optimally selecting execution rates of lower-priority tasks, but we can also increase mission effectiveness by developing costs that relate task execution rates to mission efficiency.

A.3.2.1 Cyber Utilization

In real-time system scheduling theory online schedules can be created by examining relative deadlines of independent periodic tasks as in the Earliest Deadline First (EDF) scheduling algorithm. Such optimal scheduling algorithms are dynamic by assigning task priority as jobs are released to the operating system for scheduling [58]. They therefore have the ability to respond to changing deadlines and periodic rates.

In this paper, we assume that at least part of the cyber utilization is fixed based on selected and scheduled periodicity of mission-critical tasks, and instead focus on maximizing use of the remaining resources. To that end, we assume a single task, τ, achieves the important mission goal of capturing and processing an image of our orange and black cylindrical backdrop (see Figure A.1). The task runs at execution rate r_τ Hz and has a maximum execution rate of $r_{\tau,max}$ Hz stemming from restrictions based on available cyber resources. That is, we ensure schedulability of the task based on $r_{\tau,\max}$ but allow that period to change resulting in freed cyber resources to be devoted to increased service of other tasks or simply to conserve energy. We introduce the cyber utilization term

$$U_\tau = \frac{r_\tau}{r_{\tau,max}}. \tag{A.6}$$

We note that r_τ is the rate of execution of task τ throughout the mission. We assume that cyber utilization is proportional to energy consumed by the cyber system, and as a result, minimizing it is the cyber analog to the energy minimization term of the physical system in Equation (A.1).

A.3.2.2 Mission Information

In developing our cost metrics, we seek to relate mission efficiency to cyber and physical parameters. Our specific mission objective is to aquire images of orange squares on a poster board backdrop. We can consider this mission equivalent to collecting as many orange pixels as possible. To appropriately model the collection of orange pixels we make the simplifying assumption that the orange squares of the backdrop are laid out continuously (as opposed to an alternating pattern with black) covering θ_{orange} radians of the rotation. We then find the number of frames F collected during the interval which is a function of both angular velocity and cyber task execution rate

$$F = \frac{r_\tau \theta_{\text{orange}}}{\omega}. \tag{A.7}$$

We note that this is a continuous function and does not truly represent the discrete number of frames taken. However, this approximation allows us to use existing mathematical tools to solve the problem. We then determine the amount of sweep made by the camera in F frames and including the camera's footprint γ in radians

$$\theta_F = \gamma + \theta_{\text{orange}}. \tag{A.8}$$

We assume that the camera can **only** see the colored backdrop and that all pixels will be counted as either orange or black. We can then combine Equations (A.7) and (A.8) to determine the number of orange pixels per radian seen by the camera which we call mission "information"

$$I(\omega, r_\tau) = \frac{FP\theta_{\text{orange}}}{2\pi\theta_F}$$

where P is the total number of pixels in an image (for a 640×480 camera this is 307,200 pixels).

Naturally, this metric as we have described it, should be maximized to produce the greatest mission success. Traditionally, we would simply minimize $-\int I(\omega, r_\tau)dt$ for the equivalent. This results in the metric shown in Figure (A.4). Such a metric becomes prob-

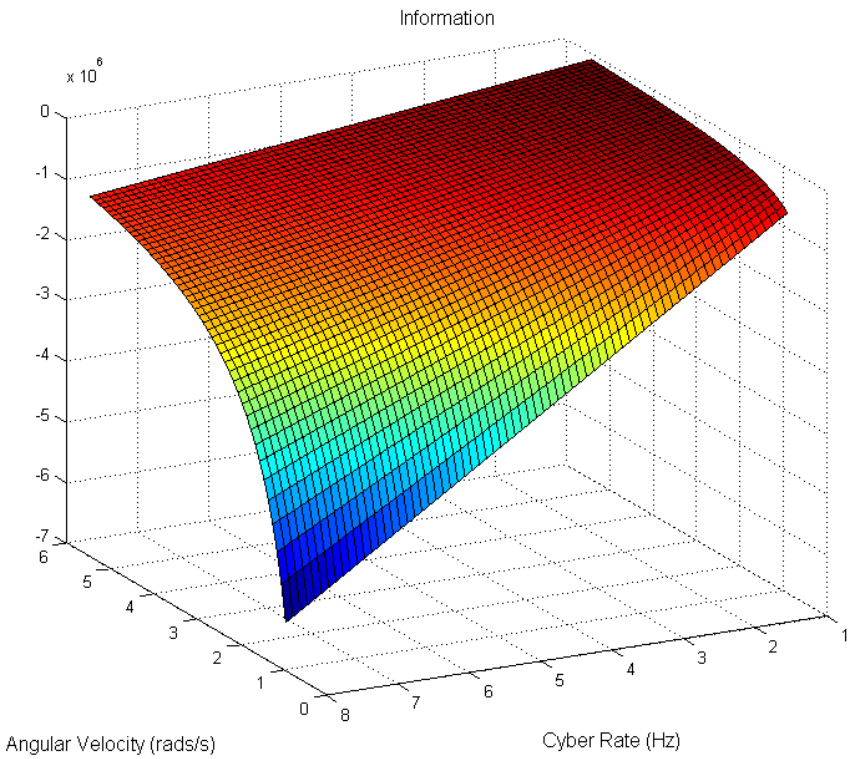


Figure A.4: $-I(\omega, r_\tau)$

lematic in the total system cost function because it is concave rather than convex. This in turn means the final cost function has a saddle point and falls off sharply toward the boundaries. To be able to solve a well-behaved convex shaped optimization problem we need a mission success metric that, when minimized, produces maximum mission success. Therefore we propose

$$M = \int \frac{1}{I(\omega, r_\tau)} dt \tag{A.9}$$

which can be seen in Figure A.5. We note the nonlinear dependence of M on both angular

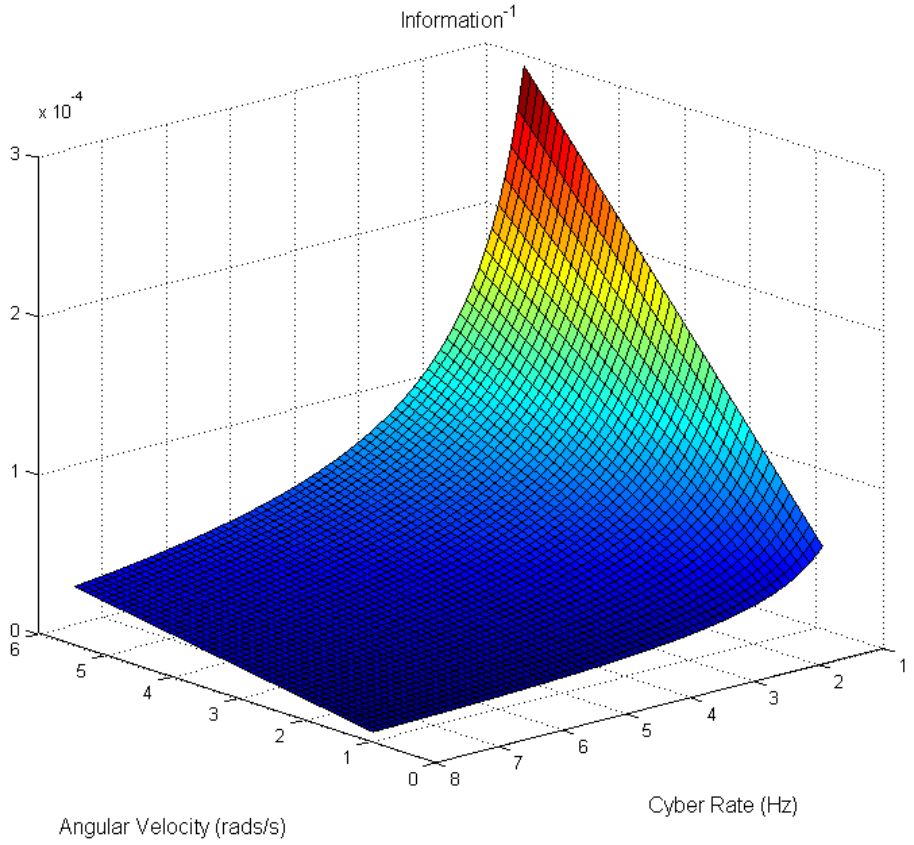


Figure A.5: Information Metric

velocity, and the cyber rate. We also note that the dependence on cyber rate falls off as a steep exponential, and falls off more gradually with aircraft velocity. This means we expect optimal solutions to achieve higher mission success by increasing cyber rate than by going slower.

A.3.2.3 Cost Function for Cyber System

The expressions in Equations (A.6) and (A.9) together comprise the cost function for the cyber system

$$J_c(\omega, r_\tau) = \beta_{c1}\frac{r_\tau}{r_{\tau,max}} + \beta_{c2}\int \frac{1}{I(\omega, r_\tau)}$$

where we have weighting terms β_{c1}, and β_{c2}. We might independently optimize over such a cost function if we were interested solely in trading cyber resource utilization cost against reward from accomplishing mission objectives.

A.3.3 CPS Cost Function

We desire, however, a holistic cost function that gives us optimal values of ω, and r_τ for the total system including mission objectives. We therefore combine J_p and J_c to obtain a CPS cost function

$$J(\omega, r_\tau) = \beta_{p1}\int P(\omega)\,dt + \beta_{p2}\int dt + \beta_{c1}\frac{r_\tau}{r_{\tau,max}} + \beta_{c2}\int \frac{1}{I(\omega, r_\tau)}dt \tag{A.10}$$

In Section A.5 we will manipulate the weights to compare physical-only optimization, cyber-only optimization, and total system optimization, to illustrate the efficacy of our co-design methodology.

A.3.3.1 Simplified Cost Function

Although Equation (A.10) could be solved for the optimal trajectories for ω, and r_τ, as in our previous work [162] we limit our solution to finding the static ω, and r_τ that minimize the cost function. If we know the total "distance" D of the mission (for TableSat we use the rotational distance of 2π radians), we can replace the integrals in (A.10) with more straightforward sums, simplifying the equation to

$$J(\omega, r_\tau) = \beta_{p1}\frac{DP(\omega)}{\omega} + \beta_{p2}\frac{D}{\omega} + \beta_{c1}\frac{r_\tau}{r_{\tau,\max}} + \beta_{c2}\frac{1}{DI(\omega, r_\tau)}. \tag{A.11}$$

Note that this function is convex with an unconstrained minimum, so finding a solution is straightforward.

Choosing appropriate weights is a difficult part of the design process, and is made much harder when different cost metrics take on different ranges of values. If we normalize the terms, the weights we choose will make intuitive sense and can be meaningfully compared between terms. Using the maximum possible values for each term, we are able to normalize

Equation (A.11) to

$$J(\omega, r_\tau) = \beta_{p1} \frac{DP(\omega)}{\omega \max\left\{E_p\right\}} + \beta_{p2} \frac{D}{\omega \max\{T\}} + \beta_{c1} \frac{r_\tau}{r_{\tau,\max}} + \beta_{c2} \frac{1}{DI(\omega, r_\tau) \max\{M\}} \tag{A.12}$$

where$\max\{E_P\}$ is found by assuming the maximum angular velocity for the mission, $\max\{T\}$ is found from the slowest angular velocity, and $\max\{M\}$ from the fastest angular velocity and slowest cyber rate. We solve the problem with numerical methods, which we describe more in Section A.5.1.

A.3.3.2 Optimization Problem

This leads to the constrained optimization problem we wish to solve

$$\begin{aligned} &\text{Minimize} J(\omega, r_\tau) \\ &\text{subject to} \omega \leq \omega_{\max} \\ &\qquad \omega \geq \omega_{\min} \\ &\qquad r_\tau \leq r_{\tau,\max} \\ &\qquad r_\tau \geq r_{\tau,\min} \end{aligned} \tag{A.13}$$

where $\omega_{\min}, \omega_{\max}, r_{\tau,\min}$, and $r_{\tau,\max}$ are determined based on limitations of the physical system along with diminishing returns of success, and are

$$0.95^{\text{rads}/\text{s}} \leq \omega \leq 5.52^{\text{rads}/\text{s}}$$
$$1\,\text{Hz} \leq r_\tau \leq 8\,\text{Hz}$$

A.4 Experiment Setup

Our objective is to survey orange segments of an orange and black paper wall surrounding a rotating table-top satellite, TableSat, using a fixed camera mounted at the center. The surveying mission requires two configurable settings, angular velocity and task rate. Below is a description of the hardware and software components of our application, as well as the assumptions and model parameters that arose from the components.

A.4.1 TableSat Overview

TableSat is a table-top satellite with one degree of rotational freedom. It has a base with a central post that comes to a conical point. The board itself balances on the post via a

screw located at the middle of the board. Because the contact is reduced to that single point, TableSat has a very low coefficient of friction. It rotates around the base using two standard computer cooling fans which take PWM signals to dictate fan rotations per minute (RPMs). The computer-on-module (COM) is a Gumstix Overo with a Tobi expansion board that provides GPIO, such as that used for PWM output. TableSat also has a camera mounted at the center that connects to the COM through the USB port.

A.4.2 Hardware

A.4.2.1 Gumstix

The TableSat COM is a Gumstix Overo with a 1GHz ARMv7 processor and 512 MB DDR RAM. The Gumstix runs Linux 2.6.36, which is not a real-time operating system (RTOS). That means that because our software timers are run at the user-level, as opposed to the kernel-level, the timers may be limited in their precision. The slow speed of the processor also means that the maximum frame capture and process rate is 8 frames per second (FPS). Finding an efficient means of employing this limited resource is therefore crucial.

A.4.2.2 Fans

The actuators are two Sunon KD1209PTB2 computer cooling fans with a max rated speed of 2600 RPM. The operating voltage ranges between 5 and 13.8 volts. They are rated to draw 0.2 amps of direct current and 2.4 watts of power.

The fan speed is set by sending a PWM signal over the control line. The characteristics of the PWM signal is specified in code by sending a byte sequence to the control line. We needed to discover which bytes corresponded to the minimum and maximum PWM signals, and ensure that the range of intermediate byte values mapped linearly to the signal. We connected the PWM line to an oscilloscope and determined what bytes corresponded to which PWM signals. In particular, we figured out what bytes corresponded to max speed. In this way, we could tell what fraction of max fan speed we were signaling. By running a series of timed experiments and fitting a curve to the resulting data, we were able to map PWM signal to angular velocity. As described in A.3.1.1, to model the energy usage of the fans, we assumed that the power draw is proportional to the average amount of time the PWM signal is up.

A.4.2.3 Camera

The webcam is a Logitech QuickCam Pro 9000 that connects to the board via USB. It has a max frame rate of 30 FPS and a resolution of 640×480. The camera also has an automatic brightness/contrast adjustment feature that must be considered and accounted for when attempting computer vision tasks.

A.4.3 Software

A.4.3.1 Frame Capture

Camera access and frame processing was done using OpenCV libraries. We wrote a function that captured a 640×480 frame from the webcam, converted it to a HSV (Hue-Saturation-Value) colorspace and then thresholded it on orange. Thresholding produces a black and white image where white pixels indicate the presence of orange on the original image. Mathematically, the resulting image has $640 \times 480 = 307,200$ pixels, each of which takes on a value, v, of either 255 or 0, where 255 indicates orange detection. The total number of orange pixels in the original image, N, is thus given by

$$N = \frac{\sum_{i=1}^{307,200} v_i}{255}$$

Once the camera is initialized, this capture-and-process task can be called whenever desired.

A.4.3.2 Cyclic Executive

Our program takes two parameters: angular velocity and the task rate for our periodic task. Since we only consider the static case in this project, we set the fan speed in the beginning that corresponds to the desired angular velocity, and the only periodic task is then capturing frames from the camera.

After the PWM signal is initialized, we wait three minutes for the TableSat to spin up to steady state angular velocity. Then, when the experimenter presses ¡ctrl¿-c, the program starts capturing frames at the specified frame rate. This is done using a cyclic executive that consists of one task, the frame capturing/pixel counting task. When the experimenter presses ¡ctrl¿-c again after the desired number of revolutions, the system stops and some metrics are displayed - the number of white pixels seen, the number of total pixels captured, and the amount of time that the cyclic executive executed for.

A.4.4 Assumptions

We assume that friction is constant, or at least consistent. This is not necessarily true, because even when the TableSat is perfectly balanced at rest, as it rotates it begins to precess due to the heterogeneous mounted components applying non-uniform forces to the board. As the TableSat precesses, it changes position on its base, which changes the coefficient of friction over time.

To properly maintain a constant speed, we would need to adjust the fan signal in response to the changing friction. However, in our experiments, we send a constant PWM signal and assume that after some time the TableSat will reach a steady state speed. As noted in A.4.2.2, we mapped PWM signals to steady state angular velocities using the results from a set of timed trials. Assuming constant (or consistent) friction is necessary for the assumption that the same PWM signal will always result in the same steady-state velocity. Since we are using open-loop control, this assumption is crucial.

A.5 Results

We first investigated the simulated impact and tradeoffs between objectives from both the cyber and physical systems with the goal of minimizing energy use and time while maximizing mission success. We hope to demonstrate that consideration given to both physical, cyber, and mission objectives can yield more well-rounded, efficient results. We first examine and analyze simulated results obtained by optimizing the cost function in Equation (A.12) to gain insight into the tradeoffs from competing objectives. We optimize over the physical system alone, the cyber system alone, and both together by selecting appropriate weights $\beta_{p1}, \beta_{p2}, \beta_{c1}$, and β_{c2}. Then we compare real-world performance of the three different optimization techniques as indicated by TableSat mission data.

A.5.1 Simulated Results

We solve the optimization problem in Equation (A.13) using numerical methods. We use MATLAB's solver `fmincon` and setting lower and upper bounds on the design parameters. This MATLAB function uses active-set optimization which utilizes sequential quadratic programming (SQP) and estimates the Hessian of the Lagrangian using the well-known BFGS algorithm [164].

Often there are auxiliary reasons for favoring one cost term over another such as length of time since the last mission, or a cloudy day resulting in poorer image quality. Since we wish to investigate the comparison of holistic CPS optimization with independent physical

and cyber system optimization we allow corresponding weights to go to zero as indicated in the 1^{st} and 2^{nd} rows of Table A.1. In the 3^{rd} row we use some *a priori* intuition to select weights that focus slightly more on time and less on cyber utilization. The analytical results of our optimization scheme and simulation are in Table A.1. These results suggest that the lowest cost solution comes by focusing on cyber parameters. This indicates there may be a problem with the weights that we are applying to the cost function. We note the specific weighting we used that slightly favors time over cyber utilization, which means we won't collect as many orange pixels. Despite these analytical results, in the next section we show experimental results that indeed validate this design methodology.

A.5.2 Experimental Results

We present in Table A.2 the results from our experiments with TableSat. As we had hoped, the lowest cost solution was indeed the total system optimization that accounts for cyber, physical, and mission objectives.

A.6 Conclusions and Future Work

The impact of the digital revolution will continue to reverberate across many fields for some time. As cyber and physical systems become more tightly integrated, the need for strong multidisciplinary co-design techniques becomes increasingly urgent. In this paper, we've implemented one such proposed optimization scheme - a cost function combining physical and cyber terms - on real hardware. We used a TableSat with a mounted camera to undertake a mission to survey orange segments of an orange and black paper landscape. We experimentally determined system-specific parameters for our equations and used the resulting optimization problem to produce an optimal angular velocity and task rate. We were then able to use these results to configure a real mission, and compare real-world performance between the traditional optimization method and the cost-function method.

Our results showed that our multi-disciplinary approach indeed resulted in better efficiency than optimizing the physical terms without considering the cyber term (or optimizing the cyber term without considering the physical term). This supports the theoretical results found by previous work.

However, these results have limited statistical significance, as they are results from only one run for each configuration. The TableSat platform was very noisy and had extremely high variance in steady state speed. We tried to calibrate it before every run such that the behavior roughly matched the experimental results we obtained earlier when mapping

Table A.1: Comparison of Simulated Results

PARAMETERS	SOLUTION TYPE	E_p	T	U_τ	M	Total	RATIO TO BEST
$\omega = 5.2\,\text{rads/s}$, $r_\tau = 2\,\text{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.5, \beta_{c1} = \beta_{c2} = 0.0$	2.87 J	1.21 s	0.25	2.21×10^{-5}	1.0963	100.8%
$\omega = 0.95\,\text{rads/s}$, $r_\tau = 3\,\text{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.0, \beta_{c1} = \beta_{c2} = 0.5$	1.30 J	6.61 s	0.38	2.70×10^{-6}	0.5457	N/A
$\omega = 2.42\,\text{rads/s}$, $r_\tau = 2\,\text{Hz}$	J with $\beta_{p1} = 1, \beta_{p2} = 1.2, \beta_{c1} = 0.8, \beta_{c2} = 1$	2.47 J	2.60 s	0.25	1.03×10^{-6}	0.7179	31.55%

Table A.2: Comparison of Experimental Results

PARAMETERS	SOLUTION TYPE	E_p	T	U_τ	M	TOTAL	RATIO TO BEST
$\omega = 5.2\,\text{rads/s}$, $r_\tau = 2\,\text{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.5, \beta_{c1} = \beta_{c2} = 0.0$	2.88 J	1.21 s	0.25	2.75×10^{-6}	0.4483	9.09%
$\omega = 0.95\,\text{rads/s}$, $r_\tau = 3\,\text{Hz}$	J with $\beta_{p1} = \beta_{p2} = 0.0, \beta_{c1} = \beta_{c2} = 0.5$	1.29 J	6.61 s	0.38	2.36×10^{-7}	0.4635	12.79%
$\omega = 2.42\,\text{rads/s}$, $r_\tau = 2\,\text{Hz}$	J with $\beta_{p1} = 1, \beta_{p2} = 1.2, \beta_{c1} = 0.8, \beta_{c2} = 1$	2.39 J	2.52 s	0.25	1.392×10^{-6}	0.4109	N/A

PWM signals to angular velocity, but this was time-consuming and statistically unsatisfying. Another result of this is that we were not able to do a parameter sweep of cost-function weights. Our information metric is also not very sophisticated and may not be representative of all the complexities present in many real applications.

This work highlighted many of the difficulties associated with working with physical systems, one of which is that some physical phenomena are extremely sensitive to initial conditions. At a macro-level this looks like inconsistency (e.g. varying steady-state speeds). To be able to use simple equations that cannot account for all possible disturbances, the system must display consistent, stable states at whatever scale the system is modeled. Then, control theory is applied to reject (most) disturbances usually via feedback. For the scope of this project, however, we used open-loop control, which made data collection difficult. For those reasons, further work on this particular application would require writing a closed-loop controller for the TableSat platform. This would also allow us to do a broad parameter sweep of our cost function.

Though further work in this area presents challenges, the effort may be well worth it. As cyber-physical systems become a more and more ubiquitous presence in our lives, we must embrace the integration of previously disparate paradigms and begin working towards a similarly integrated multidisciplinary approach.

APPENDIX B

Comprehensive CubeSat Attitude Control Plots

In this appendix we show more comprehensive (i.e. all states) plots for the CubeSat simulations in Chapter 4. The states indicated in each subplot correspond to the state description in Equation (4.3) which we also show here

$$\mathbf{x}_p = \left(\theta_1, \theta_2, \theta_3, \omega_1, \omega_2, \omega_3, H_1^w, H_2^w, H_3^w\right)$$
$$\mathbf{u}_p = (M_1, M_2, M_3)\,.$$

To recapitulate, states (1, 2, 3) are roll, pitch, and yaw in the LVLH reference frame, states (4, 5, 6) are elements of the angular velocity vector, and states (7, 8, 9) represent angular momentum of each of three reaction microwheels used in control. The control vector is composed of torques on the spacecraft resulting from the reaction microwheels. The simulation setup is described in Section 4.5.

In the plots that follow all axes have been fixed to facilitate comparison between plots. We have ordered the figures following the order in Table 4.3.

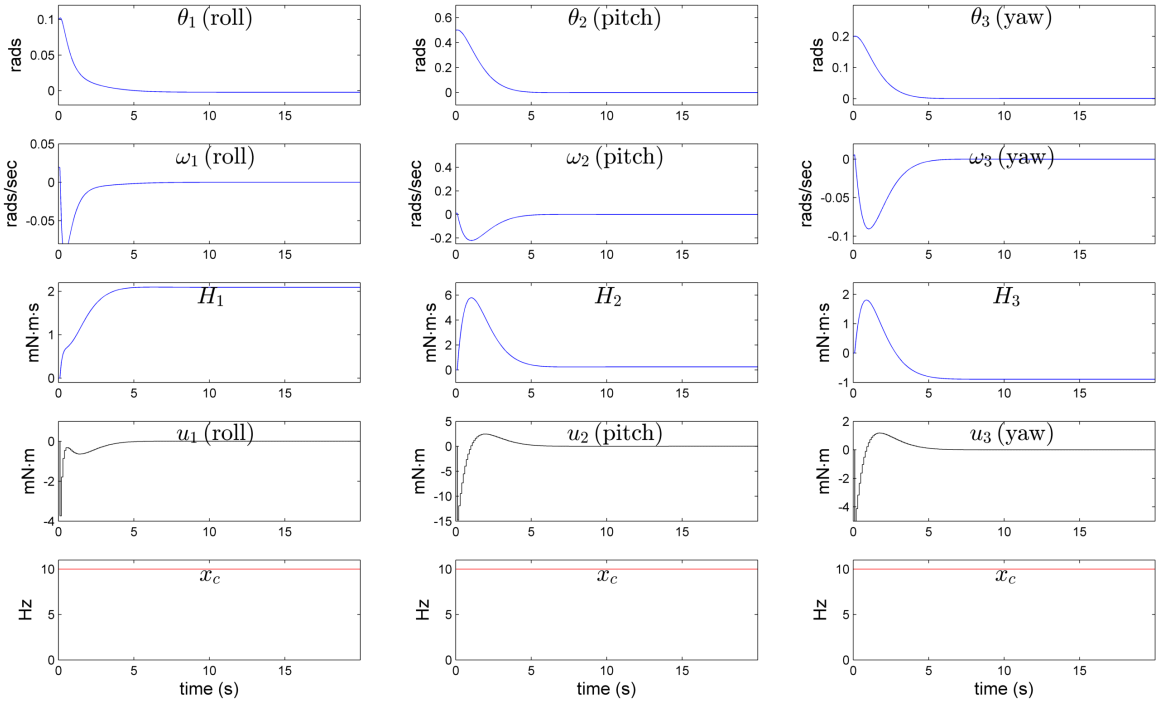


Figure B.1: CubeSat Attitude Disturbance Response Using Fixed 10 Hz DLQR Control

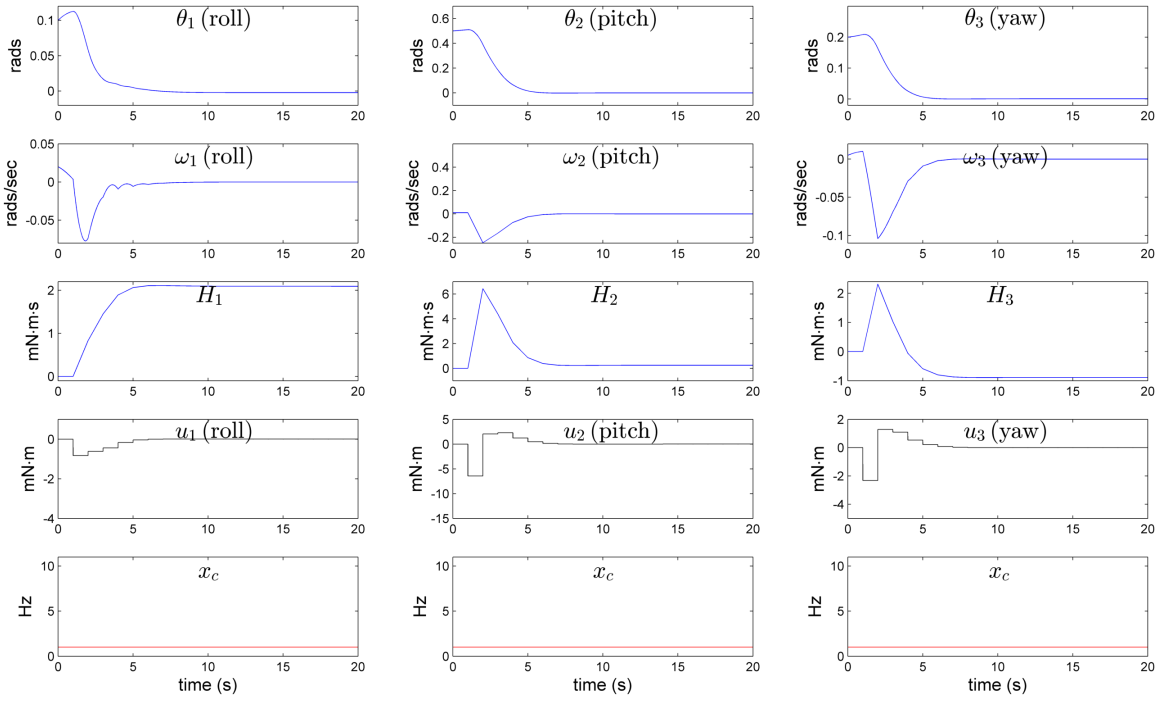


Figure B.2: CubeSat Attitude Disturbance Response Using Fixed 1 Hz DLQR Control

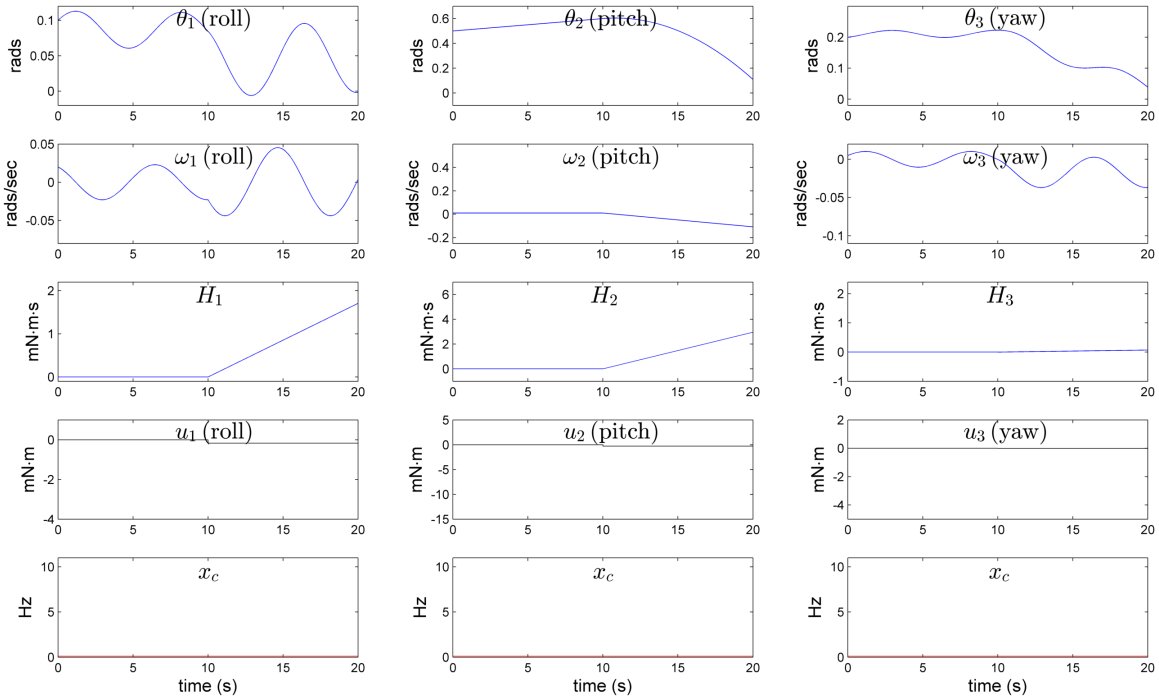


Figure B.3: CubeSat Attitude Disturbance Response Using Fixed 0.1 Hz DLQR Control

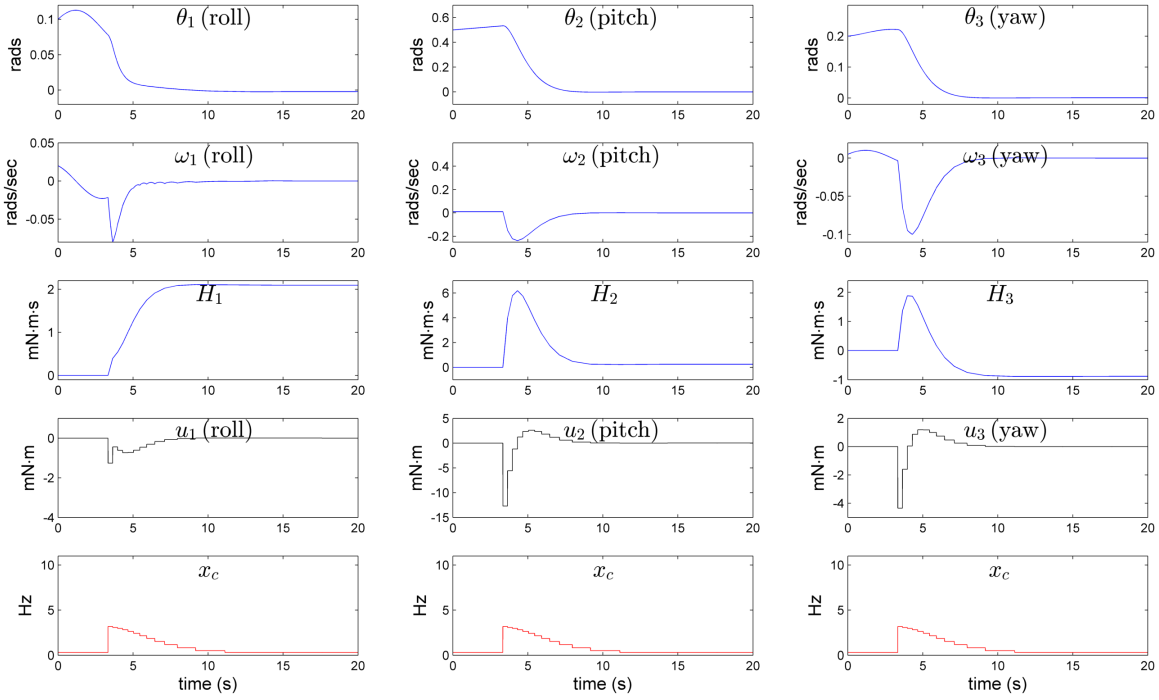


Figure B.4: CubeSat Attitude Disturbance Response Using GSDLQR Control Using $u_{c,1}$

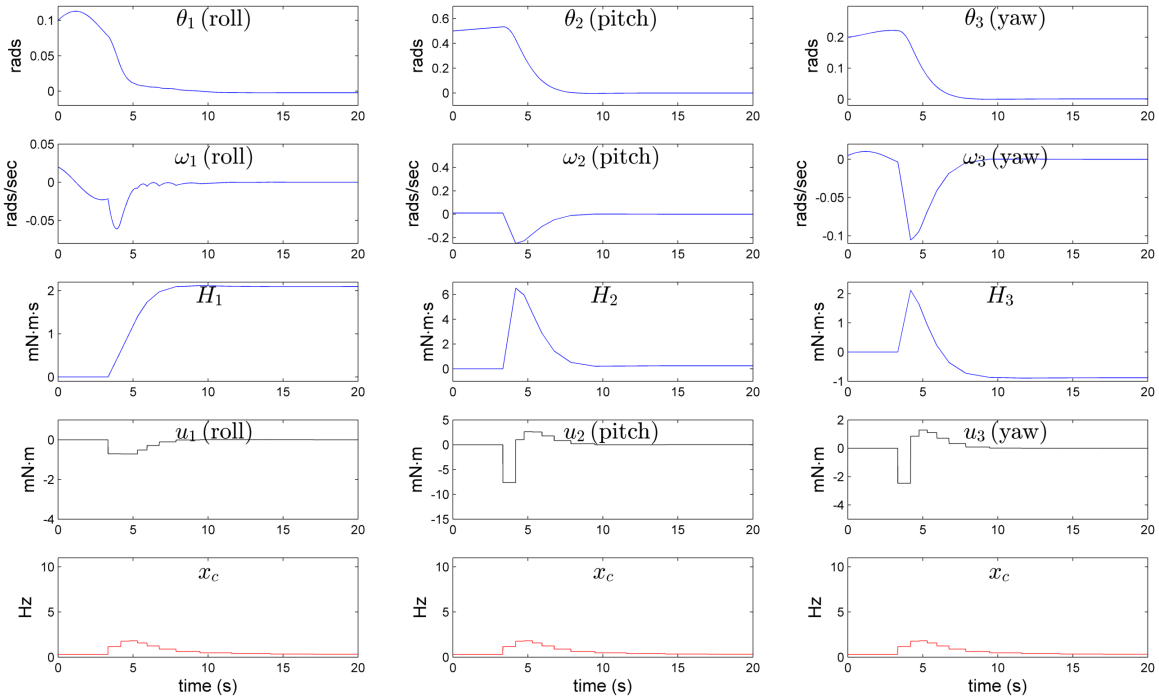


Figure B.5: CubeSat Attitude Disturbance Response Using GSDLQR Control Using $u_{c,2}$

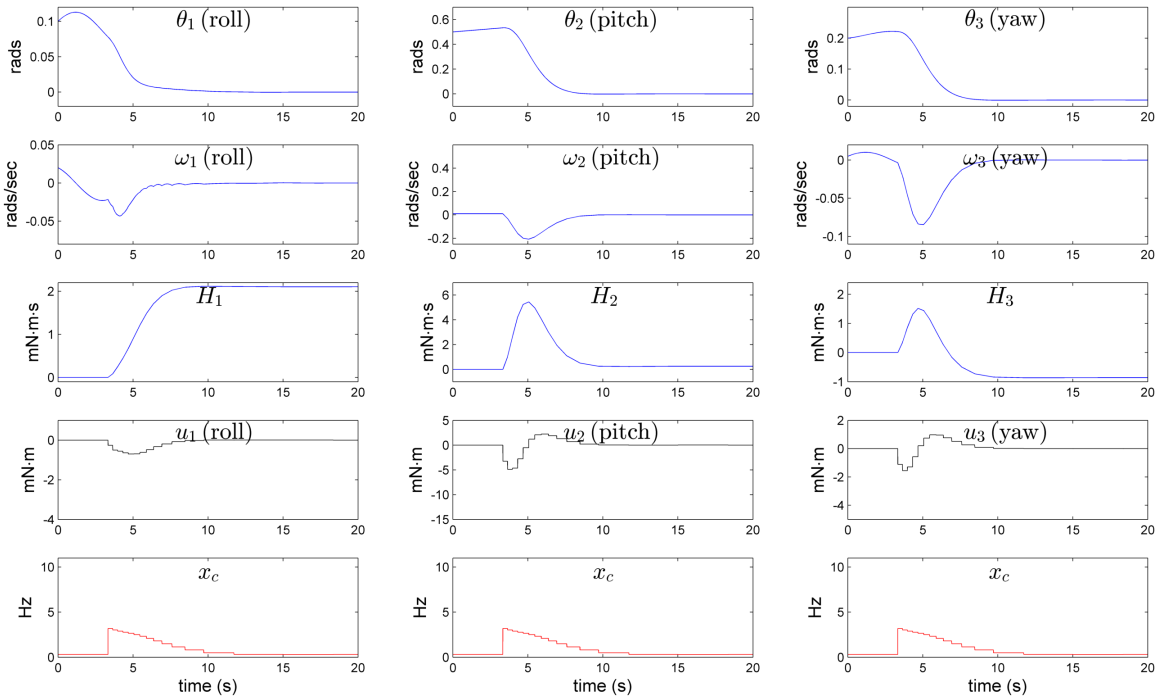


Figure B.6: CubeSat Attitude Disturbance Response Using FPRB Control Using $u_{c,1}$

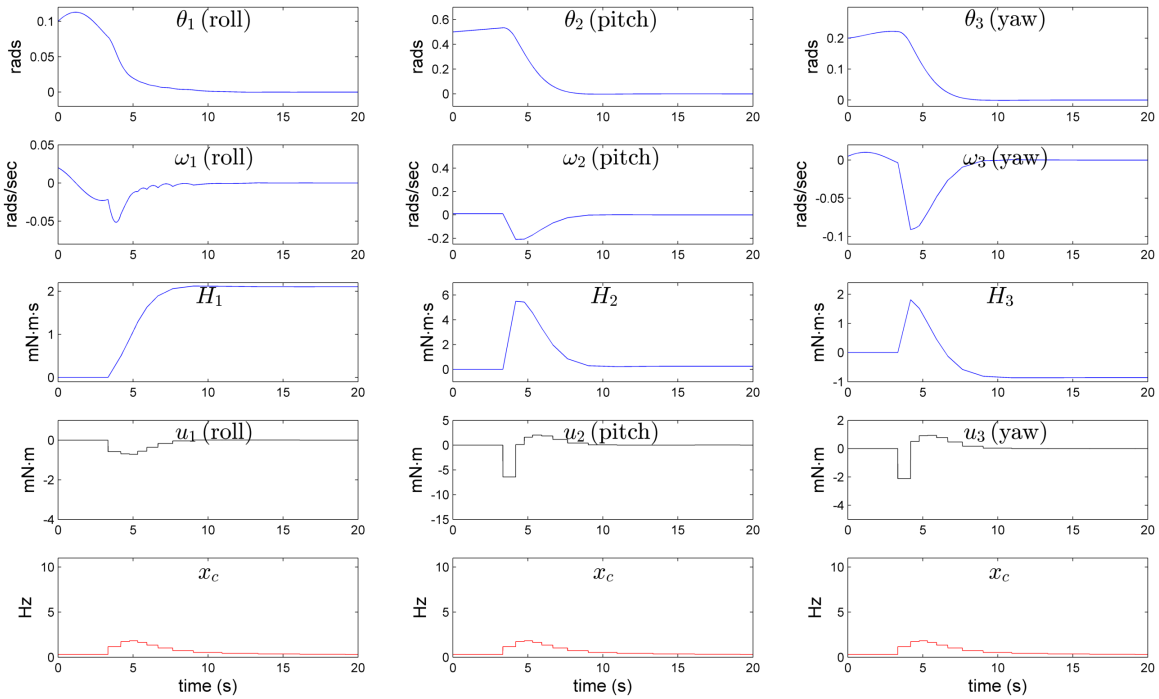


Figure B.7: CubeSat Attitude Disturbance Response Using FPRB Control Using $u_{c,2}$

BIBLIOGRAPHY

[1] Kim, K.-D. and Kumar, P., "Cyber–Physical Systems: A Perspective at the Centennial," *Proceedings of the IEEE*, Vol. 100, No. 13, 2012, pp. 1287–1308.

[2] Energetics Incorporated, "Foundations for Innovation in Cyber-Physical Systems: Workshop Report," Tech. rep., National Institute of Standards and Technology, January 2013.

[3] (NSF), N. S. F., "Cyber-Physical Systems," [Accessed 31 January 2013].

[4] PCAST 2012, "Report to the President on Capturing Domestic Competitive Advantage in Advanced Manufacturing," Tech. rep., President's Council of Advisors on Science and Technology (PCAST), Executive Office of the President, July 2012.

[5] PCAST 2011, "Ensuring American Leadership in Advanced Manufacturing," Tech. rep., President's Council of Advisors on Science and Technology (PCAST), Executive Office of the President, June 2011.

[6] NITRD 2009, "High-Confidence Medical Devices: Cyber-Physical Systems for the 21st Century Health Care," Tech. rep., Networking and Information Technology Research and Development (NITRD) Program, February 2009.

[7] PCAST 2013, "Report to the President Immediate Opportunities for Strengthening the Nation's Cybersecurity," Tech. rep., President's Council of Advisors on Science and Technology (PCAST), Executive Office of the President, November 2013.

[8] Kleissl, J. and Agarwal, Y., "Cyber-physical energy systems: focus on smart buildings," *Proceedings of the 47th Design Automation Conference*, ACM, 2010, pp. 749–754.

[9] Lee, I. and Sokolsky, O., "Medical cyber physical systems," *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, IEEE, 2010, pp. 743–748.

[10] Karnouskos, S., "Cyber-physical systems in the smartgrid," *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, IEEE, 2011, pp. 20–23.

[11] Mo, Y., Kim, T.-H., Brancik, K., Dickinson, D., Lee, H., Perrig, A., and Sinopoli, B., "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, Vol. 100, No. 1, 2012, pp. 195–209.

[12] Lee, E. A., "Cyber physical systems: Design challenges," *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, IEEE, 2008, pp. 363–369.

[13] Bini, E. and Buttazzo, G., "The Optimal Sampling Pattern for Linear Control Systems," *Automatic Control, IEEE Transactions on*, Vol. 59, No. 1, Jan 2014, pp. 78–90.

[14] Sztipanovits, J., Ying, S., Cohen, I., Corman, D., Davis, J., Khurana, H., Mosterman, P. J., Prasad, V., and Stormo, L., "STEERING COMMITTEE FOR FOUNDATIONS FOR INNOVATION IN CYBER-PHYSICAL SYSTEMS," .

[15] Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., Goodwine, B., Baras, J., and Wang, S., "Toward a Science of Cyber-Physical System Integration," *Proceedings of the IEEE*, Vol. 100, No. 1, 2012, pp. 29–44.

[16] Poovendran, R., "Cyber–Physical Systems: Close Encounters Between Two Parallel Worlds," *Proc. of the IEEE*, Vol. 98, No. 8, 2010, pp. 1363–1366.

[17] Aström, K. J. and Bernhardsson, B. M., "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, Vol. 2, IEEE, 2002, pp. 2011–2016.

[18] Velasco, M., Martí, P., and Bini, E., "Control-driven Tasks: Modeling and Analysis," *Real-Time Systems Symposium, 2008*, IEEE, 2008, pp. 280–290.

[19] Lunze, J. and Lehmann, D., "A state-feedback approach to event-based control," *Automatica*, Vol. 46, No. 1, 2010, pp. 211–215.

[20] Heemels, W., Sandee, J., and Van Den Bosch, P., "Analysis of event-driven controllers for linear systems," *International journal of control*, Vol. 81, No. 4, 2008, pp. 571–590.

[21] Lemmon, M., Chantem, T., Hu, X. S., and Zyskowski, M., "On self-triggered Full-Information H-Infinity Controllers," *Hybrid Systems: computation and control*, Springer, 2007, pp. 371–384.

[22] Tabuada, P., "Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks," *Automatic Control, IEEE Transactions on*, Vol. 52, No. 9, 2007, pp. 1680–1685.

[23] Voit, H., Annaswamy, A., Schneider, R., Goswami, D., and Chakraborty, S., "Adaptive switching controllers for systems with hybrid communication protocols," *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 4921–4926.

[24] Cambone, S., Krieg, K., Pace, P., and Linton, W., "Unmanned Aircraft Systems Roadmap 2005-2030," *Office of the Secretary of Defense*, August 2005.

[25] Andersson, J., "A survey of multiobjective optimization in engineering design," *Department of Mechanical Engineering, Linktjping University. Sweden*, 2000.

[26] Bradley, J. M. and Atkins, E. M., "Toward Continuous State-Space Regulation of Coupled Cyber-Physical Systems," *Proceedings of the IEEE*, Vol. 100, No. 1, January 2012, pp. 60–74.

[27] Bradley, J. M. and Atkins, E. M., "Coupled Cyber-Physical System Modeling and Control of a Small Satellite," *Under Review*, 2014.

[28] Bradley, J. M. and Atkins, E. M., "A Cyber-Physical Optimization Approach to Mission Success for Unmanned Aircraft Systems," *Journal of Aerospace Information Systems*, Vol. 11, No. 1, January 2014, pp. 48–60.

[29] Wiener, N., *Cybernetics or Control and Communication in the Animal and the Machine*, Vol. 25, MIT press, 1965.

[30] Wikipedia, "Internet-Related Prefixes," http://en.wikipedia.org/wiki/Internet-related_prefixes, July 2014.

[31] Denyer, N., *Plato: Alcibiades*, Cambridge University Press, 2001.

[32] Abbate, J. E., "From ARPANET to Internet: A history of ARPA-sponsored computer networks, 1966–1988," 1994.

[33] Atzori, L., Iera, A., and Morabito, G., "The internet of things: A survey," *Computer networks*, Vol. 54, No. 15, 2010, pp. 2787–2805.

[34] Farley, T., "Mobile telephone history," *Telektronikk*, Vol. 101, No. 3/4, 2005, pp. 22.

[35] Crow, B. P., Widjaja, I., Kim, J. G., and Sakai, P. T., "IEEE 802.11 wireless local area networks," *Communications Magazine, IEEE*, Vol. 35, No. 9, 1997, pp. 116–126.

[36] Bennett, S., "A brief history of automatic control," *IEEE Control Systems Magazine*, Vol. 16, No. 3, 1996, pp. 17–25.

[37] Evans, W. R., "Control system synthesis by root locus method," *American Institute of Electrical Engineers, Transactions of the*, Vol. 69, No. 1, 1950, pp. 66–69.

[38] Nyquist, H., "Regeneration theory," *Bell System Technical Journal*, Vol. 11, No. 1, 1932, pp. 126–147.

[39] Bode, H. W., "Network analysis and feedback amplifier design," 1945.

[40] Ziegler, J. and Nichols, N., "Optimum settings for automatic controllers," *trans. ASME*, Vol. 64, No. 11, 1942.

[41] Bennett, S., "Control and the digital computer: the early years," *Measurement and Control*, Vol. 37, No. 10, 2004, pp. 307–311.

[42] Bellman, R., "Dynamic programming and Lagrange multipliers," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 42, No. 10, 1956, pp. 767.

[43] Pontryagin, L. S., *Mathematical theory of optimal processes*, CRC Press, 1987.

[44] Kalman, R., "On the general theory of control systems," *IRE Transactions on Automatic Control*, Vol. 4, No. 3, 1959, pp. 110–110.

[45] Gilbert, E. G., "Controllability and observability in multivariable control systems," *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, Vol. 1, No. 2, 1963, pp. 128–151.

[46] Zhou, K., Doyle, J. C., Glover, K., et al., *Robust and optimal control*, Vol. 272, Prentice Hall New Jersey, 1996.

[47] Khalil, H. K., *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 3rd ed., 2001.

[48] Aström, K. J., *Introduction to stochastic control theory*, Courier Dover Publications, 2012.

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

[50] Cassandras, C. G. and Lafortune, S., *Introduction to discrete event systems*, Springer, 2008.

[51] Alur, R., Courcoubetis, C., Henzinger, T. A., and Ho, P.-H., *Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems*, Springer, 1993.

[52] Jury, E. and Tsypkin, Y., "On the Theory of Discrete Systems," *Automatica*, Vol. 7, No. 1, 1971, pp. 89–107.

[53] Jury, E. I., "Sampled-Data Systems, Revisited- Reflections, Recollections and Reassessments," *Joint Automatic Control Conference, San Francisco, CA*, 1980.

[54] Shin, K. G. and Ramanathan, P., "Real-time computing: A new discipline of computer science and engineering," *Proceedings of the IEEE*, Vol. 82, No. 1, 1994, pp. 6–24.

[55] Sha, L., Abdelzaher, T., Årzén, K., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., and Mok, A., "Real-time Scheduling Theory: A Historical Perspective," *Real-time Systems*, Vol. 28, No. 2, 2004, pp. 101–155.

[56] Schaller, R. R., "Moore's law: past, present and future," *Spectrum, IEEE*, Vol. 34, No. 6, 1997, pp. 52–59.

[57] Zweigle, O., van de Molengraft, R., d'Andrea, R., and Häussermann, K., "RoboEarth: connecting robots worldwide," *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ACM, 2009, pp. 184–191.

[58] Krishna, C. M. and Shin, K. G., *Real-time Systems*, Tata McGraw-Hill Education, 1997.

[59] Liu, C. and Layland, J., "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, Vol. 20, No. 1, 1973, pp. 46–61.

[60] Franklin, G., Workman, M., and Powell, D., *Digital Control of Dynamic Systems*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1998.

[61] Hespanha, J., Naghshtabrizi, P., and Xu, Y., "A survey of recent results in networked control systems," *Proc. of the IEEE*, Vol. 95, No. 1, 2007, pp. 138–162.

[62] Nilsson, J., *Real-time control systems with delays*, Ph.D. thesis, Ph. D. dissertation, Department of Automatic Control, Lund Institute of Technology, 1998.

[63] Fridman, E., Seuret, A., and Richard, J.-P., "Robust sampled-data stabilization of linear systems: an input delay approach," *Automatica*, Vol. 40, No. 8, 2004, pp. 1441–1446.

[64] Diop, S., Kolmanovsky, I., Moraal, P., and Van Nieuwstadt, M., "Preserving stability/performance when facing an unknown time-delay," *Control Engineering Practice*, Vol. 9, No. 12, 2001, pp. 1319–1325.

[65] Nilsson, J., Bernhardsson, B., and Wittenmark, B., "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, Vol. 34, No. 1, 1998, pp. 57–64.

[66] Richard, J., "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, Vol. 39, No. 10, 2003, pp. 1667–1694.

[67] Kingston, D. B. and Beard, R. W., "Real-Time Attitude and Position Estimation for Small UAVs Using Low-Cost Sensors," *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, sn, 2004, pp. 2004–6488.

[68] Zhang, F., Szwaykowska, K., Wolf, W., and Mooney, V., "Task Scheduling for Control Oriented Requirements for Cyber-Physical Systems," *Real-Time Systems Symposium, 2008*, IEEE, 2008, pp. 47–56.

[69] Krasovsky, N. N., *Stability of Motion: Applications of Lyapunov's Second Method to Differential Systems and Equations with Delay*, Stanford University Press, 1963.

[70] Kolmanovsky, V. B., *Stability of functional differential equations*, Academic press, 1986.

[71] Peet, M. M., *Stability and Control of Functional Differential Equations*, Ph.D. thesis, Stanford University, 2006.

[72] Dugard, L. and Verriest, E., *Stability and Control of Time-Delay Systems*, Springer, 1998.

[73] Gu, K., Kharitonov, V. L., and Chen, J., *Stability of Time-Delay Systems*, Birkhauser, 2003.

[74] Xia, Y., Fu, M., and Shi, P., *Analysis and Synthesis of Dynamical Systems with Time-delays*, Springer Verlag, 2009.

[75] Stépán, G., *Retarded Dynamical Systems: Stability and Characteristic Functions*, Longman Scientific & Technical ; Wiley, Burnt Mill, Harlow, Essex, England : New York, 1989.

[76] Glasson, D. P., "A new technique for multirate digital control design and sample rate selection," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 4, 1982, pp. 379–382.

[77] Powell, J. D. and Katz, P., "Sample rate selection for aircraft digital control," *AIAA Journal*, Vol. 13, No. 8, 1975, pp. 975–979.

[78] Gopal, M., *Digital Control Engineering*, New Age International, 1988.

[79] Kuo, B. C., *Digital Control Systems*, Oxford University Press, 1992.

[80] Kim, J., Lakshmanan, K., and Rajkumar, R. R., "Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems," *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, IEEE Computer Society, 2012, pp. 55–64.

[81] Fontanelli, D., Greco, L., and Bicchi, A., "Anytime Control Algorithms for Embedded Real-time Systems," *Hybrid Systems: computation and control*, 2008, pp. 158–171.

[82] Bhattacharya, R. and Balas, G., "Anytime Control Algorithm: Model Reduction Approach," *Journal of Guidance, Control, & Dynamics*, Vol. 27, No. 5, 2004.

[83] Gupta, V., "On an anytime algorithm for control," *Proceedings of the 48th IEEE Conference on Decision and Control*, IEEE, 2010, pp. 6218–6223.

[84] Arzén, K., Cervin, A., Eker, J., and Sha, L., "An introduction to control and scheduling co-design," *Proc. of the 39th IEEE Conf. on Decision and Control*, Vol. 5, IEEE, 2002, pp. 4865–4870.

[85] Cervin, A., Eker, J., Bernhardsson, B., and Årzén, K., "Feedback–feedforward scheduling of control tasks," *Real-Time Systems*, Vol. 23, No. 1, 2002, pp. 25–53.

[86] Eker, J., Hagander, P., and Årzén, K., "A Feedback Scheduler for Real-Time Control Tasks," *Control Engineering Practice*, Vol. 8, No. 12, 2000, pp. 1369–1378.

[87] Branicky, M., Phillips, S., and Zhang, W., "Scheduling and Feedback Co-Design for Networked Control Systems," *IEEE Transactions on Automatic Control*, Vol. 2, 2002, pp. 1211–1217.

[88] Zhang, W., Branicky, M., and Phillips, S., "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, Vol. 21, No. 1, 2001, pp. 84–99.

[89] Tomlin, C., Lygeros, J., and Sastry, S., "Synthesizing controllers for nonlinear hybrid systems," *Hybrid Systems: Computation and Control*, 1998, pp. 360–373.

[90] Nerode, A. and Kohn, W., "Models for Hybrid Systems: Automata, Topologies, Controllability, Observability," *Hybrid Systems*, 1993, pp. 317–356.

[91] Branicky, M. S., "Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems," *Automatic Control, IEEE Transactions on*, Vol. 43, No. 4, 1998, pp. 475–482.

[92] Branicky, M., Borkar, V., and Mitter, S., "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. on Automatic Control*, Vol. 43, No. 1, 1998, pp. 31–45.

[93] Sala, A., "Computer control under time-varying sampling period: An LMI gridding approach," *Automatica*, Vol. 41, No. 12, 2005, pp. 2077–2082.

[94] Schinkel, M., Chen, W.-H., and Rantzer, A., "Optimal control for systems with varying sampling rate," *American Control Conference, 2002. Proceedings of the 2002*, Vol. 4, IEEE, 2002, pp. 2979–2984.

[95] Gupta, V., Chung, T. H., Hassibi, B., and Murray, R. M., "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, Vol. 42, No. 2, 2006, pp. 251–260.

[96] Savkin, A. V., Evans, R. J., and Skafidas, E., "The problem of optimal robust sensor scheduling," *Systems & Control Letters*, Vol. 43, No. 2, 2001, pp. 149–157.

[97] He, Y. and Chong, E. K., "Sensor scheduling for target tracking in sensor networks," *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, Vol. 1, IEEE, 2004, pp. 743–748.

[98] Krishnamurthy, V., "Algorithms for optimal scheduling and management of hidden Markov model sensors," *Signal Processing, IEEE Transactions on*, Vol. 50, No. 6, 2002, pp. 1382–1397.

[99] Evans, J. and Krishnamurthy, V., "Optimal sensor scheduling for hidden Markov model state estimation," *International Journal of Control*, Vol. 74, No. 18, 2001, pp. 1737–1742.

[100] Kowalska, K. and Mohrenschildt, M., "An approach to variable time receding horizon control," *Optimal Control Applications and Methods*, Vol. 33, No. 4, 2012, pp. 401–414.

[101] Harrison, S., Price, M., and Philpott, M., "Task scheduling for satellite based imagery," *Proceedings of the Eighteenth Workshop of the UK Planning and Scheduling Special Interest Group*, Vol. 78, University of Salford, UK, 1999, pp. 64–78.

[102] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Lee, R., Mandl, D., Frye, S., Trout, B., Hengemihle, J., D'Agostino, J., Shulman, S., Ungar, S., Brakke, T., Boyer, D., Gaasbeck, J. V., Greeley, R., Doggett, T., Baker, V., Dohm, J., and Ip, F., "The EO-1 Autonomous Science Agent," *AAMAS '04 Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, Vol. 1, IEEE Computer Society, IEEE Computer Society, Washington, DC, 2004, pp. 420–427.

[103] Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A., "Iterative repair planning for spacecraft operations using the ASPEN system," *Artificial Intelligence, Robotics and Automation in Space*, Vol. 440, 1999, p. 99.

[104] Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C., "Remote agent: To boldly go where no AI system has gone before," *Artificial Intelligence*, Vol. 103, No. 1, 1998, pp. 5–47.

[105] Muscettola, N., "HSTS: Integrating planning and scheduling," Tech. rep., DTIC Document, 1993.

[106] Bataille, N., Lemaitre, M., and Verfaillie, G., "Efficiency and Fairness when Sharing the Use of a Satellite," *Artificial Intelligence, Robotics and Automation in Space*, Vol. 440, European Space Agency, Paris, 1999, p. 465.

[107] Bresina, J. L., Morris, R. A., and Edgington, W. R., "Optimizing Observation Scheduling Objectives," Tech. rep., Proceedings of the 1997 NASA Wokshop on Planning and Scheduling for Space, Oxnard, CA, Oct 1997.

[108] Agrawal, M., Cofer, D., and Samad, T., "Real-time adaptive resource management for advanced avionics," *Control Systems, IEEE*, Vol. 23, No. 1, feb 2003, pp. 76 – 86.

[109] Atkins, E. M. and Sanner, R. M., "QoS Tradeoffs for Guidance, Navigation, and Control," *2002 IEEE Aerospace Conference Proceedings*, Vol. 7, IEEE, Big Sky, Montana, March 2002, pp. 7–3333–7–3341 vol.7.

[110] Abdelzaher, T. F., Atkins, E., and Shin, K., "QoS negotiation in real-time systems and its application to automated flight control," *Computers, IEEE Transactions on*, Vol. 49, No. 11, 2000, pp. 1170–1183.

[111] Russ, M. and Stütz, P., "Airborne sensor and perception management: A conceptual approach for surveillance UAS," *Information Fusion (FUSION), 2012 15th International Conference on*, IEEE, IEEE, Singapore, July 2012, pp. 2444 –2451.

[112] Narayan, P., Campbell, D., and Walker, R., "Computationally adaptive multiobjective trajectory optimization for UAS with variable planning deadlines," *Aerospace conference, 2009 IEEE*, march 2009, pp. 1 –8.

[113] Kirk, D., *Optimal control theory: an introduction*, Dover Publications, 2004.

[114] Aström, K. and Wittenmark, B., *Computer-controlled systems: theory and design*, Prentice-Hall New York, 1984.

[115] Bradley, J. M. and Atkins, E. M., "Computational-Physical State Co-Regulation in Cyber-Physical Systems," *ACM/IEEE Conference on Cyber-Physical Systems*, ACM/IEEE, Chicago, IL, April 2011.

[116] Chen, M. and Kao, C., "Control of linear time-varying systems using forward Riccati equation," *Journal of dynamic systems, measurement, and control*, Vol. 119, No. 3, 1997, pp. 536–540.

[117] Weiss, A., Kolmanovsky, I., and Bernstein, D. S., "Forward-integration Riccati-based output-feedback control of linear time-varying systems," *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 6708–6714.

[118] Weiss, A., Kolmanovsky, I. V., Baldwin, M., Erwin, R. S., and Bernstein, D. S., "Forward-integration Riccati-based feedback control for spacecraft rendezvous maneuvers on elliptic orbits," *CDC*, 2012, pp. 1752–1757.

[119] Prach, A., Tekinalp, O., and Bernstein, D. S., "A Numerical Comparison of Frozen-Time and Forward-Propagating Riccati Equations for Stabilization of Periodically Time-Varying Systems," *American Control Conference*, Portland, OR, June 2014.

[120] Heidt, H., Puig-Suari, J., Moore, A., Nakasuka, S., and Twiggs, R., "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," *Small Satellite Conference*, 2000.

[121] Hughes, P. C., *Spacecraft Attitude Dynamics*, DoverPublications.com, 2012.

[122] Osburn, S. L. and Bernstein, D. S., "An Exact Treatment of the Achievable Closed-loop H_2 Performance of Sampled-data Controllers: From Continuous-time to Open-loop," *Automatica*, Vol. 31, No. 4, 1995, pp. 617–620.

[123] Khosla, P. K., "Choosing sampling rates for robot control," *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, Vol. 4, IEEE, 1987, pp. 169–174.

[124] Lappas, V., Adeli, N., Visagie, L., Fernandez, J., Theodorou, T., Steyn, W., and Perren, M., "CubeSail: A low cost CubeSat based solar sail demonstration mission," *Advances in Space Research*, Vol. 48, No. 11, 2011, pp. 1890–1901.

[125] Rugh, W. J., "Analytical framework for gain scheduling," *Control Systems, IEEE*, Vol. 11, No. 1, 1991, pp. 79–84.

[126] Aström, K. J., "Theory and Applications of Adaptive Control – A Survey," *Automatica*, Vol. 19, No. 5, 1983, pp. 471–486.

[127] Lewis, F. L., Vrabie, D., and Syrmos, V. L., *Optimal control*, John Wiley & Sons, 2012.

[128] Kessler, D. J., Reynolds, R. C., and Anz-Meador, P. D., "Orbital debris environment for spacecraft designed to operate in low Earth orbit," Tech. rep., DTIC Document, 1989.

[129] Kessler, D. J., "Orbital debris environment for spacecraft in low earth orbit," *Journal of spacecraft and rockets*, Vol. 28, No. 3, 1991, pp. 347–351.

[130] Murr, L. E. and Kinard, W. H., "Effects of low earth orbit," *American Scientist*, 1993, pp. 152–165.

[131] Garrett, H. B., "Space Environments and Survivability," *The International Handbook of Space Technology*, Springer, 2014, pp. 37–59.

[132] Svedhem, H., Drolshagen, G., Grün, E., Grafodatsky, O., and Prokopiev, U., "New results from in situ measurements of Cosmic Dust - Data from the GORID experiment," *Advances in Space Research*, Vol. 25, No. 2, 2000, pp. 309–314.

[133] Blackwell, W., Allen, G., Galbraith, C., Hancock, T., Leslie, R., Osaretin, I., Retherford, L., Scarito, M., Semisch, C., Shields, M., et al., "Nanosatellites for earth environmental monitoring: The MicroMAS project," *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, IEEE, 2012, pp. 206–209.

[134] Selva, D. and Krejci, D., "A survey and assessment of the capabilities of Cubesats for Earth observation," *Acta Astronautica*, Vol. 74, 2012, pp. 50–68.

[135] Cutler, J. W. and Bahcivan, H., "Radio Aurora Explorer: A Mission Overview," *Journal of Spacecraft and Rockets*, Vol. 51, 2014, pp. 1–9.

[136] Kayal, H., Baumann, F., Briess, K., and Montenegro, S., "Beesat: A pico satellite for the on orbit verification of micro wheels," *Recent Advances in Space Technologies, 2007. RAST'07. 3rd International Conference on*, IEEE, 2007, pp. 497–502.

[137] Tsiotras, P., Shen, H., and Hall, C., "Satellite attitude control and power tracking with energy/momentum wheels," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 23–34.

[138] Falbel, G., Puig-Suari, J., and Peczalski, A., "Sun oriented and powered, 3 axis and spin stabilized cubesats," *Aerospace Conference Proceedings, 2002. IEEE*, Vol. 1, IEEE, 2002, pp. 1–447.

[139] Clyde Space, "Small Satellite Reaction Wheels," http://www.clyde-space.com/products/reaction_wheels, May 2014.

[140] Microsat Systems Canada Inc., "MSCI MicroWheel 200," http://www.reactionwheel.com/products/MicroWheel-200.pdf, May 2014.

[141] Sinclair Interplanetary, "Microsatellite Reaction Wheels (-0.060-)," http://www.sinclairinterplanetary.com/reactionwheels/60%20mNm-sec%20wheel%202013b.pdf?attredirects=0, May 2014.

[142] Pingree, P. J., Bekker, D. L., Werne, T. A., and Wilson, T. O., "The prototype development phase of the CubeSat On-board processing Validation Experiment," *Aerospace Conference, 2011 IEEE*, IEEE, 2011, pp. 1–8.

[143] Schaffner, J., "The electronic system design, analysis, integration, and construction of the Cal Poly State University CP1 CubeSat," *Small Satellite Conference*, 2002.

[144] Asundi, S. A. and Fitz-Coy, N. G., "Design of command, data and telemetry handling system for a distributed computing architecture CubeSat," *Aerospace Conference, 2013 IEEE*, IEEE, 2013, pp. 1–14.

[145] Kranc, M. G., "Compensation of an Error-Sampled System by a Multirate Controller," *American Institute of Electrical Engineers, Part II: Applications and Industry, Transactions of the*, Vol. 76, No. 3, July 1957, pp. 149–159.

[146] Schaub, H. and Lappas, V. J., "Redundant reaction wheel torque distribution yielding instantaneous L2 power-optimal spacecraft attitude control," *Journal of guidance, control, and dynamics*, Vol. 32, No. 4, 2009, pp. 1269–1276.

[147] McClamroch, N., *Steady Aircraft Flight and Performance*, Princeton Univ Press, 2011.

[148] Zhan, P., Casbeer, D., and Swindlehurst, A., "A centralized control algorithm for target tracking with UAVs," *39th IEEE Asilomar Conference*, Vol. 13, 2005, p. 109.

[149] Sinha, A., Kirubarajan, T., and Bar-Shalom, Y., "Autonomous Ground Target Tracking by Multiple Cooperative UAVs," *Aerospace Conference, 2005 IEEE*, march 2005, pp. 1 –9.

[150] MacKay, D., *Information theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

[151] Shannon, C. E., "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Vol. 27, July, October 1948, pp. 379–423, 623–656.

[152] Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., and Suetens, P., "Multimodality image registration by maximization of mutual information," *Medical Imaging, IEEE Transactions on*, Vol. 16, No. 2, 1997, pp. 187–198.

[153] Gull, S. and Skilling, J., "Maximum entropy method in image processing," *Communications, Radar and Signal Processing, IEE Proceedings F*, Vol. 131, No. 6, 1984, pp. 646–659.

[154] Lai, J., *A hidden Markov model and relative entropy rate approach to vision-based dim target detection for UAV sense-and-avoid*, Ph.D. thesis, Queensland University of Technology, 2010.

[155] Ready, B., Taylor, C., and Beard, R., "A Kalman-filter based method for creation of super-resolved mosaicks," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, may 2006, pp. 3417 –3422.

[156] Bernstein, D. and Tsiotras, P., "A Course in Classical Optimal Control," *Unpublished notes, Version: June 15, 2009*, 2009.

[157] Panasonic, "Panasonic GP-CX161 Series Cameras," http://www.hicam.com/pana/Pana.pdf, June 2012.

[158] Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[159] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S., *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer Verlag, 2004.

[160] MathWorks, "MATLAB R2012a Documentation," http://www.mathworks.com/help/toolbox/optim/ug/fmincon.html, June 2012.

[161] Bradley, J. M., Clark, M. L., Atkins, E. M., and Shin, K. G., "Mission-Aware Cyber-Physical Optimization on a Tabletop Satellite," *AIAA Infotech@Aerospace*, Boston, MA, August 2013.

[162] Bradley, J. M. and Atkins, E. M., "Multi-Disciplinary Cyber-Physical Optimization for Unmanned Aircraft Systems," *AIAA Infotech@Aerospace*, Garden Grove, CA, June 2012.

[163] Vess, M. E. F., *System Modeling and Controller Design for a Single Degree of Freedom Spacecraft Simulator*, Master's thesis, University of Maryland, May 2005.

[164] Liu, D. and Nocedal, J., "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, Vol. 45, No. 1, 1989, pp. 503–528.