AUTONOMOUS SUSPENDED LOAD OPERATIONS VIA TRAJECTORY OPTIMIZATION AND VARIATIONAL INTEGRATORS

A Thesis
Presented to
The Academic Faculty

by

Gerardo De La Torre

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Guggenheim School of Aerospace Engineering

Georgia Institute of Technology
August 2015

AUTONOMOUS SUSPENDED LOAD OPERATIONS VIA TRAJECTORY OPTIMIZATION AND VARIATIONAL INTEGRATORS

Approved by:

Assoc. Professor Eric N. Johnson
Asst. Professor Evangelos Theodorou
Committee Chairs
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Assoc. Professor John-Paul Clarke
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Assoc. Professor Eric N. Johnson,
Co-Advisor
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Assoc. Professor Todd Murphey
Department of Mechanical Engineering
Northwestern University

Asst. Professor Evangelos Theodorou
Co-Advisor
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Christina. M. Ivler
Aviation Development Directorate
United States Army

Date Approved: 22 June 2015

ACKNOWLEDGEMENTS

"Georgia Tech is proud of its many traditions, but the one I find most exciting is our tradition of excellence. Our mission as students is NOT to follow in the footsteps of the astronauts, Nobel Prize Laureates, and presidents who graduated before us, but to EXCEED their footsteps! CRUSH the shoulders of the giants upon whom we stand! We here are all such innovative people, so I'm telling YOU if you want to change the world, YOU'RE AT GEORGIA TECH! YOU CAN DO THAT! IF YOU WANNA BUILD THE IRON MAN SUIT, YOU'RE AT GEORGIA TECH! YOU CAN DO THAT! IF YOU WANNA PLAY THEME MUSIC DURING YOUR CONVOCATION SPEECH, LIKE A BADASS, WERE AT GEORGIA TECH!!! WE CAN DO THAT! I AM DOING THAT! Congratulations on your acceptance. And brace yourselves for a hell of a ride on your way to becoming a hell of an engineer."

– Nick Selby, *Georgia Tech Freshman Convocation, 2013*

I would first like to thank my primary advisor Dr. Eric Johnson for his support. I will always be indebted to him for giving me the opportunity to work in the UAVRF. I would also like to thank everyone who help me along the way: Dr. Girish Chowdhary for being my first mentor at Georgia Tech and coaching me through my first research project, Dr. Tansel Yucelen for shaping the way in which I pursue research and teaching me how to craft a research paper, Dr. Evangelos Theodorou for becoming my second advisor and guiding me through the work I am presenting in this dissertation, Dr. John-Paul Clarke for his key insights in flight dynamics, optimization, and life, Dr. Todd Murphey for our discussions that dramatically shaped this dissertation, Dr. Christina Ivler for her support in my sponsored research and her suggestions that improved the quality of this dissertation, Dr. Wassim Haddad for sharing his deep knowledge on control engineering and travel, Dr. Panagiotis Tsiotras for challenging me during my first semester at Georgia Tech, Dr. Eric Feron for showing me that academic research should not always be so serious, Dr. Mark Costello for his precision while teaching (which I hope to have someday), Dr. Amy Pritchett for creating and

instructing the teaching practicum, Dr. Jeff Jagoda for his support throughout my stay, Dr. Stephen Ruffin for giving me the opportunity to apply to become a Sloan Fellow, Jorge Breton and the Goizueta fellows/scholars for the support I received from them, and everyone I worked with as a GoSTEM fellow for allowing me to be a part of such a great program. I will also like to thank everyone at the UAVRF for their help and support, especially the flight test crew, including: Jeong Hur, Henrik Christophersen, Claus Christmann, Daniel P. Magree, Lee Whitcher, Jack Mooney, Dmitry Bershadsky, Stephen Haviland, and Takuma Nakamura.

During my first semester at Georgia Tech, the board of regents voted to ban undocumented students from attending the top public universities in Georgia. I would like to acknowledge the four undocumented students that were forced to leave Georgia Tech in the Fall of 2010. They were barred from learning even though they were accepted and attended one of the top universities in the nation, a difficult goal no manner your background, all despite facing tremendous social-economic barriers. Ironically, the MLK memorial and the Center for Civil and Human Rights are only a few minutes away from campus. I would also like to acknowledge the undocumented construction workers who are literally building Georgia Tech while they and, in some cases, their children are banned from attending Georgia Tech. I hope that renowned institutions will recognize the potential of *all* students, regardless of social status.

I would like to thank my parents who were my very first coaches and mentors. My Mexican immigrant parents were able to raise four University of Illinois engineering scholars and, through shear will and determination, were able to achieve the fabled American dream. I would also like to thank my siblings who provided much support (and competition): Chuy, Freddy, and Maribel. Finally, my life partner, Maria Cecilia Quiñones. Without her, my time at Georgia Tech would have been meaningless. It was her support that pushed me along and kept my focus sharp.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

$\mathrm{E}\Big[X\Big]$	The expectation of random variable X.
$\mathrm{Var}\Big[X\Big]$	The variance of random variable X.
$\arg\min_x f(x),\ \arg\max_x f(x)$	The set of points x for which $f(x)$ attains its smallest (largest) value.
Ⓢ	Indication that a stochastic integral is evaluated in the Stratonovich sense.
$D_i F(y_1, y_2, \dots)$	The derivative of $F(y_1, y_2, \dots)$ with respect to its i^{th} argument.
$\mathbb{N}$	The set of natural numbers.
$\mathbb{R}$	The set of real numbers.
$\mathcal{N}(\mu, \sigma^2)$	A normal distribution with expectation μ and variance σ^2.
$\mathrm{erf}()$	The error function.
$\min_{x\in\mathcal{X}} f(x),\ \max_{x\in\mathcal{X}} f(x)$	The smallest (largest) value which $f(x)$ attains given $x \in \mathcal{X}$.
p	The generalized system configuration vector.
q	The generalized system momentum vector.

SUMMARY

Advances in machine autonomy hold great promise in advancing technology, economic markets, and general societal well-being. For example, the progression of unmanned air systems (UAS) research has demonstrated the effectiveness and reliability of these autonomous systems in performing complex tasks. UAS have shown to not only outperformed human pilots in some tasks, but have also made novel applications not possible for human pilots practical. Nevertheless, human pilots are still favored when performing specific challenging tasks. For example, transportation of suspended (sometimes called slung or sling) loads requires highly skilled pilots and has only been performed by UAS in highly controlled environments.

The presented work begins to bridge this autonomy gap by proposing a trajectory optimization framework for operations involving autonomous rotorcraft with suspended loads. The framework generates optimized vehicle trajectories that are used by existing guidance, navigation, and control systems and estimates the state of the non-instrumented load using a downward facing camera. Data collected from several simulation studies and a flight test demonstrates the proposed framework is able to produce effective guidance during autonomous suspended load operations. In addition, variational integrators are extensively studied in this dissertation. The derivation of a stochastic variational integrator is presented. It is shown that the presented stochastic variational integrator significantly improves the performance of the stochastic differential dynamical programming and the extended Kalman filter algorithms. A variational integrator for the propagation of polynomial chaos expansion coefficients is also presented. As a result, the expectation and variance of the trajectory of an uncertain system can be accurately predicted.

I

INTRODUCTION

> *You have just then copied a common item? - Yes. - Why have you bothered to do that? Why not create something new? - Because it's easier to do. - Well, isn't this sort of a joke then that you're playing on the public? - No. It gives me something to do.*
>
> – Conversation between a reporter and Andy Warhol

This dissertation investigates methods for the propagation and optimization of dynamical systems. The central and motivating application is autonomous rotorcraft with suspended load operations. In the course of developing an optimization framework for suspended load operations several questions arose: *How can system configuration propagation errors be reduced? How can stochasticity and uncertainty be accounted for when propagating a system configuration? What are the minimal necessary changes to an existing vehicle's hardware and software in order to perform successful suspended load operations?* The presented work aims to address these questions.

First, variational integrators are investigated due to their inherit ability to accurately propagate system configurations. Furthermore, a first-order linearization exists for the discrete dynamics produced by the specific variational integrator studied in this work. Therefore, algorithms that require linearization and propagation of system configurations, such as iterative optimization algorithms, can utilize the investigated variational integrator. However, some algorithms required a stochastic representation of the system (e.g. the extended Kalman filter). In this dissertation, a stochastic variational integrator is developed and the benefits of its use is studied in detailed. Furthermore, a polynomial chaos variational integrator is also derived in order to propagate the expansion coefficients of uncertain dynamical systems.

Returning to the motivating application, the variational integrator ensures that the proposed trajectory optimization framework is real-time feasible. Specifically, since a variational integrator is utilized a relatively large discretization time step can be used in the iterative optimization algorithm. The proposed framework generates optimized vehicle trajectories that are used by the vehicle's existing guidance, navigation, and control system. In order to reduce the amount of necessary hardware changes the suspended load and cable attachment assembly were not required to be instrumented. Furthermore, a downward facing camera attached to the vehicle was used for load state estimation. Data collected from several simulation studies and a flight test demonstrate the proposed framework is able to produce effective guidance during autonomous suspended load operations.

1.1 Contributions

The contributions presented in this dissertation are summarized as:

- **Development of a stochastic variational integrator.** It is shown that state and input dependent noise can be modeled as external forces and incorporated into an existing variational integrator. The first-order linearization of the discrete dynamics about a trajectory is derived. Furthermore, the expectation and covariance of the linearization are well-defined and easily computed.

- **Demonstration of benefits when the proposed variational integrator is used in the stochastic differential dynamic programming and extended Kalman filter algorithms.** A comparison between the utilization of the presented stochastic variational integrator and the standard Euler method in both algorithms is conducted. It is shown that the utilization of the variational integrator causes the performance of both algorithms to be less dependent on the size of the discretization time step. Therefore, the computational effort of both algorithms can be reduced.

- **Derivation of a polynomial chaos variational integrator.** It is demonstrated that the polynomial chaos expansion coefficients of an uncertain Hamiltonian system can be propagated using a variational integrator. The first-order linearization of the coefficient dynamics about a trajectory is also derived.

- **Development of a trajectory optimization framework for autonomous rotorcraft suspended load operations.** The main contribution of this dissertation is a trajectory optimization framework for suspended load operations. This framework requires minimal changes to an autonomous vehicle since the existing guidance, navigation and control system is utilized and a downward facing camera is used to estimate the state of the suspended load. Receding horizon iterative optimization algorithms and variational integrators are used to ensure a real-time and on-board guidance solution.

- **Demonstration of proposed trajectory optimization framework in simulation studies and a flight test.** Data collected from several simulation studies and a flight test demonstrate the proposed framework is able to produce effective guidance during autonomous suspended load operations.

1.2 Dissertation Outline

The outline of the dissertation is as follows:

- Chapter 2 summarizes relevant literature on suspended load control, guidance, and optimization. A highly abridged review of general trajectory optimization methods is then given. In addition, two trajectory optimization methods are reviewed in detailed, namely the stochastic differential dynamic programming algorithm (S-DDP) and a projection-based optimization methodology. Furthermore, a variational integrator and its linearization are reviewed through a detailed derivation. The final section in Chapter 2 describes polynomial chaos

expansion.

- A stochastic variational integrator is proposed in Chapter 3. The S-DDP and the extended Kalman filter algorithms are used to compare the performance of the variational integrator to that of the standard Euler method. It is shown that both algorithms become far less dependent on the discretization time step when the variational integrator is used to propagate system trajectories and linearize system dynamics.

- In Chapter 4, a variational integrator is formulated for the propagation of polynomial chaos expansion coefficients describing Hamiltonian systems. It is shown that the expansion coefficients of a Hamiltonian system evolve as a Hamiltonian system. As a result, a variational integrator is derived for the resulting expansion coefficient system. It is then shown that the presented integration method retains its accuracy over a large range of discretization time step sizes.

- Chapter 5 proposes a suspended load trajectory optimization framework. The framework is implemented on the GTMax, a modified Yamaha RMAX helicopter UAV, in a high fidelity simulation environment and is flight tested. A vision-based load state estimation method is used in order to reduce the amount of necessary hardware modifications. Furthermore, the framework can be used without modifying the vehicle's existing guidance, navigation, and control system. Simulation results and data from a flight test demonstrate the proposed framework can be used to effectively control the suspended load to track a given reference trajectory.

- Final conclusions are given in Chapter 6. Possible directions for future research and development are also given.

- Appendix A presents detailed derivations of variational integrators for a nonlinear mass-spring-damper system and a 3-link manipulator. Additional simulation results obtained using the proposed trajectory optimization framework are shown in Appendix B.

II

BACKGROUND

> *"Now when I was a little chap I had a passion for maps. I would look for hours at South America, or Africa, or Australia, and lose myself in all the glories of exploration. At that time there were many blank spaces on the earth, and when I saw one that looked particularly inviting on a map (but they all look that) I would put my finger on it and say, 'When I grow up I will go there'."*
>
> – Joseph Conrad, *Heart of Darkness*

In this chapter the existing relevant literature is summarized and key concepts are introduced. Specifically, past work in suspended load operations in manned rotorcraft systems, indoor and outdoor unmanned systems, and cranes is outlined. Next, a very brief overview of optimal control and trajectory optimization methods is given. Finally, detailed introductions to stochastic differential dynamical programming, projection-based optimization, variational integrators, and polynomial chaos expansion are given.

2.1 Suspended Load Operations

Automatic/semi-automatic control of loads suspended from a rotorcraft was initially investigated for manned systems during the 1970s. The goal of this research thrust was to assist pilots by actively damping the carried load. Proposed solutions included using specialized hardware such as an active winch system [2], controllable fins [38], on-load controlled aerodynamic surfaces [101], thrusters [93], an active arm [36], and reaction wheels [80]. Feedback methods to dampen the load directly with the vehicle's motions (without requiring additional hardware) were also investigated [24,39,51,73]. In recent work, there has been a strong emphasis in stability margins, handling qualities, and pilot perception. Handling qualities criteria for external load operations have

been proposed [76]. Analytical studies of cable angle/rate feedback have focused on improving stability margins, damping load oscillations, and reducing pilot workload during load placement [12, 13, 103]. A pilot display aid has been shown to be effective in damping pendulous load motion without the use of feedback control [41]. Recently, extensive piloted flight test for a task-tailored control system with fuselage and cable angle/rate feedback has shown a significant reduction of load set-down time and a large improvement in average handling qualities [53, 54].

The development of agile indoor quadrotor systems has spurred the advancement of autonomous methods for suspended load operations. A geometric control method has been shown to theoretically allow for highly aggressive load trajectory tracking, but it is yet to be seen if this framework can be implemented in practice [108, 109]. A coupled adaptive feedback and dynamic programming approach has been proposed to generate swing-free trajectories and mitigate the effect of changes in the system's center of gravity [94–96]. Additionally, reinforcement learning approaches have also been used to generate swing-free trajectories [28, 29]. Iterative optimal control algorithms, like those considered in this dissertation, have also been implemented in quadrotor systems with suspended loads [21, 22]. However, it should be noted that these methods have only been tested on highly agile, in-door quadrotors with the assistance of a precise position tracker, typically VICON. Further work is needed in order to demonstrate these approaches on less agile rotorcraft systems in outdoor flight. In addition, some of the research may not be applicable to larger platforms that are far less agile, such as the GTMax considered in this dissertation. For comparison, the GTMax, a modified Yamaha RMAX, weights 64 kg and is powered with a gasoline engine while the quadrotor used in Reference 29, an AscTec Hummingbird, weights 0.71 kg and uses electric motors.

Automatic control techniques developed for cranes have been shown to be useful

for rotorcraft systems. For example, input shaping methods have been used successfully in practice in both crane systems and UAVs [8, 64, 93, 106]. In this open-loop approach the system's input is modified such that oscillations induced by the system's trajectory are negated [107, 118]. Energy shaping and passivity-based control via interconnection and damping assignment (IDA-PBA) techniques have been applied to cranes in order to induce "virtual" damping to the system [3, 92]. Note that these methods generally try to reduce load oscillations through modification of the system's input. Therefore, aggressive maneuvers involving large swing angles will not be possible using the methods discussed above. A projection-based optimization method, described in more detail in Section 2.2.2, was used for position tracking (and not load damping) of a crane-like system in the presence of slow sensor updates [104].

There has been limited work on autonomous outdoor suspended load operations. Since precise measurements from a VICON system are not available, a GPS aided inertial navigation system is typically used to estimate the state of the vehicle. A variety of methods exist to estimate the state of the load: use of a magnetic encoder [4], attaching an IMU to the load [7], and image processing using a downward facing camera [10]. As already mention an input shaping method has been implemented and successfully flight tested on the GTMax platform [8]. Delayed cable angle feedback has also been successfully demonstrated [9, 93]. These techniques have been used for load delivery onto a moving platform [93]. A simple torque compensator method has shown to allow for single vehicle and multi-vehicle load transportation [4]. The Kaman K-MAX/BURRO autonomous helicopter has been operational in Afghanistan and has been shown to be a reliable alternative for resupply operations [19, 47, 79]. The Boeing A160T has also successfully perform suspended load operations, but has not been used in military deployments [18].

2.2 Trajectory Optimization Methods

A complete review of trajectory optimization methods and optimal control is far beyond the scope of this work. For a partial review of the subjects the reader is referred to References 5, 6, 14, 15, 20, 35, 49, 50, 65, 85, 100, and 113 (and references therein). Numerical methods that can be used to solve optimal control problems have been heavily investigated since the classical analytic solutions that are derived from calculus of variations and the Pontryagin's minimum principle are intractable except for very simple systems.

Generally, the numerical methods are categorized as *direct* or *indirect* [100]. In *indirect* approaches the calculus of variations is used to obtain multiple-point boundary-value problems. These boundary-value problems are then solved to obtain candidate trajectories and the final optimal trajectory is selected from these candidates. In *direct* methods, the optimal control problem is discretized. The resulting (non)linear programming problem is then solved through a variety of optimization methodologies.

As a representative example, the numerical approach that achieves real-time trajectory generation proposed in Reference 82 and 83 is discussed. This approach is similar to other model predictive control methodologies since it performs an online optimization by utilizing system models and predicted responses. Nonlinear system trajectories and control inputs are parameterized by B-splines. Sequential quadratic programming is then used to solve for the optimal spline coefficients. This approximation reduces the complexity of the problem, and hence, allows for real-time trajectory generation. The method has been implemented on the Caltech ducted fan [25]. NTG has been implemented for obstacle-avoiding trajectory generation on the Georgia Tech GTMax platform in high fidelity simulations and flight tests [62]. The approach can be implemented with a software library that is available for download [81].

In this dissertation, two numerical optimization algorithms are considered: differential dynamic programming (DDP) and projection-based optimization. In the

context of the presented work both methods rely on discretization of the system dynamics and are iterative in nature. While iterative algorithms may take several iterations to converge to an optimal solution, near-optimal solutions can usually be obtained in a few iterations. Near-optimal solutions can be sufficient to effectively control or guide a dynamical system with required specifications. That is, optimal solutions may not always be necessary. Furthermore, since real-time implementation is desired required computational effort is an important consideration. While other frameworks may result in optimal solutions, iterative methods, in general, can provide near-optimal solutions fairly quickly. Finally, both methods considered have been previously demonstrated in simulation studies or implemented in robotic systems.

The DDP algorithm generates optimal open and closed loop control policies by computing a quadratic approximation of the cost-to-go function and utilizing quadratically approximated state space dynamics around a trajectory [55]. The same basic principles were used to develop iterative linear quadratic regulators (iLQR) [72, 112]. Extensions of the DDP algorithm have been developed in order to address state and control constraints [67, 120]. Furthermore, the algorithm has been successfully implemented in simulation to enable robust bipedal robotic walking [86]. Finally, the stochastic differential dynamic programming (S-DDP) algorithm considers stochastic system dynamics with additive control- and state-dependent noise and finds optimal open and closed loop control policies to minimize the expectation of a given cost [111].

The projection-based optimization method defines an projection operator in order to recast the constrained optimization problem (constrained due to system dynamics) to an equivalent unconstrained optimization problem [42, 45]. A descent direction is then found for the unconstrained optimization problem using standard methods (e.g. Newton or quasi-Newton optimization). The system trajectory is then updated using the computed descent direction and the projection operator. This method was implemented for the control of an underactuated suspended load [104].

2.2.1 Stochastic Differential Dynamic Programming

Consider a class of stochastic dynamical systems that evolve according to

$$\mathrm{d}x = f(x,u)\ \mathrm{d}t + \sum_{i=1}^{m} F_i(x,u)\ \mathrm{d}\omega_i \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^p$ is the control input, and $\omega_i \in \mathbb{R}$ are independent Brownian noises. Furthermore, the considered cost is of the form

$$v(x,u,t) = \mathrm{E}\Big[h(x(t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} l(x(\tau),u(\tau),\tau)\ \mathrm{d}\tau\Big] \tag{2.2}$$

where $h(x(t_\mathrm{f})$ is the terminal cost and $l(x(t),u(t),t)$ is the instantaneous cost. The S-DDP algorithm attempts to find the optimal sequence of discrete inputs to minimize the given cost such that the continuous input is then defined as $u(T_k) = u_k,\ T_k = [t_0 + k\Delta t,\ t_0 + (k+1)\Delta t)$ where Δt is the discretization time step. As a result, the algorithm approximates continuous system trajectories by a sequence of state vectors such that $x_1 = x(t_0)$, $x_2 = x(t_0 + \Delta t)$, $\ldots$, $x_N = x(t_\mathrm{f})$. The appropriate selection of Δt depends on the given system dynamics and cost function. It should be noted that the S-DDP algorithm is iterative. Specifically, given a sequence of discrete inputs $\{u_1,\ldots,u_{N-1}\}$ the S-DDP algorithm finds the optimal control deviation such that the control input is updated as

$$u_{i+1} = u_i + \gamma\delta u_i^\star, \tag{2.3}$$

where γ is a user defined constant or is selected from an automated process (e.g. Armijo line search). However, several iterations may be needed in order to arrive at a control input that is sufficiently close to the optimal solution.

To begin the derivation[1] of the S-DDP algorithm it is assumed that a sequence of *nominal* discrete inputs $\{\bar{u}_1,\ldots,\bar{u}_{N-1}\}$ and the associated state trajectory $\{\bar{x}_1,\ldots,\bar{x}_N\}$

[1]Reference 111 presents a complete and detailed derivation of the S-DDP algorithm.

are given. Next, the first-order linearization of the system dynamics are given as

$$\delta x_{k+1} = (\mathrm{I} + \nabla_x f(x_k, u_k)\Delta t + \sum_{i=1}^{p} \nabla_x F_i(x_k, u_k)\omega_{ik})\delta x_k + (\nabla_u f(x_k, u_k)\Delta t + \sum_{i=1}^{p} \nabla_u F_i(x_k, u_k)\omega_{ik})\delta u_k + \sum_{i=1}^{p} F_i(x_k, u_k)\omega_{ik}. \quad (2.4)$$

where $\omega_{ik} \sim \mathcal{N}(0, \Delta t)$ are independent random variables. For ease of exposition, the notation is condensed to

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k + \sum_{i=1}^{m} \Gamma_{ik}\omega_{ik}. \quad (2.5)$$

where

$$A_k = \mathrm{I} + \nabla_x f(x_k, u_k)\Delta t \quad (2.6)$$

$$B_k = \nabla_u f(x_k, u_k)\Delta t \quad (2.7)$$

$$\Gamma_{ik} = \nabla_x F_i(x_k, u_k)\delta x_k + \nabla_u F_i(x_k, u_k)\delta u_k + F_i(x_k, u_k) \quad (2.8)$$

Utilizing the derived first-order linearization of the system dynamics a second-order expansion of the cost-to-go function around the nominal trajectory is obtained[2]:

$$V(\bar{x} + \delta x) = V(\bar{x}) + V_x(\bar{x})^{\mathrm{T}}\delta x + \frac{1}{2}\delta x^{\mathrm{T}} V_{xx}(\bar{x})\delta x, \quad (2.9)$$

or, in its approximated discrete form,

$$\begin{aligned} V(\bar{x}_{k+1} + \delta x_{k+1}) &= V(\bar{x}_{k+1}) + V_x(\bar{x}_{k+1})^{\mathrm{T}}\delta x_{k+1} + \delta x_{k+1}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})\delta x_{k+1} \\ &= V(\bar{x}_{k+1}) + V_x(\bar{x}_{k+1})^{\mathrm{T}}(A_k\delta x_k + B_k\delta u_k + \sum_{i=1}^{m}\Gamma_{ik}\omega_{ik}) + \\ &\frac{1}{2}(A_k\delta x_k + B_k\delta u_k + \sum_{i=1}^{m}\Gamma_{ik}\omega_{ik})^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})(\dots) \end{aligned}$$

The expectation of the second-order expansion of the cost-to-go function, $\mathrm{E}\Big[V(\bar{x}_{k+1} + \delta x_{k+1})\Big]$, is needed. The expectation of the second term in equation (2.10) is given as

$$\mathrm{E}\Big[V_x(\bar{x}_{k+1})^{\mathrm{T}}(A_k\delta x_k + B_k\delta u_k + \sum_{i=1}^{m}\Gamma_{ik}\omega_{ik})\Big] = V_x(\bar{x}_{k+1})^{\mathrm{T}}(A_k\delta x_k + B_k\delta u_k) \quad (2.10)$$

[2]For ease of exposition, notation for derivatives is condensed to $\nabla_z g = g_z$ and $\nabla_{xz} g = g_{xz}$

The expectation of the third term in equation (2.10) is given as

$$\begin{aligned}\mathrm{E}\Big[\delta x_{k+1}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})\delta x_{k+1}\Big) &= \mathrm{E}\Big[\delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\Big] + \mathrm{E}\Big[\delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k\Big] \\ &\quad + 2\mathrm{E}\Big[\delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\Big] \\ &\quad + 2\sum_{i=1}^{m} \mathrm{E}\Big[\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\Big] \\ &\quad + 2\sum_{i=1}^{m} \mathrm{E}\Big[\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k\Big] \\ &\quad + \sum_{i=1}^{m} \mathrm{E}\Big[(\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})\Gamma_{ik}\omega_{ik}\Big]. \end{aligned} \tag{2.11}$$

Note that

$$\mathrm{E}\Big[\delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\Big] = \delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k, \tag{2.12}$$

$$\mathrm{E}\Big[\delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k\Big] = \delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k, \tag{2.13}$$

$$\mathrm{E}\Big[\delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k\Big] = \delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k, \tag{2.14}$$

$$\mathrm{E}\Big[\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k\Big] = 0, \quad \forall i, \tag{2.15}$$

$$\mathrm{E}\Big[\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k\Big] = 0, \quad \forall i, \tag{2.16}$$

and

$$\begin{aligned}\mathrm{E}\Big[\omega_{ik}\Gamma_{ik}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})\Gamma_{ik}\omega_{ik}\Big] &= \Delta t\big(\delta x_k^{\mathrm{T}} \nabla_x F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_x F_i(x_k,u_k)\delta x_k \\ &\quad + \delta u_k^{\mathrm{T}} \nabla_u F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_u F_i(x_k,u_k)\delta u_k \\ &\quad + F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k,u_k) \\ &\quad + 2\delta x_k^{\mathrm{T}} \nabla_x F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_u F_i(x_k,u_k)\delta u_k \\ &\quad + 2\delta x_k^{\mathrm{T}} \nabla_x F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k,u_k) \\ &\quad + 2\delta u_k^{\mathrm{T}} \nabla_u F_i(x_k,u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k,u_k)\big). \end{aligned} \tag{2.17}$$

Therefore,

$$\begin{aligned}\mathrm{E}\Big[\delta x_{k+1}^{\mathrm{T}} V_{xx}(\bar{x}_{k+1})\delta x_{k+1}\Big] &= \delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k + \delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k \\ &\quad + 2\delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k + \delta x_k^{\mathrm{T}} \mathcal{F} \delta x_k + \delta u_k^{\mathrm{T}} \mathcal{Z} \delta u_k \\ &\quad + 2\delta x_k^{\mathrm{T}} \mathcal{L} \delta u_k + 2\delta x_k^{\mathrm{T}} \mathcal{S} + 2\delta u_k^{\mathrm{T}} \mathcal{U} + \mathcal{T}, \end{aligned} \tag{2.18}$$

where

$$\mathcal{F} = \Delta t \sum_{i=1}^{m} \nabla_x F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_x F_i(x_k, u_k) \tag{2.19}$$

$$\mathcal{Z} = \Delta t \sum_{i=1}^{m} \nabla_u F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_u F_i(x_k, u_k) \tag{2.20}$$

$$\mathcal{T} = \Delta t \sum_{i=1}^{m} F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k, u_k) \tag{2.21}$$

$$\mathcal{L} = \Delta t \sum_{i=1}^{m} \nabla_x F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) \nabla_u F_i(x_k, u_k) \tag{2.22}$$

$$\mathcal{S} = \Delta t \sum_{i=1}^{m} \nabla_x F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k, u_k) \tag{2.23}$$

$$\mathcal{U} = \Delta t \sum_{i=1}^{m} \nabla_u F_i(x_k, u_k)^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) F_i(x_k, u_k). \tag{2.24}$$

Finally, the second order expansion of the cost-to-go function can now be given as

$$\begin{aligned} V(\bar{x}_{k+1} + \delta x_{k+1}) &= V(\bar{x}_{k+1}) + V_x(\bar{x}_{k+1})^{\mathrm{T}}(A_k \delta x_k + B_k \delta u_k) + \delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k \\ &\quad + \frac{1}{2}\delta x_k^{\mathrm{T}} A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k \delta x_k + \frac{1}{2}\delta u_k^{\mathrm{T}} B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k \delta u_k \\ &\quad + \frac{1}{2}\delta x_k^{\mathrm{T}} \mathcal{F} \delta x_k + \frac{1}{2}\delta u_k^{\mathrm{T}} \mathcal{Z} \delta u_k + \delta x_k^{\mathrm{T}} \mathcal{L} \delta u_k + \delta x_k^{\mathrm{T}} \mathcal{S} + \delta u_k^{\mathrm{T}} \mathcal{U} + \frac{1}{2}\mathcal{T}. \end{aligned}$$

Now consider the discretized state action value function defined as

$$Q(x_k, u_k) = L(x_k, u_k) + V(x_{k+1}) \tag{2.25}$$

where $L(x_k, u_k) = l(x_k, u_k)\Delta t$. Note that the $Q(x_k, u_k)$ is a function of u_k and is backward propagating. By invoking Bellman's Principle of Optimality it is known that the optimal control deviation, $\delta u^\star$, is found by minimizing the state action value.

A local quadratic approximation of the state action value function is now derived as

$$\begin{aligned} Q(\bar{x}_k + \delta x_k, \bar{u}_k + \delta u_k) = Q(\bar{x}_k, \bar{u}_k) &+ \delta u_k^{\mathrm{T}} Q_u(\bar{x}_k, \bar{u}_k) + \delta x_k^{\mathrm{T}} Q_x(\bar{x}_k, \bar{u}_k) \\ &+ \frac{1}{2}\delta u_k^{\mathrm{T}} Q_{uu}(\bar{x}_k, \bar{u}_k)\delta u_k + \frac{1}{2}\delta x_k^{\mathrm{T}} Q_{xx}(\bar{x}_k, \bar{u}_k)\delta x_k \\ &+ \delta u_k^{\mathrm{T}} Q_{ux}(\bar{x}_k, \bar{u}_k)\delta x_k \end{aligned} \tag{2.26}$$

where

$$Q_x(\bar{x}_k, \bar{u}_k) = L_x(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_x(\bar{x}_{k+1}) + \mathcal{S} \tag{2.27}$$

$$Q_u(\bar{x}_k, \bar{u}_k) = L_u(\bar{x}_k, \bar{u}_k) + B_k^{\mathrm{T}} V_x(\bar{x}_{k+1}) + \mathcal{U} \tag{2.28}$$

$$Q_{xx}(\bar{x}_k, \bar{u}_k) = L_{xx}(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) A_k + \mathcal{F} \tag{2.29}$$

$$Q_{uu}(\bar{x}_k, \bar{u}_k) = L_{uu}(\bar{x}_k, \bar{u}_k) + B_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k + \mathcal{Z} \tag{2.30}$$

$$Q_{xu}(\bar{x}_k, \bar{u}_k) = L_{xu}(\bar{x}_k, \bar{u}_k) + A_k^{\mathrm{T}} V_{xx}(\bar{x}_{k+1}) B_k + \mathcal{L} \tag{2.31}$$

The optimal control deviation $\delta u^\star$ can now be solved for by minimizing the state action value as

$$\begin{aligned} \delta u_k^\star &= \arg\min_{\delta u_k} Q(\bar{x}_k + \delta x_k, \bar{u}_k + \delta u_k) \\ &= -Q_{uu}(\bar{x}_k, \bar{u}_k)^{-1}(Q_u(\bar{x}_k, \bar{u}_k) + Q_{ux}(\bar{x}_k, \bar{u}_k)\delta x_k). \end{aligned} \tag{2.32}$$

Plugging $\delta u^\star$ back into (2.26) yields a backward propagating approximation of the second-order expansion of the value function:

$$V(\bar{x}_k) = V(\bar{x}_{k+1}) + L(\bar{x}_k, \bar{u}_k) - Q_u(\bar{x}_k, \bar{u}_k) Q_{uu}(\bar{x}_k, \bar{u}_k)^{-1} Q_u(\bar{x}_k, \bar{u}_{k+1}), \tag{2.33}$$

$$V_x(\bar{x}_k) = Q_x(\bar{x}_k) - Q_u(\bar{x}_k, \bar{u}_k) Q_{uu}(\bar{x}_k, \bar{u}_k)^{-1} Q_{ux}(\bar{x}_k, \bar{u}_k)^{\mathrm{T}}, \tag{2.34}$$

$$V_{xx}(\bar{x}_k) = Q_{xx}(\bar{x}_k) - Q_{xu}(\bar{x}_k, \bar{u}_k) Q_{uu}(\bar{x}_k, \bar{u}_k)^{-1} Q_{ux}(\bar{x}_k, \bar{u}_k)^{\mathrm{T}}, \tag{2.35}$$

where the initial conditions are derived from the terminal cost, $h(\cdot)$, as

$$V(\bar{x}_N) = h(\bar{x}_N), \tag{2.36}$$

$$V_x(\bar{x}_N) = h_x(\bar{x}_N), \tag{2.37}$$

$$V_{xx}(\bar{x}_N) = h_{xx}(\bar{x}_N). \tag{2.38}$$

This completes the derivation of the S-DDP algorithm. The derived optimal control deviation, $\delta u^\star$, is used to update the nominal input. The process can then be repeated using the updated input as the new nominal input. There are several points that should be considered:

- Note that the optimal control deviation given by (2.32) contains a feed-forward and a feedback component. Therefore, the term $Q_{uu}^{-1}Q_{ux}$ gives the optimal state feedback for a particular iteration.

- $V(\cdot)$, $V_x(\cdot)$, and $V_{xx}(\cdot)$ are backwards propagating. Therefore, the approximation of the value function can be computed given a nominal input, the associated nominal trajectory, and the linearized system dynamics along the nominal trajectory. Furthermore, given the approximation of the value function, the optimal control deviation $\delta u^\star$ can be propagated.

- Recall that $\delta u^\star$ is the optimal control deviation and the control input is updated using a step size, γ (see equation 2.3). In the S-DDP (and DDP) implementations found in this dissertation an Armijo line search is used to automaticly select an appropriate step size [63].

- Note that the linearization scheme given in equation (2.4) is not unique. In the following chapters, it is highlighted that replacing the Euler linearization with a linearization derived from a variational integrator greatly improves the performance of the S-DDP algorithm.

- Note that the nominal state trajectory is obtained by propagating (or simulating) the system forward with the given nominal control input. Therefore, it is safe to assume that the performance of the S-DDP algorithm is dependent on the accuracy of this propagation. In the following chapters, it is highlighted that variational integrators greatly improve the performance of the S-DDP algorithm due to their ability to propagate system configurations forward in time

with great accuracy.

- Deterministic dynamics can be considered by removing the diffusion vector fields, $F_i(x, u) = 0$.

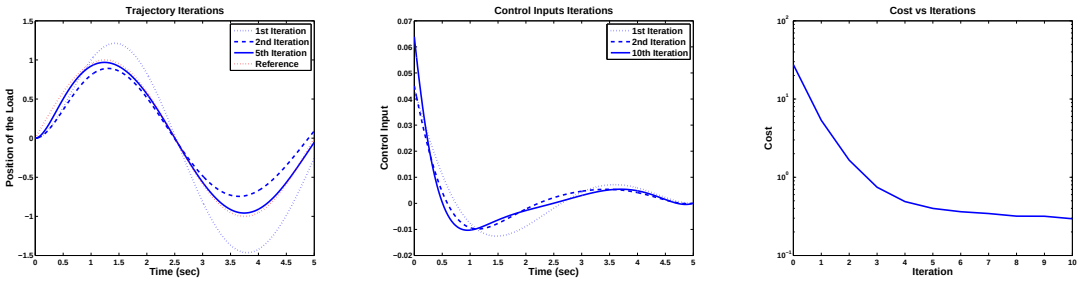
The S-DDP algorithm is outlined in Algorithm 1.

Algorithm 1 The S-DDP Algorithm with an Armijo Line Search

Require:
Initial discrete control input $u(t)$, Algorithm parameters α, β, ϵ
Cost function $v(x, u, t)$, Stochastic dynamics $\mathrm{d}x = f(x, u)\ \mathrm{d}t + \sum_{i=1}^{m} F_i(x, u)\ \mathrm{d}\omega_i$
while Cost updates results in more than ϵ in difference **do**
  Find the discretized expected trajectory
  Linearize the value function and system dynamics along the trajectory
  Approximate the value function through back-propagation
  Compute $\delta u^\star$ and $\delta x^\star$
  while $\mathrm{Cost_p} > \mathrm{Cost} + \alpha\beta(\delta x^{\mathrm{T}} \nabla_x v(x, u, t) + \delta u^{\mathrm{T}} \nabla_u v(x, u, t))$ **do**
    Find the proposed input $u_\mathrm{p} \leftarrow u + \beta^j \delta u^\star$ and corresponding trajectory, x_p
    Find proposed cost $\mathrm{Cost_p} \leftarrow v(x_\mathrm{p}, u_\mathrm{p}, t)$ and update $j \leftarrow j + 1$ if needed
  end while
  Update the control, trajectory and Cost: $u \leftarrow u_\mathrm{p}, x \leftarrow x_\mathrm{p}, \mathrm{Cost} \leftarrow \mathrm{Cost_p}$
end while

2.2.1.1 Overview of the S-DDP Algorithm

The stochastic differential dynamic programming (S-DDP) algorithm numerically solves nonlinear stochastic optimal control problems using first and second order expansions of stochastic dynamics and cost along nominal trajectories. It is based on the classic differential dynamical programming algorithm, but considers stochastic dynamical systems of the form $\mathrm{d}x = f(x, u)\ \mathrm{d}t + F(x, u)\ \mathrm{d}\omega$ where x is the system state, u is the control input, and $d\omega$ is Brownian noise. The S-DDP algorithm considers a cost parameterized as $v(x, u, t) = E[\int_{t_0}^{t_\mathrm{f}} l(x(\tau), u(\tau), \tau)\ \mathrm{d}\tau + h(x(t_\mathrm{f}))]$ where $h(\cdot)$ is the terminal cost and $l(\cdot)$ is the running cost. Note that the cost is an expectation since the underlining system dynamics are stochastic. The essence of the algorithm lies in finding the optimal local variation in control $\delta u^\star$ that minimizes the given cost. First, given an initial discrete control input $u(t)$ the discretized expected



(a) (b) (c)

Figure 2.1: (a) and (b): The optimized state trajectory and control input at a variety of iterations. (c): Optimized cost as a function of number of optimization iterations.

trajectory of the system is found. Then the value function and system dynamics are linearized along the trajectory. An approximation of the value function is found through backwards-propagation. The approximated value function is used to find $\delta u^\star$, $\delta x^\star$ (the optimal state deviation), and the optimal feedback gain. The discrete control input is updated as $u(t) = u(t) + \gamma \delta u^\star(t)$ and the process can then be repeated. In this paper γ is found through an Armijo line search (see Reference 63 for further details) and the algorithm is terminated when consecutive final costs differ by less then a defined amount. It should be noted that the algorithm relies heavily on the accuracy of the propagation of the expected trajectory of the system and the linearization of the system dynamics. The process is outline in Algorithm 1.

As a simple demonstrative example consider a deterministic and linear mass-spring system where its mass is unity and the spring stiffness constant is 1 N/m. Furthermore, consider the reference-tracking based cost given as

$$v(x,u,t) = \int_{t_0}^{t_\mathrm{f}} \left(10(x(\tau) - x_\mathrm{ref}(\tau))^2 + (\dot{x}(\tau) - \dot{x}_\mathrm{ref}(t))^2 + u^2(\tau)\right) \mathrm{d}\tau. \tag{2.39}$$

The DDP algorithm was used to find the optimal input. The initial nominal input was given as $u(t) = 0$. Figure 2.1 shows the state trajectory and control input at a variety of iterations and the optimized cost as a function of the number of optimization iterations. Notice the iterative nature of the DDP algorithm.

2.2.2 Projection-Based Optimization

A complete treatment of projection operator strategies for the optimization of trajectory functionals is given in References 42, 44, and 45. Furthermore, this method has been considered in a variety of systems and generalization to Lie groups have been reported in the literature [43, 46, 89, 102]. In the context of the work presented in this dissertation, the constrained optimization problem is formulated with a cost of the form[3]

$$v(\xi) = h(x(t_{\mathrm{f}}) + \int_{t_0}^{t_{\mathrm{f}}} l(x(\tau), u(\tau), \tau)\ \mathrm{d}\tau \tag{2.40}$$

$$\begin{aligned} l(x(\tau), u(\tau), \tau) &= (x(\tau) - x_{\mathrm{d}}(\tau))^{\mathrm{T}} Q (x(\tau) - x_{\mathrm{d}}(\tau)) \\ &\quad + (u(\tau) - u_{\mathrm{d}}(\tau))^{\mathrm{T}} R (u(\tau) - u_{\mathrm{d}}(\tau)) \end{aligned} \tag{2.41}$$

$$l(x(\tau), u(\tau), \tau) = (x(t_{\mathrm{f}}) - x_{\mathrm{d}}(t_{\mathrm{f}}))^{\mathrm{T}} Q_{\mathrm{f}} (x(t_{\mathrm{f}}) - x_{\mathrm{d}}(t_{\mathrm{f}})) \tag{2.42}$$

subject to the system dynamics

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0, \tag{2.43}$$

where $\xi = (x, u)$ is the trajectory of the system over the time horizon $t = [t_0, t_{\mathrm{f}}]$, $\xi_{\mathrm{d}} = (x_{\mathrm{d}}(\cdot), u_{\mathrm{d}}(\cdot))$ is the desired trajectory and Q, Q_{f}, and R are positive definite matrices. Furthermore, it is not assumed that ξ_{d} satisfies the given system dynamics. The trajectory manifold of the system $\mathcal{T}$ is defined such that if $\eta = (x(\cdot), u(\cdot))$ satisfies the system dynamics $\dot{x}(t) = f(x(t), u(t))$ then $\eta \in \mathcal{T}$. Next, a projection operator, $\mathcal{P}$, is defined as the mapping

$$\mathcal{P} : \phi = (\alpha, \mu) \to \xi = (x, u) \in \mathcal{T}. \tag{2.44}$$

Note that the projection operator simply maps an arbitrary trajectory to a trajectory in $\mathcal{T}$. If $\mathcal{P}$ exist then the consider constrained optimization problem can be recast as

[3]The presented discussion closely follows Section 3 in Reference 45.

an unconstrained optimization problem where

$$\min_{\xi \in \mathcal{T}} v(\xi) = \min_{\xi} v(\mathcal{P}(\xi)). \tag{2.45}$$

For the class of problems considered here the projection operator is given as

$$x(t_0) = \alpha(t_0) \tag{2.46}$$

$$u(t) = \mu(t) + K(t)(\alpha(t) - x(t)) \tag{2.47}$$

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.48}$$

where $K(t)$ is a time-varying feedback gain matrix found by solving a finite horizon linear quadratic regulator (LQR) on a linearized representation of the system [88]. Specifically, suppose the (LQR) cost is parameterized as

$$J(x, u, t) = \int_{t_0}^{t_\mathrm{f}} \left(Gx(\tau)^2 + Fu(\tau)^2 \right) \, \mathrm{d}\tau + G_\mathrm{f} x(t_\mathrm{f})^2, \tag{2.49}$$

then a discrete approximation of the time-varying feedback gain matrix $K(t)$ is found via a discrete backward propagating Riccati equation:

$$P_J = G_\mathrm{f} \tag{2.50}$$

$$P_{j-1} = G + A_j^\mathrm{T} P_j A_j - A_j^\mathrm{T} P_j B (R_j + B_j^\mathrm{T} P_j B_j)^{-1} B_j^\mathrm{T} P_j A_j, \tag{2.51}$$

$$K_j = (F + B_j^\mathrm{T} P_j B_j)^{-1} B_j^\mathrm{T} P_j A_j \tag{2.52}$$

where (A_j, B_j) is the linearization of the system dynamics at time t_j and J is the number of discretization points ($t_J = t_\mathrm{f}$).

Now suppose the ξ_i is the current trajectory iterate (the optimized trajectory at the current iteration) using Newton's descent methods the descent direction is given as

$$\zeta_i = \arg\min_{\zeta} Dv(\xi_i) \cdot \zeta + \frac{1}{2} D^2 v(\mathcal{P}(\xi_i)) \cdot (\zeta, \zeta). \tag{2.53}$$

In this section, $Dg(\xi_i)$ and $D^2 g(\xi_i)$ are the first and second Frechet derivatives of the Banach space functional g, respectively. Since the considered dynamical system

evolves in a vector space, $D^2v(\mathcal{P}(\xi_i)) \cdot (\zeta, \zeta)$ is well-defined and is computed as

$$D^2v(\mathcal{P}(\xi_i)) \cdot (\zeta, \zeta) = D^2v(\xi_i) \cdot (\zeta, \zeta) + Dv(\xi_i) \cdot D^2\mathcal{P}(\xi_i) \cdot (\zeta, \zeta). \tag{2.54}$$

If the quasi-Newton method is considered equation (2.54) is modified to

$$D^2v(\mathcal{P}(\xi_i)) \cdot (\zeta, \zeta) = D^2v(\xi_i) \cdot (\zeta, \zeta). \tag{2.55}$$

A rigorous treatment of the differentiation of the projection operator and a method for calculating $D^2\mathcal{P}(\xi)$ is presented in Reference 45. Note that equation (2.53) is a linear quadratic problem defined using the first and second derivatives of the nonlinear system dynamics and the cost, $v(\xi)$, about ξ_i. The solution is obtained by numerically integrating the associated differential Riccati equations. A line search is then perform in order to find the optimize step size, γ, such that

$$\gamma_i = \underset{\gamma \in (0,1]}{\arg\min}\, v(\mathcal{P}(\xi_i + \gamma\zeta_i)). \tag{2.56}$$

For implementations presented in this dissertation an Armijo line search is used. The trajectory is then updated:

$$\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i\zeta_i). \tag{2.57}$$

References 42,44, and 45 present further important mathematical details not included here. Figure 2.2 (from Reference 44) provides a visual representation of the algorithm.

Algorithm 2 Projection-Based Optimization

Require:
Initial trajectory ξ_0, Algorithm parameters α, β, ϵ
Cost function $v(x, u, t)$, System dynamics $dx = f(x, u)dt$
while Cost updates results in more than ϵ in difference **do**
Linearize the value function and system dynamics along the trajectory, ξ_i
Compute the time-varying feedback gain matrix, $K(t)$
Compute the search direction, ζ_i
Find the optimized step size, γ_i
Update the trajectory, ξ_{i+1}
end while

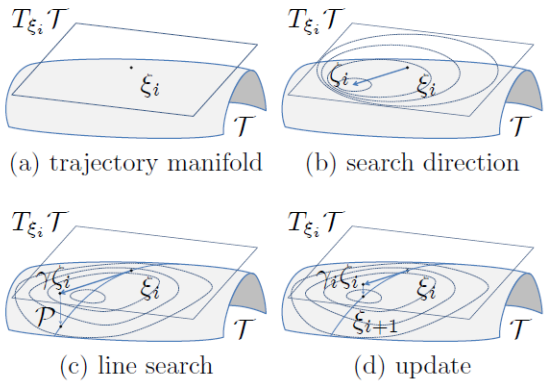


Figure 2.2: The projection-based optimization methodology: (a) the tangent space $T_{\xi_i}\mathcal{T}$ is computed by linearizing the system about the trajectory ξ_i, (b) the search direction ζ_i is computed, (c) the optimal step size is computed through a line search, γ_i, (d) a new updated trajectory ξ_{i+1} is obtained. Image from Reference 44.

Notice that in both the DDP algorithm and the Projection-Based optimization framework the optimal control deviation is sought. In the DDP algorithm, the deviation is found through approximation of the value function that arises from the considered cost and system dynamics. In the Projection-Based optimization framework, the problem is recasted to a simpler problem and, through the use of a projection operator, the computed optimal control deviation is found for the original problem. Furthermore, both methods rely on system propagation and linearization.

2.3 Variational Integrators and Structured Linearization

Variational integrators have been extensively studied and developed in the past decades [68, 69, 77, 90]. Through discretization of Hamilton's variational principle algorithms can be developed to propagate a Hamiltonian system's discrete trajectory. If the system is dissipative or forced the Lagrange-d'Alembert principle is similarly invoked. Since variational integrators are derived from the system's fundamental characteristics they are inherently well-suited to handle the integration of Euler-Lagrange equations.

Variational integrators have been shown to have many desirable properties not found in general ordinary differential equation numerical solvers [68]. For example, for the case where no external forces are present the integrator is symplectic and momentum conserving [77]. These benefits also extend to complex systems involving holonomic constraints, elastodynamics, and two-body collisions [30, 70, 115].

In this dissertation variational integrators are shown to enable real-time nonlinear trajectory optimization. Specifically, a variational integrator and its linearization are utilized in the differential dynamic programing (DDP) algorithm used to obtain optimal trajectories for the studied suspended load system. Furthermore, a stochastic variational integrator, a polynomial chaos variational integrator, and their linearizations are developed in this dissertation. It is shown that the performance of the stochastic differential dynamic programing (S-DDP) algorithm improves when the developed stochastic variational integrator is utilized when compared to utilizing the Euler method. Similarly, when compared to the Euler method the polynomial chaos variational integrator is shown to more accurately propagate expansion coefficients. The following sections review the development of the variational integrator and its linearization utilized in the implementation of the proposed real-time optimal trajectory generation framework.

2.3.1 A Variational Integrator

To begin, consider the Lagrangian $L(q(t), \dot{q}(t))$ of a dynamical system given as

$$L(q(t), \dot{q}(t)) = T(q(t), \dot{q}(t)) - V(q(t)), \tag{2.58}$$

where q is the state configuration vector, $\dot{q}$ is its time derivative, $T(q(t), \dot{q}(t))$ describes the system's kinetic energy, and $V(q(t))$ describes the system's potential energy. The action, S, can now be defined as

$$S[q(t)] = \int_{t_0}^{t_\mathrm{f}} L(q(\tau), \dot{q}(\tau))\ \mathrm{d}\tau. \tag{2.59}$$

The Least Action principle can now be used to derive the following variational relation

$$\delta S[q(t)] = \delta \int_{t_0}^{t_\mathrm{f}} L(q(\tau), \dot{q}(\tau))\ \mathrm{d}\tau = 0 \tag{2.60}$$

that results, when minimized, in the classical Euler-Lagrange equations [75]:

$$\frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{q}}(q, \dot{q}) - \frac{\partial L}{\partial q}(q, \dot{q}) = 0. \tag{2.61}$$

Note equation (2.61) provides the fundamental characteristics of a dynamical system and describes how the system configuration vector propagates through time. However, it does not provide any method or algorithm that can be used to solve for the trajectory of the system. Simply using numerical integration schemes developed for general second order differential equations will result in numerical errors since the system's fundamental characteristics are ignored.

On the other hand, variational integrators approximate the continuous trajectory of mechanical systems with a sequence of discrete points while ensuring (or strongly enforcing) the conservation of fundamental quantities such as momentum and energy [77]. Specifically, for the variational integrator considered in this dissertation, a sequence of system configuration vectors $\{(t_0, q_0), (t_1, q_1), \ldots, (t_n, q_n)\}$ is found such that the continuous system trajectory is approximated as $q_m \approx q(t_m)$ where $\Delta t = t_{i+1} - t_i$ is the discretization time step. Derivations for the same variational integrator are given in References 58 and 90

The derivation of the variational integrator begins with approximating the action integral over a small time interval with a generalized midpoint approximation

$$L_\mathrm{d}(q_k, q_{k+1}) = L((1-\alpha)q_k + \alpha q_k, \frac{q_{k+1} - q_k}{\Delta t})\Delta t \approx \int_{t_k}^{t_{k+1}} L(q(\tau), \dot{q}(\tau))\ \mathrm{d}\tau, \tag{2.62}$$

where $\alpha \in [0, 1]$ defines the integration approximation, $\alpha = 1/2$ results in second order accuracy as discussed in Reference 115, and $L_\mathrm{d}(q_k, q_{k+1})$ is referred to as the *discrete Lagrangian*. The action integral defined in equation (2.59) can be approximated as

an action sum using discrete Lagrangians:

$$S[q(t)] \approx \sum_{k=0}^{n-1} L_{\mathrm{d}}(q_k, q_{k+1}). \tag{2.63}$$

The Least Action principle can now be used to derive the following variational relation

$$\delta S[q(t)] \approx \sum_{k=0}^{n-1} (D_1 L_{\mathrm{d}}(q_k, q_{k+1}) \cdot \delta q_k + D_2 L_{\mathrm{d}}(q_k, q_{k+1}) \cdot \delta q_{k+1}) = 0. \tag{2.64}$$

Equivalently,

$$\delta S[q(t)] \approx \sum_{k=1}^{n-1} (D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + D_2 L_{\mathrm{d}}(q_{k-1}, q_k)) \cdot \delta q_k = 0, \tag{2.65}$$

since $\delta q_0 = \delta q_n = 0$. Note that the Least Action principle requires that the variations of the action sum be zero for any δq_k. As a result, the implicit difference Discrete Euler-Lagrange (DEL) equation is given as

$$D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + D_2 L_{\mathrm{d}}(q_{k-1}, q_k) = 0. \tag{2.66}$$

Notice that the DEL equation is the discrete time equivalent to the classical Euler-Lagrange equation (2.61). However, the DEL equation provides a manner in which the system's discrete configuration can be propagated. Given two consecutive system configurations q_k and q_{k-1},

$$f(q_{k+1}) = D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + D_2 L_{\mathrm{d}}(q_{k-1}, q_k), \tag{2.67}$$

can be solve implicitly for the next configuration q_{k+1}. Therefore, given q_0 and q_1 equation (2.67) can be solved iteratively to find $q_2, \ldots, q_n$. A simple root finder algorithm outline in Algorithm 3 can be used to solve equation (2.67). The required derivative $Df(\cdot)$ is given as

$$Df(q_{k+1}) = D_2 D_1 L_{\mathrm{d}}(q_k, q_{k+1}). \tag{2.68}$$

It should be noted that the required derivatives in equations (2.67), (2.68) and

Algorithm 3 Simple Root Finder

while $|f(q_{k+1})| > \epsilon_{\text{tol}}$ **do**
  $q_{k+1} \leftarrow q_{k+1} - Df^{-1}(q_{k+1}) \cdot f(q_{k+1})$
end while

those needed for the linearization of the integrator equations, presented in the following section, can be found using the chain rule and equation (2.62) [58]:

$$D_1 L_{\text{d}}(q_k, q_{k+1}) = \frac{\partial}{\partial q} L(q, \dot{q})(1-\alpha)\Delta t - \frac{\partial}{\partial \dot{q}} L(q, \dot{q}) \tag{2.69}$$

$$D_2 L_{\text{d}}(q_k, q_{k+1}) = \frac{\partial}{\partial q} L(q, \dot{q})\alpha\Delta t + \frac{\partial}{\partial \dot{q}} L(q, \dot{q}) \tag{2.70}$$

$$\begin{aligned} D_1 D_1 L_{\text{d}}(q_k, q_{k+1}) = {} & \frac{\partial^2}{\partial q \partial q} L(q, \dot{q})(1-\alpha)^2\Delta t - \frac{\partial^2}{\partial \dot{q} \partial q} L(q, \dot{q})(1-\alpha) \\ & - \frac{\partial^2}{\partial q \partial \dot{q}} L(q, \dot{q})(1-\alpha) + \frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q, \dot{q}) \frac{1}{\Delta t} \end{aligned} \tag{2.71}$$

$$\begin{aligned} D_2 D_1 L_{\text{d}}(q_k, q_{k+1}) = {} & \frac{\partial^2}{\partial q \partial q} L(q, \dot{q})(1-\alpha)\alpha\Delta t + \frac{\partial^2}{\partial \dot{q} \partial q} L(q, \dot{q})(1-\alpha) \\ & - \frac{\partial^2}{\partial q \partial \dot{q}} L(q, \dot{q})\alpha - \frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q, \dot{q}) \frac{1}{\Delta t} \end{aligned} \tag{2.72}$$

$$\begin{aligned} D_1 D_2 L_{\text{d}}(q_k, q_{k+1}) = {} & \frac{\partial^2}{\partial q \partial q} L(q, \dot{q})(1-\alpha)\alpha\Delta t - \frac{\partial^2}{\partial \dot{q} \partial q} L(q, \dot{q})\alpha \\ & + \frac{\partial^2}{\partial q \partial \dot{q}} L(q, \dot{q})(1-\alpha) - \frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q, \dot{q}) \frac{1}{\Delta t} \end{aligned} \tag{2.73}$$

$$\begin{aligned} D_2 D_2 L_{\text{d}}(q_k, q_{k+1}) = {} & \frac{\partial^2}{\partial q \partial q} L(q, \dot{q})\alpha^2\Delta t + \frac{\partial^2}{\partial \dot{q} \partial q} L(q, \dot{q})\alpha \\ & + \frac{\partial^2}{\partial q \partial \dot{q}} L(q, \dot{q})\alpha + \frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q, \dot{q}) \frac{1}{\Delta t} \end{aligned} \tag{2.74}$$

2.3.1.1 The Forced Case

External forces can also be incorporated into the derivation of the variational integrator. When considering continuous trajectories, the Lagrange-d'Alembert principle is used to generalize the Euler-Lagrange equation (2.61) by modifying the variation of the action, δS, to

$$\delta S[q(t)] = \delta \int_{t_0}^{t_{\text{f}}} L(q(\tau), \dot{q}(\tau))\ \mathrm{d}\tau + \int_{t_0}^{t_{\text{f}}} F(q(\tau), \dot{q}(\tau), u(\tau)) \cdot \delta q\ \mathrm{d}\tau \tag{2.75}$$

where $F(q(\tau), \dot{q}(\tau), u(\tau))$ represents the total external forcing acting on the system and u is the system's control input (if any). Minimization of the variational relation leads to the forced Euler-Lagrange equation:

$$\frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{q}}(q, \dot{q}) - \frac{\partial L}{\partial q}(q, \dot{q}) = F(q(t), \dot{q}(t), u(t). \tag{2.76}$$

Similar to the discretization of the Lagrangian, the left, $F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k)$, and right, $F_{\mathrm{d}}^{+}(q_k, q_{k+1}, u_k)$, discrete forces are introduce in order to obtain a discrete equivalent to (2.76). The variation of the continuous external force is approximated over a small time interval as

$$F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) \cdot \delta q_k + F_{\mathrm{d}}^{+}(q_k, q_{k+1}, u_k) \cdot \delta q_{k+1} \approx \int_{t_k}^{t_{k+1}} F(q(\tau), \dot{q}(\tau), u(\tau)) \cdot \delta q \ \mathrm{d}\tau \tag{2.77}$$

where a generalized midpoint approximation can be used to define the the left and right discrete forces as

$$F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) = \frac{1}{2} F((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1} - q_k}{\Delta t}, u_k)\Delta t, \tag{2.78}$$

$$F_{\mathrm{d}}^{+}(q_k, q_{k+1}, u_k) = \frac{1}{2} F((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1} - q_k}{\Delta t}, u_k)\Delta t, \tag{2.79}$$

and $u_k = u(t_k)$. The action sum (2.65) can then be modified and the resulting forced DEL equation is given as

$$D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) + D_2 L_{\mathrm{d}}(q_{k-1}, q_k) + F_{\mathrm{d}}^{+}(q_{k-1}, q_k, u_{k-1}) = 0. \tag{2.80}$$

or, equivalently, in the *position-momentum* form the forced DEL equation is given as

$$p_k + D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) = 0, \tag{2.81}$$

$$p_{k+1} = D_2 L_{\mathrm{d}}(q_k, q_{k+1}) + F_{\mathrm{d}}^{+}(q_k, q_{k+1}, u_k). \tag{2.82}$$

Note that p_k does not depend on q_{k+1} and in the unforced case p_k is the momentum quantity conserved by the integrator [58, 115]. Furthermore, the previously defined

two-step mapping $(q_{k-1}, q_k) \rightarrow (q_{k+1})$ can now be replaced with a one step mapping $(q_k, p_k) \rightarrow (q_{k+1}, p_{k+1})$. The integrator equation and its derivative are now defined as

$$f(q_{k+1}) = p_k + D_1 L_\mathrm{d}(q_k, q_{k+1}) + F_\mathrm{d}^-(q_k, q_{k+1}, u_k) \tag{2.83}$$

$$Df(q_{k+1}) = D_2 D_1 L_\mathrm{d}(q_k, q_{k+1}) + D_2 F_\mathrm{d}^-(q_k, q_{k+1}, u_k) \tag{2.84}$$

As before, given q_0, q_1, and the control input, $u(t)$, equation (2.83) can be solved iteratively to find $q_2, \ldots, q_n$.

2.3.1.2 The Constrained Case

Holonomic constraints can also be incorporated into the presented variational integrator. Specifically, the considered constraints are of the form

$$h(q) = [h_1(q), \ldots, h_m(q)]^\mathrm{T}, \tag{2.85}$$

where the system configuration is said to be valid if $h(q) = 0$. Holomonic constraints restrict the set of possible system configurations to lie in a sub-manifold. Therefore, during propagation the computed system configurations should lie in the desired sub-manifold. The forced DEL equations can now be modified to incorporate holonomic constraints [78]:

$$D_1 L_\mathrm{d}(q_k, q_{k+1}) + F_\mathrm{d}^-(q_k, q_{k+1}, u_k) + D_2 L_\mathrm{d}(q_{k-1}, q_k) + F_\mathrm{d}^+(q_{k-1}, q_k, u_{k-1}) = Dh^\mathrm{T}(q_k)\lambda_k, \tag{2.86}$$

$$h(q_{k+1}) = 0. \tag{2.87}$$

The term $Dh^\mathrm{T}(q_k)\lambda_k$ can be seen to represent a force that imposes the constraint and λ_k is the discrete Lagrange multiplier that defines the magnitude of this force. Note that the inclusion of the equation $h(q_{k+1}) = 0$ ensures that each discrete system configuration q_k observes the defined holomonic constraints. The integrator equation

and its derivative are now defined as

$$f(q_{k+1}, \lambda_k) = \begin{bmatrix} p_k + D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) - Dh^{\mathrm{T}}(q_k)\lambda_k \\ h(q_{k+1}) \end{bmatrix}, \quad (2.88)$$

$$Df(q_{k+1}, \lambda_k) = \begin{bmatrix} D_2 D_1 L_{\mathrm{d}}(q_k, q_{k+1}) + D_2 F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k) & -Dh^{\mathrm{T}}(q_k) \\ Dh(q_{k+1}) & 0 \end{bmatrix}. \quad (2.89)$$

The system configuration, (q_k, p_k), and the Lagrange multipliers, λ_k, are all propagated. The simple root finder algorithm is modified such that the estimate of the discrete Lagrangian multipliers are also updated:

$$\begin{bmatrix} q_{k+1} \\ \lambda_k \end{bmatrix} \leftarrow \begin{bmatrix} q_{k+1} \\ \lambda_k \end{bmatrix} - Df^{-1}(q_{k+1}, \lambda_k) \cdot f(q_{k+1}, \lambda_k) \quad (2.90)$$

2.3.1.3 Propagation of a Mass-Spring System

In order to further elucidate the reviewed variational integrator, consider a simple linear mass-spring system with unity mass and spring stiffness constant of 1 N/m. The Lagrangian of the system is computed as

$$L(q, \dot{q}) = \frac{1}{2}\dot{q}^2 - \frac{1}{2}q^2. \quad (2.91)$$

If $\alpha = 0.5$ then the resulting DEL equations in the *position-momentum* form are computed as

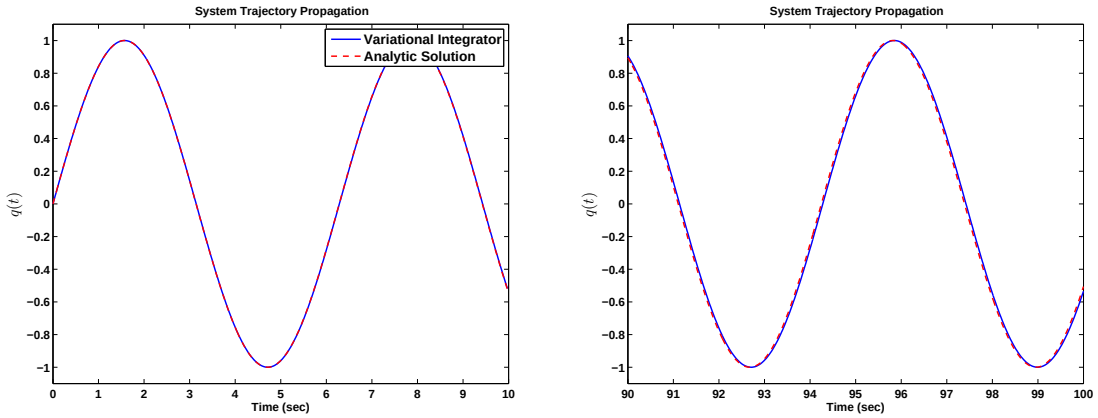
$$p_k - \frac{(q_{k+1} + q_k)}{4}\Delta t - \frac{q_{k+1} - q_k}{\Delta t} = 0, \quad (2.92)$$

$$p_{k+1} = -\frac{(q_{k+1} + q_k)}{4}\Delta t + \frac{q_{k+1} - q_k}{\Delta t}, \quad (2.93)$$

and the derivative of the integrator equation is

$$Df(q_{k+1}) = -\frac{1}{4}\Delta t - \frac{1}{\Delta t}. \quad (2.94)$$

It is now assumed that the initial conditions are given as $q(t_0) = 0$ and $\dot{q}(t_0) = 1$. In order to be consistent with the midpoint approximation used in this example ($\alpha = \frac{1}{2}$)



(a) (b)

Figure 2.3: (a) and (b): Propagation of the mass position, $q(t)$, with initial condition $q(t_0) = 0$ and $\dot{q}(t_0) = 1$ in time intervals $t = [0, 10]$ and $t = [90, 100]$, respectively.

initial conditions were set as $q_1 = \frac{1}{2}\dot{q}(t_0)\Delta t$ and $q_0 = -\frac{1}{2}\dot{q}(t_0)\Delta t$. Note that the selection of q_0 and q_1 is not unique.

The quantity p_1 can be found through equation (2.93) and equation (2.92) can then be used to obtain q_2. This process can be repeated indefinitely. Figure 2.3 shows the system trajectories solved using the analytic solution ($q(t) = \sin(t)$) and the variational integrator when $\Delta t = 0.02$. Note that the variational integrator is able to remain accurate despite a long propagation time.

2.3.2 Structured Linearization

As discussed in Reference 56 the one-step mapping (2.81) implicitly defines the function

$$x_{k+1} = g(x_k, u_k), \tag{2.95}$$

where $x_{k+1} = [p_{k+1}^{\mathrm{T}}, q_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$. In this section, it is shown that the linearization of the implicitly defined function $g(x_k, u_k)$ can be found explicitly (first shown in Reference 56). Furthermore, the linearization is computed using only derivatives of the discrete Lagrangian and forces. Specifically, the derived forced DEL equations (2.81) and (2.82) are used to obtain a first-order linearization of the discrete dynamics of the

form

$$\delta x_{k+1} = \nabla_x g(x_k, u_k)\delta x_k + \nabla_u g(x_k, u_k)\delta u_k, \tag{2.96}$$

or, equivalently,

$$\begin{bmatrix} \delta q_{k+1} \\ \delta p_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial q_{k+1}}{\partial q_k} & \frac{\partial q_{k+1}}{\partial p_k} \\ \frac{\partial p_{k+1}}{\partial q_k} & \frac{\partial p_{k+1}}{\partial p_k} \end{bmatrix} \begin{bmatrix} \delta q_k \\ \delta p_k \end{bmatrix} + \begin{bmatrix} \frac{\partial q_{k+1}}{\partial u_k} \\ \frac{\partial p_{k+1}}{\partial u_k} \end{bmatrix} \delta u_k. \tag{2.97}$$

To begin, equation (2.81) is implicitly differentiated with respect to q_k:

$$\begin{aligned} &\frac{\partial}{\partial q_k}[p_k + D_1 L_{k+1} + F^-_{k+1} = 0], \\ &0 + D_1 D_1 L_{k+1} + D_2 D_1 L_{k+1}\frac{\partial q_{k+1}}{\partial q_k} + D_1 F^-_{k+1} + D_2 F^-_{k+1}\frac{\partial q_{k+1}}{\partial q_k} = 0, \\ &\frac{\partial q_{k+1}}{\partial q_k} = -M^{-1}_{k+1}[D_1 D_1 L_{k+1} + D_1 F^-_{k+1}] \end{aligned} \tag{2.98}$$

where $M_{k+1} = D_2 D_1 L_{k+1} + D_2 F^-_{k+1}$ is assumed to be non-singular at q_k, p_k, and u_k. For ease of exposition, the discrete Lagrangian notation is condensed to

$$L_{k+1} = L_{\mathrm{d}}(q_k, q_{k+1}), \tag{2.99}$$

and that of the discrete forces to

$$F^{\pm}_{k+1} = F^{\pm}_{\mathrm{d}}(q_k, q_{k+1}, u_k). \tag{2.100}$$

Repeating this procedure yields

$$\frac{\partial q_{k+1}}{\partial p_k} = -M^{-1}_{k+1}, \tag{2.101}$$

$$\frac{\partial q_{k+1}}{\partial u_k} = -M^{-1}_{k+1} D_3 F^-_{k+1}. \tag{2.102}$$

The remaining derivatives can be found by explicitly differentiating (2.82)

$$\frac{\partial p_{k+1}}{\partial q_k} = [D_2 D_2 L_{k+1} + D_2 F^+_{k+1}]\frac{\partial q_{k+1}}{\partial q_k} + D_1 D_2 L_{k+1} + D_1 F^+_{k+1}, \tag{2.103}$$

$$\frac{\partial p_{k+1}}{\partial p_k} = [D_2 D_2 L_{k+1} + D_2 F^+_{k+1}]\frac{\partial q_{k+1}}{\partial p_k}, \tag{2.104}$$

$$\frac{\partial p_{k+1}}{\partial u_k} = [D_2 D_2 L_{k+1} + D_2 F^+_{k+1}]\frac{\partial q_{k+1}}{\partial u_k} + D_3 F^+_{k+1}. \tag{2.105}$$

Note that q_{k+1} is needed in order to evaluate the derivatives and can be found by solving (2.81). A linearization for the constrained case and second-order linearizations (constrained and unconstrained) are formulated in Reference 56. Section A.1 in the Appendix presents a detailed derivation of a variational integrator and its linearization for a nonlinear mass-spring-damper system.

2.3.3 Overview of the Variational Integrator and its Linearization

The reviewed variational integrator is used to accurately propagate the configuration of a (nonlinear) Hamiltonian system. Unlike algorithms that are used to numerically solve general ordinary differential equations, variational integrators preserve a system's characteristic properties by conserving (or nearly conserving) quantities like momentum and energy. The discrete equivalent of the Euler-Lagrange equations can be found through discretization of the system's Lagrangian and approximating the action integral with an action sum. The Least Action principle can then be applied to the resulting discrete approximation of the action integral to obtain the discrete Euler-Lagrange (DEL) equations. Unlike their continuous equivalents, the DEL equations result in an implicit two-step mapping $(q_{k-1}, q_k) \rightarrow (q_{k+1})$ that is used to propagate a system's configuration forward. The implicit mapping can be solved using a simple root finder algorithm. By appealing to the Lagrange-d'Alembert principle, external forces can also be incorporated into the discrete approximation. Finally, if the resulting integrator equation is placed in its position-momentum form a one-step mapping $(q_k, p_k) \rightarrow (q_{k+1}, p_{k+1})$ is obtained. A first-order linearization of the discrete system dynamics about a trajectory can also be computed through implicit and explicit differentiation of the integrator equation (in its position-momentum form).

2.4 Polynomial Chaos

First introduced by Wiener and later generalized, polynomial chaos expansion provides a manner in which second-order stochastic processes can be decomposed into

a infinite summation of polynomials [16, 91, 116, 119]. The methodology has been the subject of recent research due to its ability to quantify uncertainty in stochastic dynamical systems. Specifically, the effect of system uncertainty on the state trajectories can be characterized. Polynomial chaos has been used in robust control design, stability analysis, nonlinear estimation, and uncertainty analysis [26, 31–33, 48, 87].

To formally introduce polynomial chaos expansion consider the complete probability space Γ given by $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the set of events (σ-algebra on Ω), and P is the probability measure function. Furthermore, suppose $\lambda(\omega)$ is a random variable (ω is a random event) with probability density function $\rho(\lambda)$. A general second-order stochastic process $X(\lambda) \in \mathcal{L}_2(\Omega, \mathcal{F}, P)$ can be represented by a polynomial chaos expansion as

$$X(\lambda) = \sum_{i=0}^{\infty} x_i \phi_i(\lambda(\omega)), \tag{2.106}$$

where $x_i \in \mathbb{R}$ are the expansion coefficients and $\phi_i(\lambda(\omega)) \in \mathcal{L}_2(\Omega, \mathcal{F}, P)$ are orthogonal polynomials on Γ with respect to the probability density function, $\rho(\lambda)$, such that

$$\int_{\Gamma} \rho(\lambda)\phi_i(\lambda(\omega))\phi_j(\lambda(\omega)) \, \mathrm{d}\lambda = \delta_{ij}, \tag{2.107}$$

where δ_{ij} is the Kronecker delta product. If the probability density function $\rho(\lambda)$ is Gaussian then Hermite polynomials can be selected to be the orthogonal polynomials. Similarly, the following pairs of distributions and polynomials can be similarly associated: (Uniform, Legendre), (Gamma, Laguerre), and (Beta, Jacobi). Furthermore, the generalized polynomial chaos (gPC) framework allows for the construction of polynomial functions that are orthogonal with respect to arbitrary density functions. Note that the expansion weights, x_i, are computed as

$$x_i = \int_{\Gamma} \rho(\lambda) X(\lambda) \phi_i(\lambda(\omega)) \, \mathrm{d}\lambda. \tag{2.108}$$

Typically, in practice the expansion is truncated to a finite sum:

$$X(\lambda) \approx \sum_{i=0}^{r} x_i \phi_i(\lambda(\omega)). \tag{2.109}$$

The convergence of the sequence and its characteristics as r increases is a current area of research [27]. An attractive characteristic of the expansion is that the expectation and variance of the stochastic process $X(\lambda)$ are easily computed:

$$\mathrm{E}[X(\lambda)] = x_0, \tag{2.110}$$

$$\mathrm{Var}[X(\lambda)] = \sum_{i=i}^{\infty} x_i^2 \approx \sum_{i=i}^{r} x_i^2. \tag{2.111}$$

As a simple illustrative example, suppose $\lambda(\omega)$ is a standard normal deviate ($\lambda \sim \mathcal{N}(0,1)$), $X(\lambda) = A + B\lambda + C\lambda^2$, and $r = 2$ then

$$X(\lambda) \approx x_0 + x_1\lambda + x_2(\lambda^2 - 1) \tag{2.112}$$

where

$$x_0 = \int_\Gamma \rho(\lambda)\big(A + B\lambda + C\lambda^2\big)\,\mathrm{d}\lambda = A + C \tag{2.113}$$

$$x_1 = \int_\Gamma \rho(\lambda)\big(A\lambda + B\lambda^2 + C\lambda^3\big)\,\mathrm{d}\lambda = B \tag{2.114}$$

$$x_2 = \int_\Gamma \rho(\lambda)\big(A(\lambda^2 - 1) + B(\lambda^3 - \lambda) + C(\lambda^4 - \lambda^2)\big)\,\mathrm{d}\lambda = 2C, \tag{2.115}$$

and, therefore,

$$X(\lambda) \approx A + B\lambda + 2C\lambda^2 - C. \tag{2.116}$$

If $\lambda(\omega) = [\lambda_1(\omega), \lambda_2(\omega)]^{\mathrm{T}}$ is a random *vector* the approximate expansion is modified such that [110]

$$X(\lambda_1, \lambda_2) \approx \sum_{(i,j):(0\le i+j\le p)} x_i^1 \phi_i^1(\lambda_1) x_j^2 \phi_j^2(\lambda_2), \tag{2.117}$$

where p is the largest polynomial degree in the expansion. Equivalently, a set of orthogonal polynomials can be defined as $\phi_k(\lambda_1, \lambda_2) = \phi_i^1(\lambda_1)\phi_j^2(\lambda_2)$ such that the approximate expansion is expressed as

$$X(\lambda) \approx \sum_{k=0}^{L} x_k \phi_k(\lambda_1, \lambda_2), \tag{2.118}$$

where $L = \frac{(n+p)!}{n!p!} - 1$ and $x_k = x_i x_j$. Generalizing to the case where $\lambda(\omega)$ is of arbitrary size is straight forward.

III

A STOCHASTIC VARIATIONAL INTEGRATOR

"In proving foresight may be vain:
The best-laid schemes o' mice an' men
Gang aft agley,
An' lea'e us nought but grief an' pain,
For promis'd joy!"

– Robert Burns, *To a Mouse*

In this chapter a variational integrator for stochastic dynamical Hamiltonian systems is formulated. The system's stochasticity is represented using state and control dependent diffusion vector fields and Brownian motion. The resulting stochastic differential equation (SDE) is evaluated in the Stratonovich sense. As before, the variational integrator provides an implicit one-step mapping from the current system configuration to the next. Using the methodology reviewed in Chapter 2, the implicit mapping is utilized in order to obtain a stochastic first-order linearization of system dynamics about a trajectory.

The benefits of using the formulated variational integrator and its linearization in an iterative optimization process are then investigated. Through numerical experiments it is shown that the stochastic differential dynamical programming (S-DDP) algorithm (reviewed in Chapter 2) becomes less dependent on the discretization time step and more predictable when it utilizes the proposed integrator. Furthermore, it is shown that a significant reduction in computational time can be achieved without sacrificing the algorithm's performance. Therefore, the proposed variational integrator can be used to enable real-time implementation of nonlinear optimal control algorithms.

The benefits of using the proposed stochastic variational integrator in the extended Kalman filter (EKF) algorithm is also investigated. As in the S-DDP case, the EKF algorithm is shown to be less dependent on the discretization time step when the proposed integrator and its linearization are used. Furthermore, it is also shown that the algorithm remains stable when sensor updates become less frequent. Therefore, the proposed integrator may reduce the computational, power, and sensing requirements of systems by allowing the use of large discretization time steps and reduced observation update rates.

Throughout the chapter the presented methodology is compared with the standard Euler method. The Euler method relies on first-order approximations in order to propagate system configurations forward and obtain linearizations of system dynamics about a trajectory. Specifically, consider the equations describing a stochastic Hamiltonian system of the form

$$\mathrm{d}x = f(x)\ \mathrm{d}t + \sum_{i=1}^{p} F_i(x)\ \mathrm{d}\omega_i \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the system state and $d\omega_i$ is independent Brownian noise. The resulting explicit integration equation using the Euler method is

$$x_{k+1} = x_k + f(x_k, u_k)\Delta t + \sum_{i=1}^{p} F_i(x_k, u_k)\Delta\omega_{ik}, \tag{3.2}$$

and its linearization is given by

$$\begin{aligned}\delta x_{k+1} = (\mathrm{I} + \nabla_{x_k} f(x_k, u_k)\Delta t + \sum_{i=1}^{p} \nabla_{x_k} F_i(x_k, u_k)\Delta\omega_{ik})\delta x_k \\ + (\nabla_{u_k} f(x_k, u_k)\Delta t + \sum_{i=1}^{p} \nabla_{u_k} F_i(x_k, u_k)\Delta\omega_{ik})\delta u_k + \sum_{i=1}^{p} F_i(x_k, u_k)\Delta\omega_{ik},\end{aligned} \tag{3.3}$$

where Δt is the discretization time step and $\Delta\omega_{ik} \sim \mathcal{N}(0, \Delta t)$ are independent random variables. It should be noted that the Euler method was considered due to its universal use and its straightforward linearization. Comparisons to commonly used multi-step methods (Runge-Kutta, Rosenbrock, etc) is an area of future research [1, 52, 98].

3.1 Formulation

This section presents a stochastic variational integrator used to obtain accurate discretized trajectories of stochastic dynamical Hamiltonian systems given as [11, 84]

$$\mathrm{d}\frac{\partial L(q,\dot{q})}{\partial \dot{q}} = \frac{\partial L(q,\dot{q})}{\partial q}\,\mathrm{d}t + F_0(q,\dot{q},u)\,\mathrm{d}t + \sum_{i=1}^{p} F_i(q,\dot{q},u)\,\circledS\,\mathrm{d}\omega_i,$$

$$q(0) = q_0, \quad \dot{q}(0) = \dot{q}_0, \tag{3.4}$$

where $L(q,\dot{q})$ is the mechanical system's Lagrangian given as the difference between the system's kinetic energy, $T(q,\dot{q})$, and potential energy, $V(q)$, such that $L(q,\dot{q}) = T(q,\dot{q}) - V(q)$, q is the state configuration vector, $\dot{q}$ is the time derivative of the state configuration vector, $\mathrm{d}\omega_i,\ i = 1,\ldots,m$ are independent Brownian noises, $F_i(q,\dot{q}),\ i = 1,\ldots,m$ are the diffusion vector fields, $F_0(q,\dot{q},u)$ is the forcing function representing deterministic non-conservative external forces, and $\circledS$ indicates the stochastic integral is evaluated in the Stratonovich sense [66]. A variational integrator computes a discretized trajectory $\mathbf{q} = \{q_0,\ldots,q_N\}$ that approximates the systems trajectory $q(t)$ such that $q_k \approx q(t_k)$, where $t_0 = 0$, $t_N = t_\mathrm{f}$ and $t_{k+1} - t_k = \Delta t$.

The section also presents the derivation for a first-order linearization of the presented variational integrator and state-space realizations of stochastic Hamiltonian systems. A complete and rigorous treatment of variational integrators was presented in References 40 and 77. Scalable variational integrators were implemented to generic mechanical systems in generalized coordinates in Reference 58. First- and second-order linearizations of a deterministic variational integrator was formulated in Reference 56. The presented stochastic variational integrator and its first-order linearization are based on the deterministic variational integrator and its linearization reported in Reference 58 and 56. In addition, stochastic variational integrators have also been investigated in References 11, 84, and 114.

3.1.1 Stochastic Variational Integrators

As in Chapter 2, the discrete Lagrangian, $L_{\mathrm{d}}(q_k, q_{k+1})$, is introduced to obtain an approximation of the action integral over a short interval,

$$L_{\mathrm{d}}(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(q(s), \dot{q}(s))\ \mathrm{d}s. \tag{3.5}$$

The generalized midpoint approximation is used to formulate L_{d} as $L_{\mathrm{d}}(q_k, q_{k+1}) = L\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}\big)\Delta t$, where $\alpha \in [0,1]$ and $\alpha = 1/2$ results in second order accuracy as discussed in [115]. Left and right discrete forces, $F_{\mathrm{d}}^{-}(q_k, q_{k+1}, u_k)$ and $F_{\mathrm{d}}^{+}(q_k, q_{k+1}, u_k)$, approximate the non-conservative forces as

$$\int_{t_k}^{t_{k+1}} F_0(q(s), \dot{q}(s), u(s)) \cdot \delta q\ \mathrm{d}s \approx F_{0,\mathrm{d}}^{-}/2 \cdot \delta q_k + F_{0,\mathrm{d}}^{+}/2 \cdot \delta q_{k+1}, \tag{3.6}$$

and the Stratonovich integral used to represent the stochastic effects is converted to its equivalent Itō's representation (see Reference 66) and then approximated as

$$\begin{aligned}\int_{t_k}^{t_{k+1}} F_i(q, \dot{q}, u) \cdot \delta q\ \circledS\ \mathrm{d}\omega_i &= \frac{1}{2}\int_{t_k}^{t_{k+1}} F_i' F_i \cdot \delta q\ \mathrm{d}t + \int_{t_k}^{t_{k+1}} F_i \cdot \delta q\ \mathrm{d}\omega_i \\ &\approx {F'}_{i,\mathrm{d}}^{-} F_{i,\mathrm{d}}^{-}/4 \cdot \delta q_k + {F'}_{i,\mathrm{d}}^{+} F_{i,\mathrm{d}}^{+}/4 \cdot \delta q_{k+1} + F_{i,d}^{s} \Delta\omega_{i_k} \cdot \delta q_k,\end{aligned}$$

where

$$F_{0,\mathrm{d}}^{\pm} = F_0\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}, u_k\big)\Delta t, \tag{3.7}$$

$$F_{j,\mathrm{d}}^{\pm} = F_j\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}, u_k\big)\Delta t, \tag{3.8}$$

$${F'}_{j,\mathrm{d}}^{\pm} = F_j'\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}, u_k\big), \tag{3.9}$$

$$F_{j,\mathrm{d}}^{s} = F_j\big(q_k, \frac{q_k - q_{k-1}}{\Delta t}, u_k\big), \tag{3.10}$$

$u_k = u(t_k)$, and $\Delta\omega_{i_k} = \omega_i(t_{k+1}) - \omega_i(t_k) \sim \mathcal{N}(0, \Delta t)$. The discrete form of the Lagrange-d'Alembert principle can be used to relate the derived approximations as

$$\delta \sum_{k=0}^{N-1} L_{k+1} + \frac{1}{2}\sum_{k=0}^{N-1}\big(F_{k+1}^{-} \cdot \delta q_k + F_{k+1}^{+} \cdot \delta q_{k+1}\big) + \sum_{k=0}^{N-1} F_k^s \Delta\omega_k \cdot \delta q_k = 0, \tag{3.11}$$

where $F_k^{\pm} = F_{0,\mathrm{d}}^{\pm}(q_{k-1}, q_k, u_{k-1}) + \sum_{i=1}^{p} F'^{\pm}_{i,\mathrm{d}}(q_{k-1}, q_k, u_{k-1}) F_{i,\mathrm{d}}^{\pm}(q_{k-1}, q_k, u_{k-1})/2$, $F_k^s = [F_{1,\mathrm{d}}^s(q_{k-1}, q_k, u_k), \ldots, F_{p,\mathrm{d}}^s(q_{k-1}, q_k, u_k)]$, $L_k = L_{\mathrm{d}}(q_{k-1}, q_k)$, and $\Delta\omega_k = [\Delta\omega_{1,k}, \ldots, \Delta\omega_{p,k}]^{\mathrm{T}}$. Solving equation (3.11) leads to a forced Discrete Euler-Lagrange (DEL) equation

$$D_2 L_k + D_1 L_{k+1} + F_k^+ + F_{k+1}^- + F_k^s \Delta\omega_k = 0. \tag{3.12}$$

The integrator equation can now be defined as

$$f(q_{k+1}) = p_k + D_1 L_{k+1} + F_{k+1}^- + F_k^s \Delta\omega_k, \tag{3.13}$$

where, for computation convenience,

$$p_k = D_2 L_k + F_k^+. \tag{3.14}$$

Note that p_k does not depend on q_{k+1} and, in the unforced case, p_k is the generalized momentum quantity conserved by the integrator as discussed in Reference 58. Furthermore, the derivative of (3.13) with respect to q_{k+1} is given as

$$Df(q_{k+1}) = D_2 D_1 L_{k+1} + D_2 F_{k+1}^-. \tag{3.15}$$

The integrator equation (3.13) is an implicit one-step mapping $(q_k, p_k) \to (q_{k+1}, p_{k+1})$ and is solved using Algorithm 3 (see Chapter 2) as shown in Reference 58. Note that if q_0 and q_1 are known then $q_k,\ k \geq 1$ can be obtained by utilizing the presented integrator.

3.1.2 Stochastic Structured Linearization

As in Chapter 2, the derived forced DEL equation (3.12) can be used to obtain a first-order linearization of the discrete dynamics of the form[1]

$$\begin{bmatrix} \delta q_{k+1} \\ \delta p_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial q_{k+1}}{\partial q_k} & \frac{\partial q_{k+1}}{\partial p_k} \\ \frac{\partial p_{k+1}}{\partial q_k} & \frac{\partial p_{k+1}}{\partial p_k} \end{bmatrix} \begin{bmatrix} \delta q_k \\ \delta p_k \end{bmatrix} + \begin{bmatrix} \frac{\partial q_{k+1}}{\partial u_k} \\ \frac{\partial p_{k+1}}{\partial u_k} \end{bmatrix} \delta u_k + \begin{bmatrix} \frac{\partial q_{k+1}}{\partial \Delta\omega_k} \\ \frac{\partial p_{k+1}}{\partial \Delta\omega_k} \end{bmatrix} \delta \Delta\omega_k. \quad (3.16)$$

To begin the forced DEL equation is represented in its equivalent position-momentum form as (see [56] for further details)

$$p_k + D_1 L_{k+1} + F^-_{k+1} + F^s_k \Delta\omega_k = 0, \quad (3.17)$$

$$p_{k+1} = D_2 L_{k+1} + F^+_{k+1}. \quad (3.18)$$

Implicitly differentiating equation (3.17) with respect to q_k yields

$$\begin{aligned} &\frac{\partial}{\partial q_k}[p_k + D_1 L_{k+1} + F^-_{k+1} + F^s_k \Delta\omega_k = 0], \\ &0 + D_1 D_1 L_{k+1} + D_2 D_1 L_{k+1} \frac{\partial q_{k+1}}{\partial q_k} + D_1 F^-_{k+1} + D_2 F^-_{k+1} \frac{\partial q_{k+1}}{\partial q_k} + D_2 F^s_k \Delta\omega_k = 0, \\ &\frac{\partial q_{k+1}}{\partial q_k} = -M^{-1}_{k+1}[D_1 D_1 L_{k+1} + D_1 F^-_{k+1} + D_2 F^s_k \Delta\omega_k] \end{aligned} \quad (3.19)$$

where $M_{k+1} = D_2 D_1 L_{k+1} + D_2 F^-_{k+1}$ is assumed to be non-singular at q_k, p_k, and u_k. Repeating this procedure yields

$$\frac{\partial q_{k+1}}{\partial p_k} = -M^{-1}_{k+1}, \quad (3.20)$$

$$\frac{\partial q_{k+1}}{\partial u_k} = -M^{-1}_{k+1}[D_3 F^-_{k+1} + D_3 F^s_k \Delta\omega_k], \quad (3.21)$$

$$\frac{\partial q_{k+1}}{\partial \Delta\omega_k} = -M^{-1}_{k+1} F^s_k. \quad (3.22)$$

[1] As discussed in [58] and in Chapter 2, the one-step mapping (3.13) implicitly defines a function $g(x_k, u_k, \Delta\omega_k)$ such that $x_{k+1} = g(x_k, u_k, \Delta\omega_k)$ where $x_{k+1} = [p^{\mathrm{T}}_{k+1}, q^{\mathrm{T}}_{k+1}]^{\mathrm{T}}$. The linearization is equivalently defined as $\delta x_{k+1} = \nabla_{x_k} g(x_k, u_k, \Delta\omega_k)\delta x_k + \nabla_{u_k} g(x_k, u_k, \Delta\omega_k)\delta u_k + \nabla_{\Delta\omega} g(x_k, u_k, \Delta\omega_k)\Delta\omega_k$.

The remaining derivatives can be found by explicitly differentiating (3.18)

$$\frac{\partial p_{k+1}}{\partial q_k} = [D_2 D_2 L_{k+1} + D_2 F_{k+1}^+] \frac{\partial q_{k+1}}{\partial q_k} + D_1 D_2 L_{k+1} + D_1 F_{k+1}^+, \tag{3.23}$$

$$\frac{\partial p_{k+1}}{\partial p_k} = [D_2 D_2 L_{k+1} + D_2 F_{k+1}^+] \frac{\partial q_{k+1}}{\partial p_k}, \tag{3.24}$$

$$\frac{\partial p_{k+1}}{\partial u_k} = [D_2 D_2 L_{k+1} + D_2 F_{k+1}^+] \frac{\partial q_{k+1}}{\partial u_k} + D_3 F_{k+1}^+, \tag{3.25}$$

$$\frac{\partial p_{k+1}}{\partial \Delta\omega_k} = [D_2 D_2 L_{k+1} + D_2 F_{k+1}^+] \frac{\partial q_{k+1}}{\partial \Delta\omega_k}. \tag{3.26}$$

Note that q_{k+1} is needed in order to evaluate the derivatives and can be found by solving (3.13). Furthermore, note that, in general, the partial derivatives with respect to q_k and u_k are stochastic. However, these derivatives are affine in $\Delta\omega_k$ (see (3.19) and (3.21)). That is, the derivatives are of the form $A + B\Delta\omega_k$ where A and B are deterministic. Therefore, the expectation of the linearization of the discrete dynamics (3.16) can be found.

It should be noted that F_k^s was formulated in equations (3.10) and (3.11) such that it did not depend on q_{k+1}. This was possible since the Stratonovich integral was converted to its equivalent Itō's representation. Note that by invoking this equivalence, we replace an explicit dependence of q_{k+1} with a projection of stochastic effects on the system, captured by the term $\frac{1}{2}\int_{t_k}^{t_{k+1}} F_i' F_i \cdot \delta q \, \mathrm{d}t$. As a result, M_{k+1} is not stochastic and, therefore, ratios of stochastic quantities are avoided (e.g. $\frac{\partial q_{k+1}}{\partial q_k} = -M_{k+1}^{-1}[D_1 D_1 L_{k+1} + D_1 F_{k+1}^- + D_2 F_k^s \Delta\omega_k]$). The expectation and central moments of the linearization are well-defined and computed easily. Note that the change in the representation of the stochastic effects avoids ratios of stochastic quantities, but this is not to say that using a Stratonovich representation in this case is ill-posed. Conversion between a Stratonovich and a Itō representation (or another representation) is often done when one definition is more convenient.

3.2 Application to the Stochastic DDP Algorithm

3.2.1 3-Link Planar Manipulator

This example demonstrates the benefits of using the proposed stochastic variational integrator and its linearization when implementing the S-DDP algorithm. An experiment involving the control of a dynamical system representing a human finger (3-link planar manipulator) (see [71] for system model and parameters) is considered[2]. It is shown that the solution of the S-DDP algorithm when utilizing the variational integrator is far less dependent on the discretization step size than when the Euler method is used. The lack of dependence allows the algorithm to obtain a solution utilizing a relatively large step size that approximates a solution acquired utilizing a small step size. As a result, the computational time of the algorithm can be significantly reduced without degrading its performance. All experiments were conducted using MATLAB scripts.

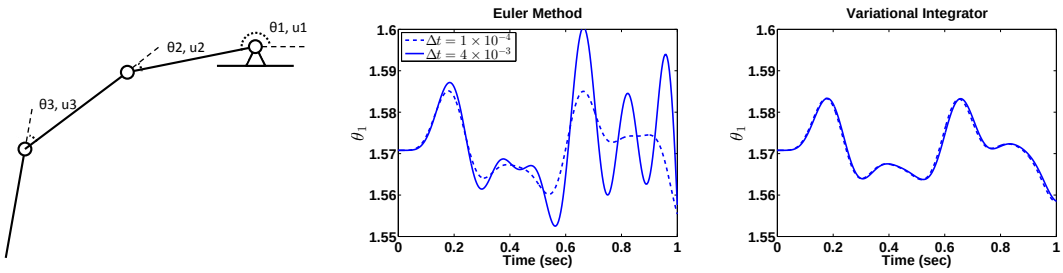
As shown in Figure 3.1a the dynamical system is described by three coordinates given by the relative angles between adjacent links, $\theta(t) = [\theta_1(t), \theta_2(t), \theta_3(t)]$, and three control inputs, $u(t) = [u_1(t), u_2(t), u_3(t)]$. Figures 3.1b and 3.1c show a comparison between the propagation of $\theta_1(t)$ using the Euler method and the presented variational integrator with two different step sizes subject to initial conditions $\theta(t_0) = [\pi/2, 0, 0]$ and $\dot{\theta}(t_0) = [0, 0, 1]$ and $u(t) = 0$. Note that the accuracy of the Euler method is degraded when the discretization step size is increased while the variational integrator is negligibly affected.

For all cases considered the reference tracking based cost function is

$$J(t) = \int_{t_0}^{t_\mathrm{f}} [(\theta(\tau) - \theta_t(\tau))^2 + u(\tau)^2] \, \mathrm{d}\tau + (\theta(t_\mathrm{f}) - \theta_t(t_\mathrm{f}))^2, \tag{3.27}$$

where $\theta_t(t) = [1{+}0.25\sin(4\pi t), 0.1\sin(2\pi t), 0]$ and $t_\mathrm{f}{-}t_0 = 1.5$. The S-DDP algorithm parameters (see Algorithm 1 in Chapter 2) were set to $\epsilon = 1 \times 10^{-4}$, $\alpha = 1 \times 10^{-8}$,

[2]The variational integrator and its linearization for a 3-link planar manipulator is formulated in Appendix A.



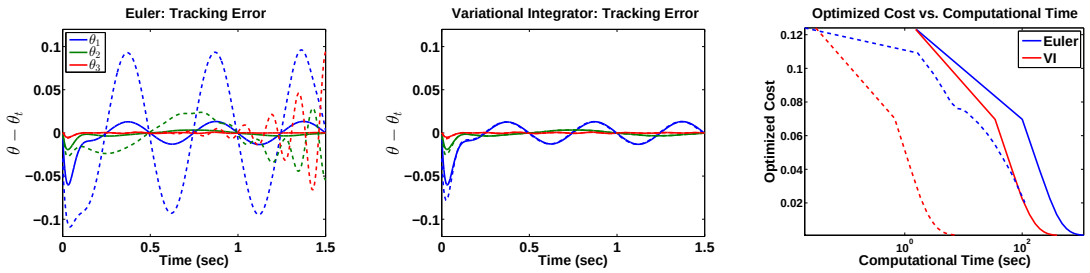
(a) (b) (c)

Figure 3.1: (a): Diagram of the studied dynamical system. (b) and (c): Propagation of the $\theta_1(t)$ with initial condition $\theta(t_0) = [\pi/2, 0, 0]$ and $\dot{\theta}(t_0) = [0, 0, 1]$ subject to input $u(t) = 0$. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-3}$ while solid lines indicate $\Delta t = 4 \times 10^{-3}$.

and $\beta = 0.25$.

In order to examine the performance of the variational integrator in a variety of situations three cases were considered: white noise, input-dependent noise, and state-dependent noise. In the first case, $F_i(q, \dot{q}, u) = Ce_i,\ C = 5 \times 10^{-5},\ i = \{1, 2, 3\}$.[3] As discussed in Reference 111 if the additive noise is neither state nor control dependent the S-DDP algorithm is equivalent to the classic DDP solution. Figures 3.2a and 3.2b display the calculated optimal solutions using the Euler method and the variational integrator with two different step sizes. It is concluded that the Euler method is far more dependent on the discretization step size. Figure 3.2c shows the optimized cost as a function of the running computational time of the S-DDP algorithm. Note that when the variational integrator is used the required computational time can be drastically reduced without significantly changing the optimized cost. Table 3.1 and Figure 3.5a show that the optimized cost increases much more rapidly as the discretization time step is increased when the Euler method is used. Additionally, the number of algorithm iterations is more predictable when the variational integrator is used. Depending on the allowable deviation from the true optimal solution (assuming it is obtained when $\Delta t = 0.0001$) the required computational time can be drastically

[3]The standard basis is used to define vectors $e_i, i \in \{1, 2, 3\}$ (e.g. $e_1 = [1, 0, 0]^{\mathrm{T}}$).



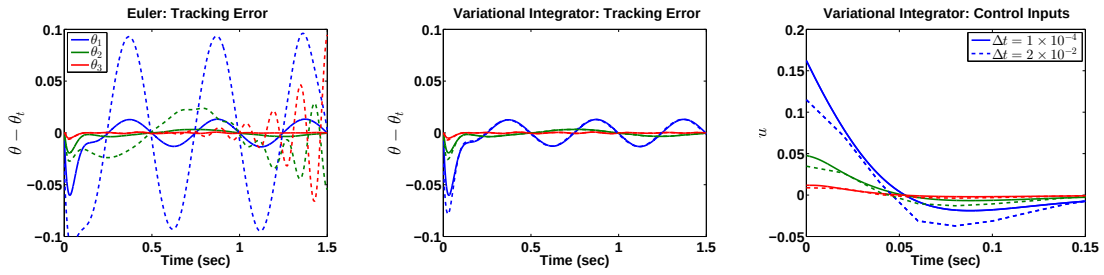
(a) (b) (c)

Figure 3.2: (a) and (b): Tracking error of the system states in Case 1. (c): Optimized cost versus the running computational time of the S-DDP algorithm (computational time plotted on a logarithmic scale). Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 6 \times 10^{-3}$.

reduced by utilizing a larger step size when the variational integrator is used. Finally, it should be noted that the growth of the optimized cost when utilizing the presented variational integrator is cause, in part, by the reduction of the controller's bandwidth. However, how much of the growth can be attributed to the change in the controller's bandwidth and not other effects (integrator accuracy, numerical precision, etc.) is an area of future research.

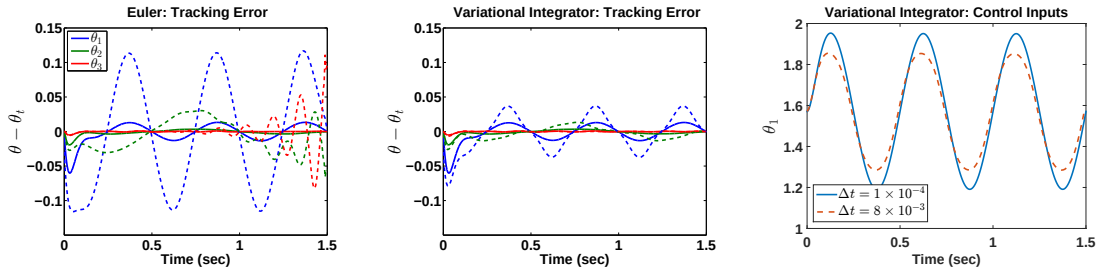
The next case considers input-dependent noise where $F_i(q, \dot{q}, u) = Ce_i u_i^2(t),\ i = \{1, 2, 3\}$, $C = 5 \times 10^{-4}$. As shown in the first case, Figures 3.3a and 3.3b show that the optimized solution obtained utilizing the Euler method is far more dependent on the discretization step size. Furthermore, note that since $\Delta\omega_{i_k} \sim \mathcal{N}(0, \Delta t)$ the perceived amount of noise at discretization points increases with Δt. Therefore, as shown in Figure 3.3c the optimal control input will be dependent on the time step since it is advantageous to reduced the amount of induced noise. As a result, the optimized cost should increase as the time step increases. However, how much of the growth can be attributed to increases in noise and not other effects already discussed is unknown. Nevertheless, as shown in Table 3.1 and Figure 3.5b the growth in the cost is far greater when the Euler method is used.

State-dependent noise is considered in the final case, $F_i(q, \dot{q}, u) = C(\theta_i(t) -$



(a) (b) (c)

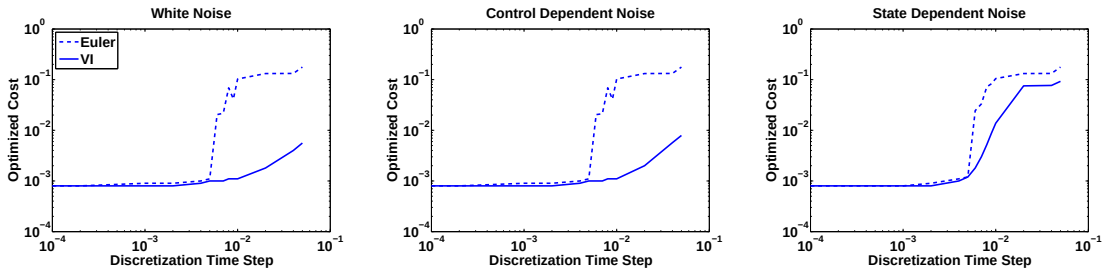
Figure 3.3: (a) and (b): Tracking error of the system states in Case 2. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 6 \times 10^{-3}$. (c): Optimal control inputs when the variational integrator method is used (initial 0.1 seconds shown). Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 2 \times 10^{-2}$.



(a) (b) (c)

Figure 3.4: (a) and (b): Tracking error of the system states in Case 3. (c): Optimal θ_1 trajectories when the variational integrator is used. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 8 \times 10^{-3}$.

$\theta_i(t_0))^2$, $i = \{1, 2, 3\}, C = 5 \times 10^{-4}$. Figures 3.3a and 3.3b display the calculated optimal solutions using the Euler method and the variational integrator with two different step sizes. Note both methods are strongly dependent on the discretization step size. As in Case 2, the perceived amount of noise at discretization points increases with Δt. In this case, it is advantageous to keep the system near the initial equilibrium state. Therefore, a trade-off between the tracking performance and the amount of induce noise occurs. As illustrated in Figure 3.4, the optimized state trajectory is dependent on the step size. However, as shown in Table 3.1 the growth in the optimized cost is greater when the Euler method is used.



(a) (b) (c)

Figure 3.5: Optimized cost as a function of the discretization time step. Dotted lines indicate the Euler method was used and solid lines indicate that the variational integrator was used.

Table 3.1: Summary of Data Collected from Numerical Experiments

Δt	0.0001	0.001	0.002	0.005	0.006	0.007	0.008	0.009	0.01	0.02
White Noise, Euler										
Cost	0.0008	0.0009	0.0009	0.0011	0.0204	0.0217	0.0689	0.0413	0.1031	0.1320
CPU Time (sec)	1119.7	107.7	62.0	56.6	112.5	175.4	26.5	240.6	26.4	2.3
Iterations	13	13	14	31	71	130	23	236	31	6
White Noise, VI										
Cost	0.0008	0.0008	0.0008	0.0010	0.0010	0.0010	0.0011	0.0011	0.0011	0.0018
CPU Time (sec)	393.5	39.4	19.9	8.5	7.2	6.0	5.3	4.6	4.3	2.4
Iterations	13	13	13	13	13	13	13	13	14	14
Input-Dependent, Euler										
Cost	0.0008	0.0009	0.0009	0.0011	0.0204	0.0212	0.0690	0.0410	0.1031	0.1320
CPU Time (sec)	989.2	98.1	54.1	51.1	100.2	154.4	23.3	224.0	24.8	2.5
Iterations	13	13	14	31	71	131	23	241	31	6
Input-Dependent, VI										
Cost	0.0008	0.0008	0.0008	0.0010	0.0010	0.0010	0.0011	0.0011	0.0011	0.0020
CPU Time (sec)	412.1	38.1	19.1	7.8	6.4	5.3	4.8	4.6	4.2	2.2
Iterations	13	13	13	13	13	13	13	13	14	14
State-Dependent, Euler										
Cost	0.0008	0.0008	0.0009	0.0012	0.0244	0.0332	0.0720	0.0831	0.1051	0.1320
CPU Time (sec)	990.4	104.9	52.8	48.4	86.0	128.0	95.4	64.6	24.3	1.9
Iterations	13	14	14	31	63	107	92	71	28	6
State-Dependent, VI										
Cost	0.0008	0.0008	0.0008	0.0012	0.0018	0.0030	0.0052	0.0088	0.0138	0.0754
CPU Time (sec)	371.3	37.9	20.2	9.3	7.5	6.4	5.9	4.9	4.1	1.1
Iterations	13	13	14	15	15	15	15	15	14	4

3.2.2 Effect of Noise Intensity

In this example the effect of noise intensity on the optimized trajectory and input is studied. In order to focus on the fundamental relationship of noise intensity and the optimized trajectory and input, a simple nonlinear mass-spring-damper system is considered in this example. Specifically, consider a system with unity mass, a spring force given as $f_s = -5x^3$, and a damping force given as $f_s = -0.1\dot{x}^4$. Furthermore, the reference-tracking based cost given as

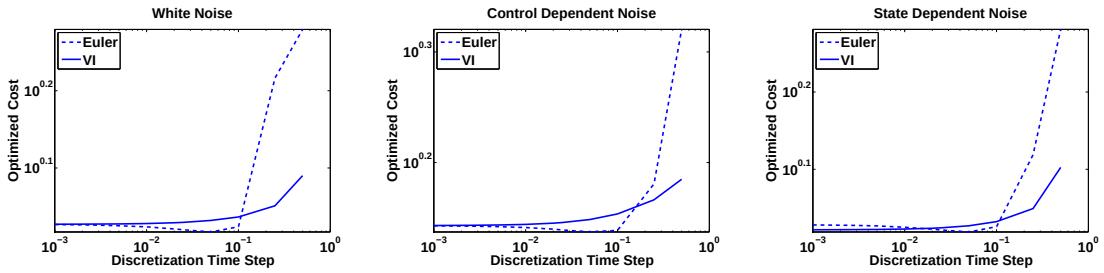
$$v(x,u,t) = \int_{t_0}^{t_\mathrm{f}} \left(5(x(\tau) - x_\mathrm{ref}(\tau))^2 + (\dot{x}(\tau) - \dot{x}_\mathrm{ref}(t))^2 + u^2(\tau)\right) \mathrm{d}\tau. \tag{3.28}$$

As before, three different cases are considered. The additive noise is parameterized when white noise, input-dependent noise, and state-dependent noise as $F(q,\dot{q},u) = C_1x(t)^2$, $F(q,\dot{q},u) = C_2u(t)^2$, and $F(q,\dot{q},u) = C_3x(t)^2$, respectively. As in the previous example, Figure 3.6 shows that the S-DDP algorithm is far less dependent on the discretization time step when the variational integrator is used.

The effect of noise intensity on the optimized input and time varying feedback gain when input-dependent noise is considered is shown in Figure 3.7 ($\Delta t = 1 \times 10^{-3}$)[5]. Note that as noise intensity increases, the magnitude of the optimized input tends to be smaller. Furthermore, the feedback gain can also be seen to be dependent on the intensity. When $C_2 = 2.0$, there is a large spike in the time-varying feedback gain at around $t = 1$. Figures 3.8 and 3.9 display the distribution of the response of the system over 1500 Monte Carlo simulations when the optimal control policy (feedforward and feedback) was implemented. As expected, increasing the noise intensity results in a larger standard deviation in the observed trials. However, notice that the effect of the noise is mitigated by the optimized feedback controller.

[4]Section A.1 in the Appendix presents a detailed derivation of a variational integrator and its linearization for a nonlinear mass-spring-damper system.

[5]Note that in the case of white noise, as shown by Equation 2.32 and in Reference 111, the optimal solution obtain by the SDDP algorithm is identical to the one obtained in the deterministic case. Therefore, the effect of noise intensity in this case is not investigated



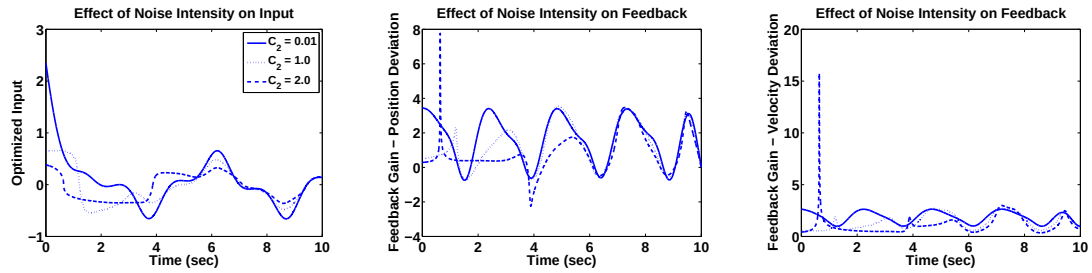
(a) (b) (c)

Figure 3.6: Optimized cost as a function of the discretization time step. $C_3 = 1$, $C_2 = 1$, and $C_1 = 0.5$ when white noise, input-dependent noise, and state-dependent noise were considered, respectively. Dotted lines indicate the Euler method was used and solid lines indicate that the variational integrator was used.

The effect of noise intensity on the optimized trajectory and time varying feedback gain when input-dependent noise is considered is shown in Figure 3.10 ($\Delta t = 1 \times 10^{-3}$). Note that as noise intensity increases, the magnitude of the optimized input tends to be smaller. Furthermore, the feedback gain is allowed to become more aggressive as the noise intensity increases. Since the optimized trajectory becomes more conservative a more aggressive feedback gain is needed to ensure performance is maintained. Intuitively, as noise intensity increases the predictive response of the system becomes less informative and, as a result, the feedback gain is relied on more. Figures 3.11 and 3.12 display the distribution of the response of the system over 1500 Monte Carlo simulations when the optimal control policy was implemented. As before, increasing the noise intensity results in a larger standard deviation in the observed trials. Nevertheless, the computed control policy was able to mitigate the effects of the system's stochasticity.

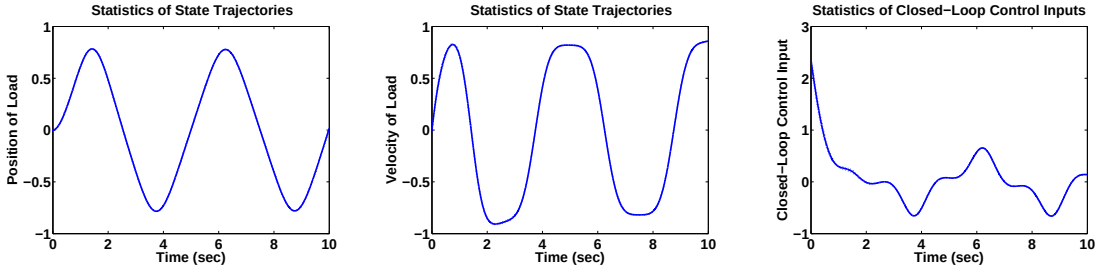
3.3 Application to the Extended Kalman Filter Algorithm

In this section the benefits of using the proposed stochastic variational integrator and its linearization when implementing the extended Kalman filter algorithm are demonstrated [37, 105]. As in the previous section, an experiment involving the control of a dynamical system representing a human finger (3-link planar manipulator) (see [71]



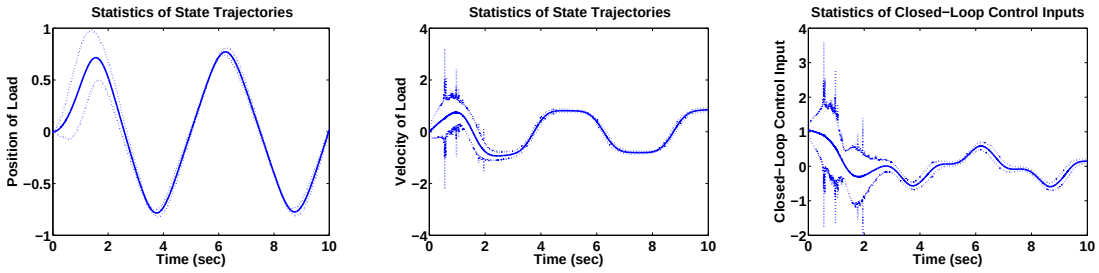
(a) (b) (c)

Figure 3.7: The computed optimized input and feedback gains for different noise parameterizations. Solid, dotted, and dashed lines indicate that the input-dependent noise was parameterized as $C_2 = 0.01$, $C_2 = 1.0$ and $C_2 = 2.0$, respectively.



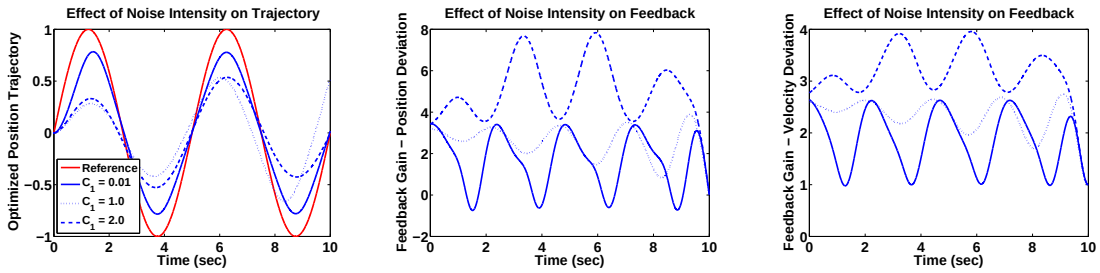
(a) (b) (c)

Figure 3.8: Statistics of 1500 Monte Carlo simulations when the optimal control policy was implemented for input-dependent noise parameterized as $C_2 = 0.01$. The solid line shows the mean response while the dotted lines are drawn 2 computed standard deviations away in either direction.



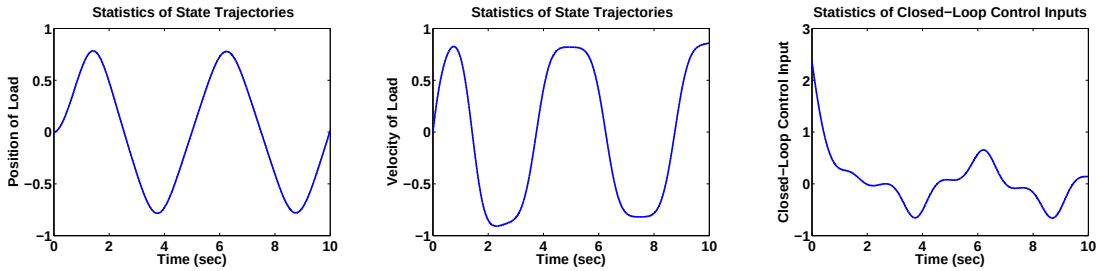
(a) (b) (c)

Figure 3.9: Statistics of 1500 Monte Carlo simulations when the optimal control policy was implemented for input-dependent noise parameterized as $C_2 = 0.5$. The solid line shows the mean response while the dotted lines are drawn 2 computed standard deviations away in either direction.



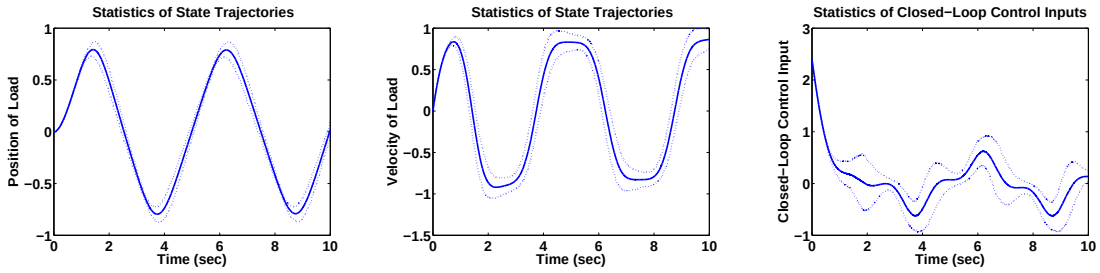
(a) (b) (c)

Figure 3.10: The computed optimized trajectory and feedback gains for different noise parameterizations. Solid, dotted, and dashed lines indicate that the state-dependent noise was parameterized as $C_2 = 0.01$, $C_2 = 1.0$ and $C_2 = 2.0$, respectively.



(a) (b) (c)

Figure 3.11: Statistics of 1500 Monte Carlo simulations when the optimal control policy was implemented for state-dependent noise parameterized as $C_1 = 0.01$. The solid line shows the mean response while the dotted lines are drawn 2 computed standard deviations away in either direction.



(a) (b) (c)

Figure 3.12: Statistics of 1500 Monte Carlo simulations when the optimal control policy was implemented for state-dependent noise parameterized as $C_1 = 0.5$. The solid line shows the mean response while the dotted lines are drawn 2 computed standard deviations away in either direction.

for system model and parameters) is considered. It is shown that the EKF solution is far less dependent on the discretization step size when utilizing the variational integrator. As a result, sensor information can be incorporated less frequently since the filter solution is much more stable. Therefore, the proposed integrator may reduce the computational, power, and sensing requirements of systems by allowing the use of large discretization time steps and reduced observation update rates. Similar results are presented in Reference 34 for a linear filter. All experiments were conducted using MATLAB scripts.

For this study consider dynamics describing a stochastic Hamiltonian system that have a linearization of the following form

$$x_{k+1} = F_k(x_k)x_k + \sum_{i=0}^{p} L_{k,i}(x_k)\Delta\omega_{i_k} \tag{3.29}$$

$$z_k = H_k x_k + v_k \tag{3.30}$$

where $\Delta w_{k,i},\ i \in \{1,\ldots,p\}$ and v_k are independent random variables such that $\Delta w_{k,i} \sim \mathcal{N}(0,\Delta t)$ and $v_k \sim \mathcal{N}(0,R_k)$. Note that the structure of the linearization restricts dynamics of the considered system to be affine in $\Delta\omega_{i_k}$. It is assumed (explicit or implicit) mappings from $(x_k,\omega_{1_k},\ldots,\omega_{p_k}) \to (x_{k+1})$ and from $(x_k,v_k) \to (z_k)$ exists such that the dynamical system can be described in the following form

$$x_{k+1} = f(x_k,\omega_{1_k},\ldots,\omega_{p_k}), \tag{3.31}$$

$$z_k = h(x_k) + v_k \tag{3.32}$$

Note that $f(\cdot,\cdot)$ is implicitly defined by the variational integrator and explicitly defined when the Euler method is utilized and, therefore, both methodologies conform to equation (3.31). Additionally, the first-order linearization given by (3.16) conforms to equation (3.29) since the linearization is affine in $\Delta\omega_k$. The linearization given by (3.3) also conforms to equation (3.29).

Next define $\hat{x}_{k|k-1}$ as the *a priori* state estimate, $\hat{x}_{k|k}$ as the *a posteriori* state estimate, $P_{k|k-1}$ as the *a priori* error covariance matrix, and $P_{k|k}$ as the *a posteriori*

error covariance matrix. The state estimate and error covariance matrix are propagated as described by Algorithm 4. It should be noted that a measurement update is not needed at every time step.

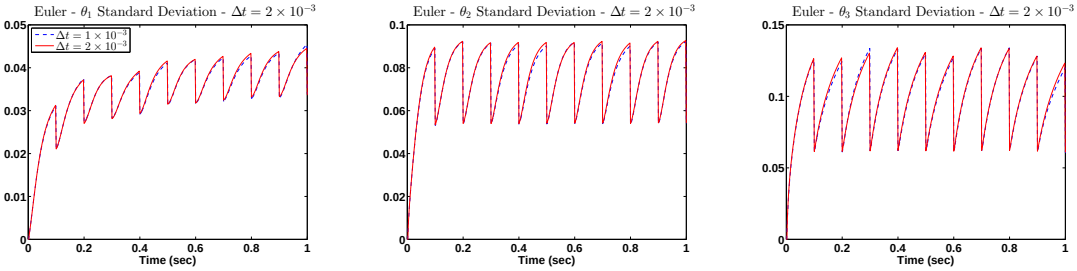
Algorithm 4 Extended Kalman Filter

$\hat{x}_{k|k-1} \leftarrow f(\hat{x}_{k-1|k-1}, 0)$
$P_{k|k-1} \leftarrow F_{k-1} P_{k-1|k-1} F_{k-1}^{\mathrm{T}} + \sum_{i=0}^{p} L_{k-1,i} L_{k-1,i}^{\mathrm{T}} \Delta t$
if measurement available **then**
    $K_k \leftarrow P_{k|k-1} H_k^{\mathrm{T}} / (H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k)$
    $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}))$
    $P_{k|k} \leftarrow (\mathrm{I} - K_k H_k) P_{k|k-1}$
else
    $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1}$
    $P_{k|k} \leftarrow P_{k|k-1}$
end if

The dynamical system considered in the presented numerical experiments is shown in Figure 3.1a and is described by three coordinates given by the relative angles between adjacent links, $\theta(t) = [\theta_1(t), \theta_2(t), \theta_3(t)]$, and three control inputs, $u(t) = [u_1(t), u_2(t), u_3(t)]$. Furthermore, encoders at the link joints give measurements of the relative angles ($h(\theta) = \theta_k$) and the measurement update rate is 10 Hertz (i.e. a new measurement is available every 0.1 seconds). No control inputs are considered for these experiments ($u(t) = 0$) and initial conditions were set to $\theta(t_0) = [\pi, 0, 0]$, $\dot{\theta}(t_0) = [0, 0, 1]$, and $P_{0|0} = 0$. Therefore, the initial state estimate for the EKF utilizing the Euler method was $\hat{x}_{0|0} = [\theta(t_0), \dot{\theta}(t_0)]^{\mathrm{T}}$ and the initial state estimate for the EKF utilizing the variational integrator was $\hat{x}_{0|0} = [\theta(t_0), D_2 L_0(\theta(t_0) - \dot{\theta}(t_0)\Delta t, \theta(t_0))]^{\mathrm{T}}$.

In the first case, the process noise is parameterized as $L_i(q, \dot{q}, u) = C e_i,\ i = \{1, 2, 3\}, C = 5 \times 10^{-4}$ and the observation noise was characterized as $R_k = 5 \times 10^{-3} \mathrm{I}_3$.[6] Figures 3.13, 3.14, 3.15, 3.16, 3.17, 3.18 and 3.19 were generated using the same process and observation noise profiles and the system was propagated using a discretization time step of $\Delta t = 1 \times 10^{-3}$. The Euler method was used to propagate

[6]The standard basis is used to define vectors $e_i, i \in \{1, 2, 3\}$ (e.g. $e_1 = [1, 0, 0]^{\mathrm{T}}$)

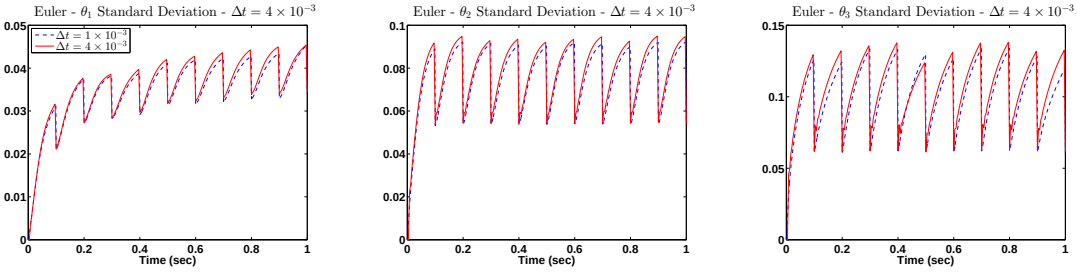


(a) (b) (c)

Figure 3.13: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the Euler method. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 2 \times 10^{-3}$, respectively.

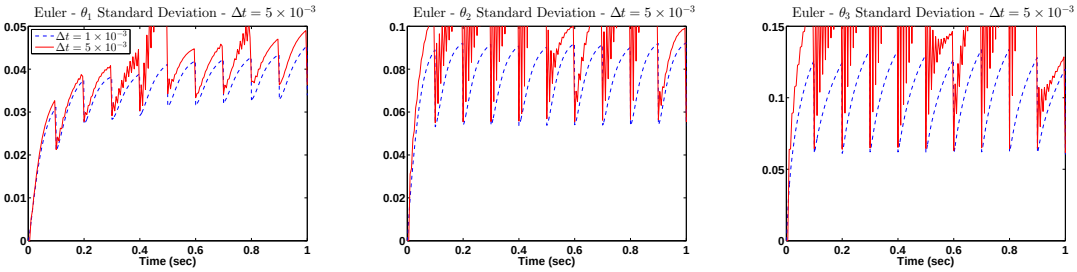
the system when the EKF utilizing the Euler method was tested. Likewise, the variational integrator was used to propagate the system when the EKF utilizing the variational integrator was tested. Figures 3.13, 3.14 and 3.15 show the EKF solution obtained when the Euler method was used for a variety of discretization time steps. Notice that there is a rapid degradation of the solution as the time step is increased and the solution obtained when $\Delta t = 5 \times 10^{-3}$ is erratic. On the other hand, Figures 3.16, 3.17, 3.18 and 3.19 show the EKF solution obtained when the variational integrator was used for a variety of discretization time steps. It is easily noted that a much slower degradation of the solution occurs when compared to Figures 3.13, 3.14 and 3.15. In fact, a significant degradation is only seen when the time step is increase to $\Delta t = 2 \times 10^{-2}$ and even at this large time step the solution is relatively stable and well-behaved. The estimated error variances were validated through a series of 2500 Monte Carlo trials where the process and observation noise profiles were varied. Figure 3.20 shows that the EKF solution utilizing the variational integrator is able to accurately predict the mean squared error when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$.

In the second case, the process noise is dependent on the system configuration and is parameterized as $L_i(q, \dot{q}, u) = Ce_i(\theta_i(t) - \theta_i(t_0))^2,\ i = \{1, 2, 3\},\ C = 1 \times 10^{-3}$ and $L_i(q, \dot{q}, u) = Ce_{i-3},\ i = \{4, 5, 6\},\ C = 5 \times 10^{-4}$. As before, Figures 3.21 and 3.22 were generated using the same process and observation noise profiles and the



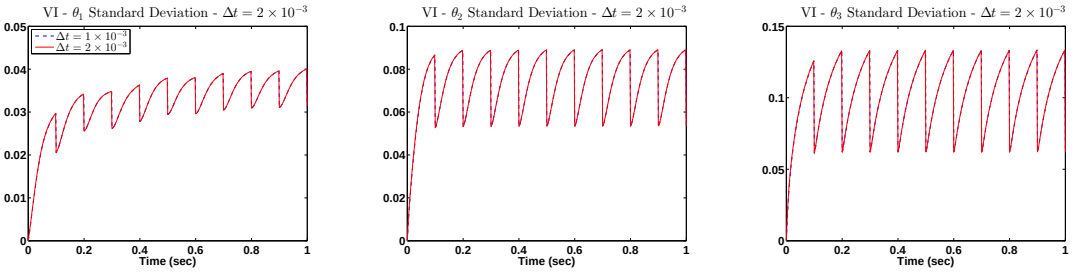
(a) (b) (c)

Figure 3.14: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the Euler method. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 4 \times 10^{-3}$, respectively.



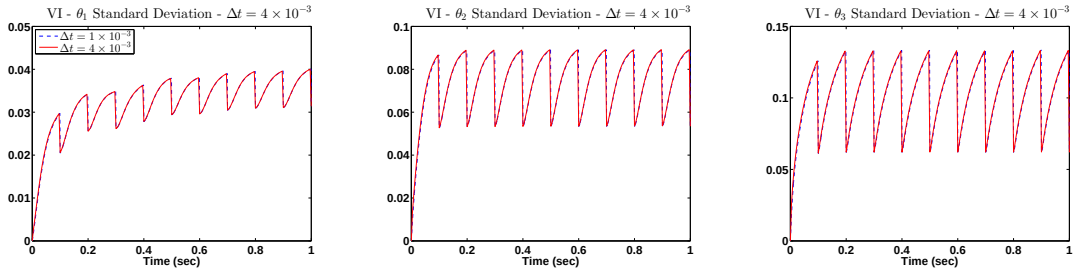
(a) (b) (c)

Figure 3.15: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the Euler method. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.



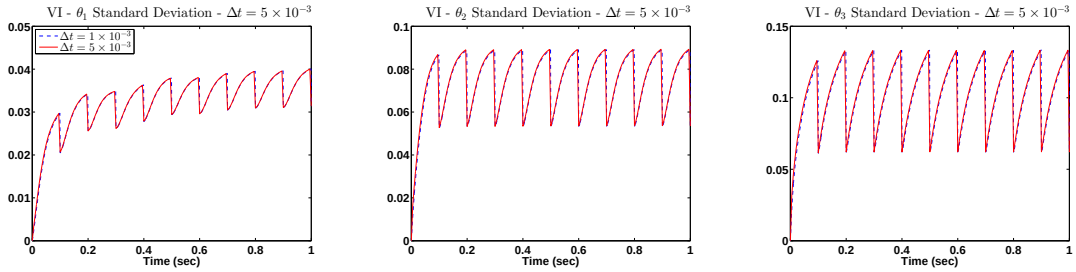
(a) (b) (c)

Figure 3.16: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the proposed variational integrator. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 2 \times 10^{-3}$, respectively.



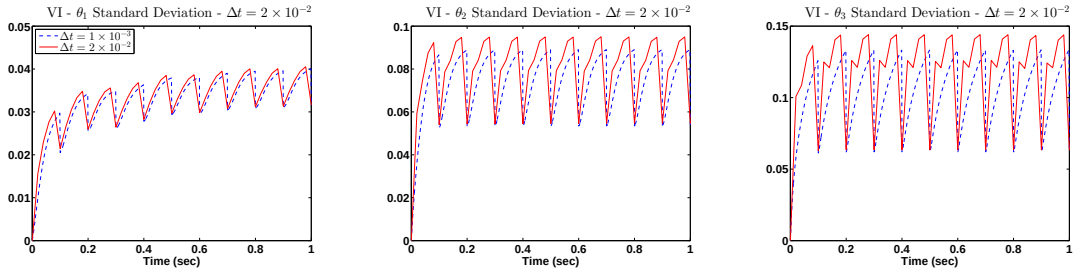
(a) (b) (c)

Figure 3.17: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the proposed variational integrator. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 4 \times 10^{-3}$, respectively.



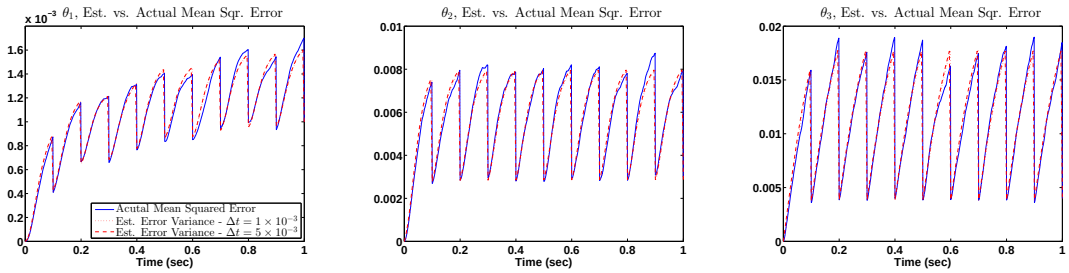
(a) (b) (c)

Figure 3.18: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the proposed variational integrator. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.



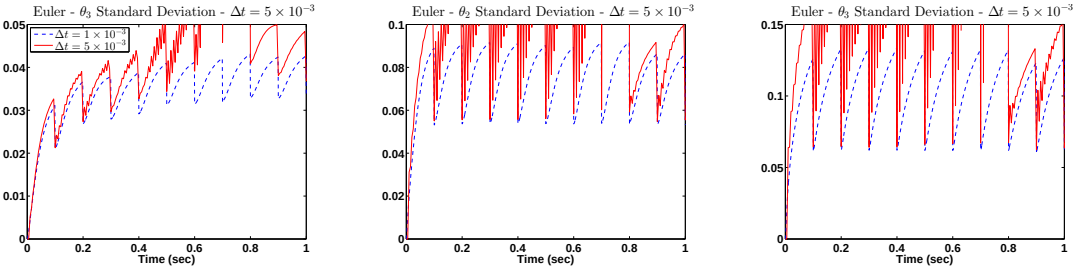
(a) (b) (c)

Figure 3.19: Case 1: The estimated standard deviations of the estimation error calculated by the EKF utilizing the proposed variational integrator. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 2 \times 10^{-2}$, respectively.



(a) (b) (c)

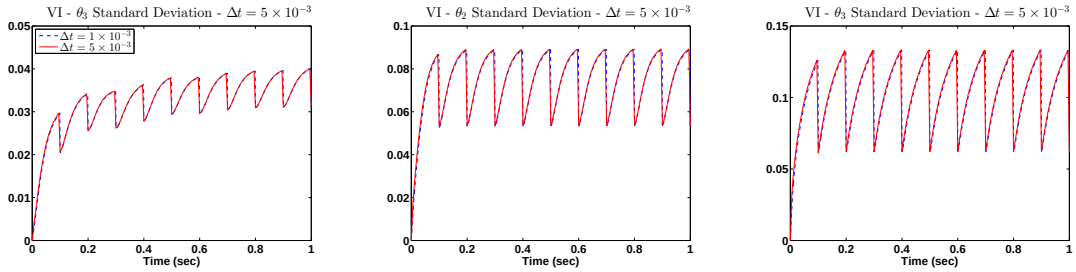
Figure 3.20: Case 1: The estimated variance of the estimation error calculated by the EKF utilizing the proposed variational integrator and the calculated mean squared error over 2500 Monte Carlo trials. The solid line corresponds to the the calculated mean squared error and the dotted and dashed lines correspond to the estimate obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.



(a) (b) (c)

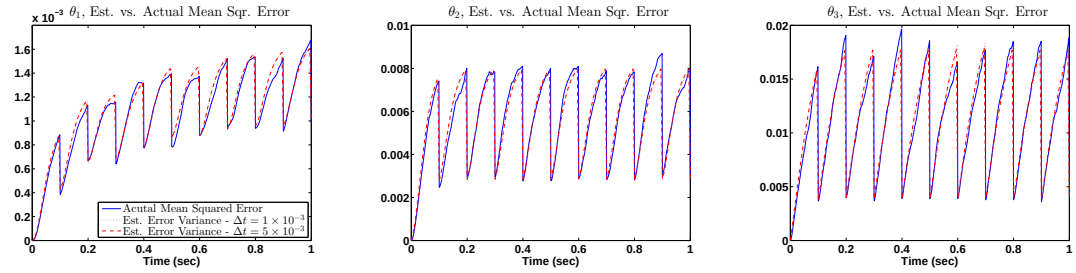
Figure 3.21: Case 2: The estimated standard deviations of the estimation error calculated by the EKF utilizing the Euler method. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.

system was propagated using a discretization time step of $\Delta t = 1 \times 10^{-3}$. Figure 3.21 and 3.22 show the EKF solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$. As in the first case, when the Euler method is used the solution becomes unstable when $\Delta t = 5 \times 10^{-3}$. On the other hand, the EKF solutions utilizing the variational integrator are virtually identical. Therefore, it can be concluded that increasing the discretization time step affects the EKF algorithm much more if the Euler method is used. The estimated error variances were validated by a series of 2500 Monte Carlo trials where the process and observation noise profiles were varied. Figure 3.23 shows that the EKF solution utilizing the variational integrator was able to accurately predict the mean squared error when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$.



(a) (b) (c)

Figure 3.22: Case 2: The estimated standard deviations of the estimation error calculated by the EKF utilizing the proposed variational integrator. The dotted and solid lines correspond to the solutions obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.



(a) (b) (c)

Figure 3.23: Case 2: The estimated variance of the estimation error calculated by the EKF utilizing the proposed variational integrator and the calculated mean squared error over 2500 Monte Carlo trials. The solid line corresponds to the the calculated mean squared error and the dotted and dashed lines correspond to the estimate obtained when $\Delta t = 1 \times 10^{-3}$ and $\Delta t = 5 \times 10^{-3}$, respectively.

3.4 Conclusion

In this chapter a stochastic variational integrator and its linearization were presented. The stochastic differential dynamical programming algorithm was used to compare the performance of the variational integrator to that of the standard Euler method. It was demonstrated that the S-DDP algorithm was far less dependent on the discretization time step when the variational integrator was used to propagate system trajectories and linearize system dynamics. Furthermore, the extended Kalman filter algorithm was also used to compare the two methodologies. Similar benefits were seen in this case. Specifically, the solution obtained was much less affected by changes in the discretization time step when the variational integrator was used to propagate system trajectories and linearize system dynamics. Furthermore, the filter was more well-behaved when the variational integrator was implemented. It is stressed that these benefits are not limited to the S-DDP and EKF algorithms and similar improvements can be expected to any process that utilizes propagated system trajectories or linearized system dynamics. Finally from a implementation standpoint, the use of variational integrators may enable real-time implementation of nonlinear optimal control algorithms and reduce the computational, power, and sensing requirements of control and estimation technologies.

IV

A POLYNOMIAL CHAOS VARIATIONAL INTEGRATOR

> *"We look upon a thing as the effect of chance when we see nothing regular in it, nothing that manifests design, and when furthermore, we are ignorant of the causes that brought it about. Thus, chance has no reality in itself. It is nothing but a term for expressing our ignorance of the way in which the various aspects of a phenomenon are interconnected and related to the rest of nature."*
>
> – Pierre Simon de Laplace, *Essai philosophique sur les probabilitès*

In this chapter a variational integrator for the propagation of polynomial chaos expansion coefficients describing a Hamiltonian system is formulated. Using the previously reported result showing that expansion coefficients form a Hamiltonian system a variational integrator is derived for the resulting expansion coefficient system. A simple numerical example is presented to demonstrate the benefits of using a variational integrator in this unique setting. Specifically, the presented integration method retains its accuracy over a large range of discretization time step sizes. As a result, the computational time required for integration should be able to be reduced. Therefore, the presented variational integrator may be a viable when the number of expansion coefficients is large or when the uncertain system is highly nonlinear.

4.1 Formulation

The description of a Hamiltonian system in the context of polynomial chaos expansion and the proof that expansion coefficients form a Hamiltonian system was first introduce in Reference 97. Specifically, consider a system described by an Hamiltonian $H(q,p;\lambda) = T(q,p;\lambda) + V(q;\lambda)$, where $T(q,p;\lambda)$ is the total kinetic energy of the system, $V(q;\lambda)$ is the potential energy of the system, and the parameter λ is a vector representing uncertain parameters with a known probability density $\rho(\lambda)$. The

propagation of the uncertain system can be derived using the classical Hamiltonian equations:

$$\dot{q}_i = \frac{\partial H(q,p;\lambda)}{\partial p_i} \tag{4.1}$$

$$\dot{p}_i = -\frac{\partial H(q,p;\lambda)}{\partial q_i}, \tag{4.2}$$

where the generalized polynomial chaos (gPC) expansions of the generalized coordinates, q_i, and momentum, p_i, are given as

$$q_i = \sum_{k=0}^{\infty} Q_{ik}\phi_k(\lambda), \tag{4.3}$$

$$p_i = \sum_{k=0}^{\infty} P_{ik}\phi_k(\lambda), \tag{4.4}$$

where functions $\phi_k(\lambda)$ form a set of orthogonal polynomials (see Section 2.4 for further details). The propagation of the expansion coefficients can be described using equations (4.1), (4.2), (4.3), and (4.13):

$$\dot{Q}_{ik} = \int_{\Gamma} \dot{q}_i\phi_k(\lambda)\rho(\lambda)d\lambda = \int_{\Gamma} \frac{\partial H}{\partial p_i}\phi_k(\lambda)\rho(\lambda)\ \mathrm{d}\lambda \tag{4.5}$$

$$\dot{P}_{ik} = \int_{\Gamma} \dot{p}_i\phi_k(\lambda)\rho(\lambda)d\lambda = -\int_{\Gamma} \frac{\partial H}{\partial q_i}\phi_k(\lambda)\rho(\lambda)\ \mathrm{d}\lambda. \tag{4.6}$$

Furthermore, define the expected value of the Hamiltonian as

$$\hat{H}(Q,P) = \int_{\Gamma} H(q,p;\lambda)\rho(\lambda)\ \mathrm{d}\lambda \tag{4.7}$$

where Q and P are the gPC expansion coefficients.

Theorem 1 [Theorem 3.1 [97]] The gPC expansion coefficients $\{Q,P\}$ and the expected value of the Hamiltonian, $\hat{H}(Q,P)$, form a Hamiltonian system where the expansion coefficient propagation as follows:

$$\dot{Q}_{ik} = \frac{\partial \hat{H}}{\partial P_{ik}} \tag{4.8}$$

$$\dot{P}_{ik} = -\frac{\partial \hat{H}}{\partial Q_{ik}}. \tag{4.9}$$

Proof The proof follows directly from computing partial derivatives of the expected value of the Hamiltonian:

$$\begin{aligned}
\frac{\partial \hat{H}}{\partial P_{ik}} &= \int_{\Gamma} \Big(\sum_{s=1}^{n} \frac{\partial H}{\partial q_s} \frac{q_s}{\partial P_{ik}} + \frac{\partial H}{\partial p_s} \frac{p_s}{\partial P_{ik}} \rho(\lambda) \Big) \, \mathrm{d}\lambda \\
&= \int_{\Gamma} \sum_{s=1}^{n} \frac{\partial H}{\partial p_s} \frac{p_s}{\partial P_{ik}} \rho(\lambda) \, \mathrm{d}\lambda \\
&= \int_{\Gamma} \sum_{s=1}^{n} \sum_{r=1}^{\infty} \frac{\partial H}{\partial p_s} \delta_{is} \delta_{kr} \phi_r(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \\
&= \int_{\Gamma} \frac{\partial H}{\partial p_i} \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \\
&= \dot{Q}_{ik}, \qquad (4.10)
\end{aligned}$$

and, similarly,

$$\begin{aligned}
\frac{\partial \hat{H}}{\partial Q_{ik}} &= - \int_{\Gamma} \Big(\sum_{s=1}^{n} \frac{\partial H}{\partial q_s} \frac{q_s}{\partial Q_{ik}} + \frac{\partial H}{\partial p_s} \frac{p_s}{\partial Q_{ik}} \rho(\lambda) \Big) \, \mathrm{d}\lambda \\
&= - \int_{\Gamma} \sum_{s=1}^{n} \frac{\partial H}{\partial q_s} \frac{q_s}{\partial Q_{ik}} \rho(\lambda) \, \mathrm{d}\lambda \\
&= - \int_{\Gamma} \sum_{s=1}^{n} \sum_{r=1}^{\infty} \frac{\partial H}{\partial q_s} \delta_{is} \delta_{kr} \phi_r(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \\
&= - \int_{\Gamma} \frac{\partial H}{\partial q_i} \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \\
&= - \dot{P}_{ik} \qquad (4.11)
\end{aligned}$$

■

Theorem 1 highlights a fundamental property of Hamiltonian systems. The symplectic structure of Hamiltonian equations (4.1) and (4.2) is preserved when the expansion coefficients are propagated. It is easily seen that any representation of the generalized coordinates and momentum that is composed with orthogonal basis functions will inherit a symplectic structure. In fact, the standard representations $q = [q_1, \ldots, q_n]$ and $p = [p_1, \ldots, p_n]$ are composed with orthogonal basis functions (the standard Cartesian basis) and, therefore, equations (4.1) and (4.2) are akin to (4.8) and (4.9). Therefore, the symplectic structure, needed for the formulation of

variational integrators, is preserved regardless of the complexity of coordinate representation, so long as orthogonality is present.

As noted in Reference 97, the coefficients of a truncated expansion (see equation (2.109)) will also form a Hamiltonian system relative to the quasi-expected value of the Hamiltonian regardless of the degree of expansion. As a result, the non-physical expansion coefficients behave like physical system configuration states. Therefore, coefficients from truncated expansions can be numerically propagated forward in time using the same methods used to propagate Hamiltonian systems. Under this train of thought, consider the resulting truncated expansion coefficient Hamiltonian,

$$H(Q,P;\lambda) = T(\tilde{q},\tilde{p};\lambda) + V(\tilde{p};\lambda) = T(Q,P;\lambda) + V(P;\lambda) \tag{4.12}$$

where the generalized system configuration and momentum are approximated as

$$q_i \approx \tilde{q}_i = \sum_{k=0}^{r} Q_{ik}\phi_k(\lambda), \tag{4.13}$$

$$p_i \approx \tilde{p}_i = \sum_{k=0}^{r} P_{ik}\phi_k(\lambda). \tag{4.14}$$

In general, the Hamiltonian in equation (4.12) is an aprroximation of $H(q,p;\lambda)$. However, as the degree of the approximation increases, $r \to \infty$, the resulting expansion coefficient Hamiltonian converges to the uncertain system's Hamiltonian, $H(Q,P;\lambda) \to H(q,p;\lambda)$. Similarly, the resulting truncated expansion coefficient Lagrangian is formulated as

$$L(Q,\dot{Q};\lambda) = T(\tilde{q},\dot{\tilde{q}};\lambda) - V(\tilde{q};\lambda) = T(Q,\dot{Q};\lambda) - V(Q;\lambda) \tag{4.15}$$

where the gPC expansion of the system configuration and its time derivative are given as

$$q_i \approx \tilde{q}_i = \sum_{k=0}^{r} Q_{ik}\phi_k(\lambda), \tag{4.16}$$

$$\dot{q}_i \approx \dot{\tilde{q}}_i = \sum_{k=0}^{r} \dot{Q}_{ik}\phi_k(\lambda). \tag{4.17}$$

In the proceeding presentation Q is viewed as an extended system configuration state and $\dot{Q}$ as its derivative. The forced Euler-Lagrange equation can now be used to obtain an ordinary differential equation describing the evolution of the expansion coefficients:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{Q}}(Q,\dot{Q}) - \frac{\partial L}{\partial Q}(Q,\dot{Q}) = F(Q(\tau),\dot{Q}(\tau),u(\tau)). \tag{4.18}$$

where

$$F_i(Q(t),\dot{Q}(t),u(t);\lambda) = \sum_{k=0}^{r} F_{ik}(Q(t),\dot{Q}(t),u(t);\lambda)\phi_k(\lambda). \tag{4.19}$$

Note that equation (4.18) gives a second order ordinary differential equation, but, as discussed in Chapter 2, does not provide any method or algorithm that can be used to solve for the propagation of the expansion coefficients.

As in Chapters 2 and 3, a discrete approximation of the Lagrangian in conjunction with the Hamiltonian variational principle are used in order to obtain a variational integrator. The forced discrete Euler-Lagrange (DEL) equation for the considered polynomial chaos expansion system is derived to be

$$P_k + D_1 L_{\mathrm{d}}(Q_k, Q_{k+1};\lambda) + F_{\mathrm{d}}^{-}(Q_k, Q_{k+1}, u_k;\lambda) = 0 \tag{4.20}$$

$$P_{k+1} = D_2 L_{\mathrm{d}}(Q_k, Q_{k+1};\lambda) + F_{\mathrm{d}}^{+}(Q_k, Q_{k+1}, u_k;\lambda) \tag{4.21}$$

where the sequence of expansion coefficient vectors $\{(t_0,Q_0),(t_1,Q_1),\ldots,(t_n,Q_n)\}$ approximate the continuous expansion coefficient evolution such that $Q_m \approx Q(t_m)$ where $\Delta t = t_{i+1} - t_i$ is the discretization time step. The discrete Lagrangian $L_{\mathrm{d}}(Q_k, Q_{k+1};\lambda)$ approximates the action integral over a small time interval such that

$$L_{\mathrm{d}}(Q_k, Q_{k+1};\lambda) = L((1-\alpha)Q_k + \alpha Q_{k+1}, \frac{Q_{k+1}-Q_k}{\Delta t};\lambda)\Delta t. \tag{4.22}$$

where $\alpha \in [0,1]$ parameterizes a generalized midpoint approximation and $\alpha = 1/2$ results in second order accuracy as discussed in Reference 115. $F_{\mathrm{d}}^{-}(Q_k, Q_{k+1}, u_k;\lambda)$

and $F_{\mathrm{d}}^{+}(Q_k, Q_{k+1}, u_k; \lambda)$ are the left and right discrete forces, respectively, and approximate continuous external forces over a small time interval:

$$F_{\mathrm{d}}^{-}(Q(t), \dot{Q}(t), u(t); \lambda) \cdot \delta Q_k + F_{\mathrm{d}}^{+}(Q(t), \dot{Q}(t), u(t); \lambda) \cdot \delta Q_{k+1} \approx \int_{t_k}^{t_{k+1}} F(Q(t), \dot{Q}(t), u(t); \lambda) \cdot \delta Q \, \mathrm{d}\tau. \tag{4.23}$$

When integrating a deterministic (Chapter 2) or a stochastic (Chapter 3) unforced Hamiltonian system q_k can be viewed as the momentum quantity conserved by the integrator. However, in this context P_k does not have an equivalent physical representation and can be viewed as a quantity that facilitates integration. The derivative of the integrator equation is defined as

$$Df(Q_{k+1}) = D_2 D_1 L_{\mathrm{d}}(Q_k, Q_{k+1}) + D_2 F_{\mathrm{d}}^{-}(Q_k, Q_{k+1}, u_k; \lambda) \tag{4.24}$$

Expressions $D_1 L_{\mathrm{d}}(Q_k, Q_{k+1}; \lambda)$, $D_2 L_{\mathrm{d}}(Q_k, Q_{k+1}; \lambda)$, and $D_2 D_1 L_{\mathrm{d}}(Q_k, Q_{k+1}; \lambda)$ can be derived similarly to the expressions given in equations (2.69), (2.70), and (2.73). Furthermore, the partial derivatives needed to evaluate these derived expressions are obtained using the same methodology presented in the proof of Theorem 4.1:

$$\frac{\partial L(Q, \dot{Q}; \lambda)}{\partial Q_{ik}} = \int_{\Gamma} \frac{\partial L(Q, \dot{Q}; \lambda)}{\partial \tilde{q}_i} \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \tag{4.25}$$

$$\frac{\partial L(Q, \dot{Q}; \lambda)}{\partial \dot{Q}_{ik}} = \int_{\Gamma} \frac{\partial L(Q, \dot{Q}; \lambda)}{\partial \dot{\tilde{q}}_i} \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \tag{4.26}$$

$$\frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial Q_{jm} \partial Q_{ik}} = \int_{\Gamma} \frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial \tilde{q}_j \partial \tilde{q}_i} \phi_m(\lambda) \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \tag{4.27}$$

$$\frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial \dot{Q}_{jm} \partial \dot{Q}_{ik}} = \int_{\Gamma} \frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial \dot{\tilde{q}}_j \partial \dot{\tilde{q}}_i} \phi_m(\lambda) \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \tag{4.28}$$

$$\frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial Q_{jm} \partial \dot{Q}_{ik}} = \int_{\Gamma} \frac{\partial^2 L(Q, \dot{Q}; \lambda)}{\partial \tilde{q}_j \partial \dot{\tilde{q}}_i} \phi_m(\lambda) \phi_k(\lambda) \rho(\lambda) \, \mathrm{d}\lambda \tag{4.29}$$

As in Chapters 2 and 3, a first-order linearization of the discrete dynamics can be found:

$$\begin{bmatrix} \delta Q_{k+1} \\ \delta P_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial Q_{k+1}}{\partial P_k} & \frac{\partial P_{k+1}}{\partial p_k} \\ \frac{\partial P_{k+1}}{\partial Q_k} & \frac{\partial P_{k+1}}{\partial P_k} \end{bmatrix} \begin{bmatrix} \delta Q_k \\ \delta P_k \end{bmatrix} + \begin{bmatrix} \frac{\partial Q_{k+1}}{\partial u_k} \\ \frac{\partial P_{k+1}}{\partial u_k} \end{bmatrix} \delta u_k,$$

The expressions for the required derivatives can be found using the same methodology used to obtain equations (2.98)-(2.105).

4.2 Simple Application

In order to further elucidate the presented variational integrator, consider a simple linear mass-spring system with unity mass and an uncertain spring stiffness constant represented as $(5 + \sigma\lambda)\ N/m$ where $\lambda \sim \mathcal{N}(0, 1)$ and σ is a measure of uncertainty (σ^2 gives the variance of the spring stiffness). If the expansion is selected such that $r = 3$, the resulting truncated expansion coefficient Lagrangian is given as

$$\begin{aligned} L(Q, \dot{Q}; \lambda) = &\frac{1}{2} m(\dot{Q}_{10}\phi_0 + \dot{Q}_{11}\phi_1 + \dot{Q}_{12}\phi_2 + \dot{Q}_{13}\phi_3)^2 \\ &- \frac{1}{2}(K + \sigma\lambda)(Q_{10}\phi_0 + Q_{11}\phi_1 + Q_{12}\phi_2 + Q_{13}\phi_3)^2 \end{aligned} \tag{4.30}$$

The variational integrator can now be derived using equation (2.69), (2.70), (2.73), and (4.20). Recall partial derivatives of polynomial chaos expansions are calculated as in equations (4.25)-(4.29) As an illustrative example the partial $\frac{\partial L}{\partial \dot{Q}_{13}}$ is calculated as

$$\begin{aligned} \frac{\partial L}{\partial Q_{13}} &= \int_\Gamma \frac{\partial L(Q, \dot{Q}; \lambda)}{\partial \tilde{q}_1} \phi_3(\lambda)\rho(\lambda)\ \mathrm{d}\lambda, \\ &= -\int_\Gamma (K + \sigma\lambda)(Q_{10} + Q_{11}\lambda + Q_{12}(\lambda^2 - 1) + Q_{13}(\lambda^3 - 3\lambda))(\lambda^3 - 3\lambda)\rho(\lambda)\ \mathrm{d}\lambda, \\ &= -6KQ_{13} - 6\sigma Q_{12}, \end{aligned} \tag{4.31}$$

since[1]

$$\int_\Gamma \big(KQ_{10}\lambda^3 - 3KQ_{10}\lambda + \sigma Q_{10}\lambda^4 - 3\sigma Q_{10}\lambda^2\big)\rho(\lambda)\ \mathrm{d}\lambda = 0,$$
$$\int_\Gamma \big(KQ_{11}\lambda^4 - 3KQ_{11}\lambda^2 + \sigma Q_{11}\lambda^5 - 3\sigma Q_{11}\lambda^3\big)\rho(\lambda)\ \mathrm{d}\lambda = 0,$$
$$\int_\Gamma \big(\sigma Q_{12}(\lambda^6 - 4\lambda^4 + 3\lambda^2) + KQ_{13}(\lambda^5 - 4\lambda^3 + 3\lambda)\big)\rho(\lambda)\ \mathrm{d}\lambda = 6\sigma Q_{12},$$
$$\int_\Gamma \big(\sigma Q_{13}(\lambda^7 - 6\lambda^5 + 9\lambda^3) + KQ_{13}(\lambda^6 - 6\lambda^4 + 9\lambda^2)\big)\rho(\lambda)\ \mathrm{d}\lambda = 6KQ_{13}.$$

The remaining derivatives required to derive the variational integrator and its linearization are given by

$$\frac{\partial L}{\partial Q} = \begin{bmatrix} -KQ_{10} - \sigma Q_{11} \\ -KQ_{11} - \sigma Q_{10} - 2\sigma Q_{12} \\ -2KQ_{12} - 2\sigma Q_{11} - 6\sigma Q_{13} \\ -6KQ_{13} - 6\sigma Q_{12} \end{bmatrix}, \quad \frac{\partial L}{\partial \dot{Q}} = \begin{bmatrix} m\dot{Q}_{10} \\ m\dot{Q}_{11} \\ 2m\dot{Q}_{12} \\ 6m\dot{Q}_{13} \end{bmatrix}, \tag{4.32}$$

$$\frac{\partial^2 L}{\partial Q \partial Q} = \begin{bmatrix} -K & -\sigma & 0 & 0 \\ -\sigma & -K & -2\sigma & 0 \\ 0 & -2\sigma & -2K & -6\sigma \\ 0 & 0 & -6\sigma & -6K \end{bmatrix}, \quad \frac{\partial^2 L}{\partial \dot{Q} \partial \dot{Q}} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & 2m & 0 \\ 0 & 0 & 0 & 6m \end{bmatrix},$$

$\frac{\partial^2 L}{\partial \dot{Q} \partial Q} = 0$, and $\frac{\partial^2 L}{\partial Q \partial \dot{Q}} = 0$.

The forced Euler-Lagrange equation provides the differential equations needed to propagate the expansion coefficients using the Euler method (see equation [3.2]):

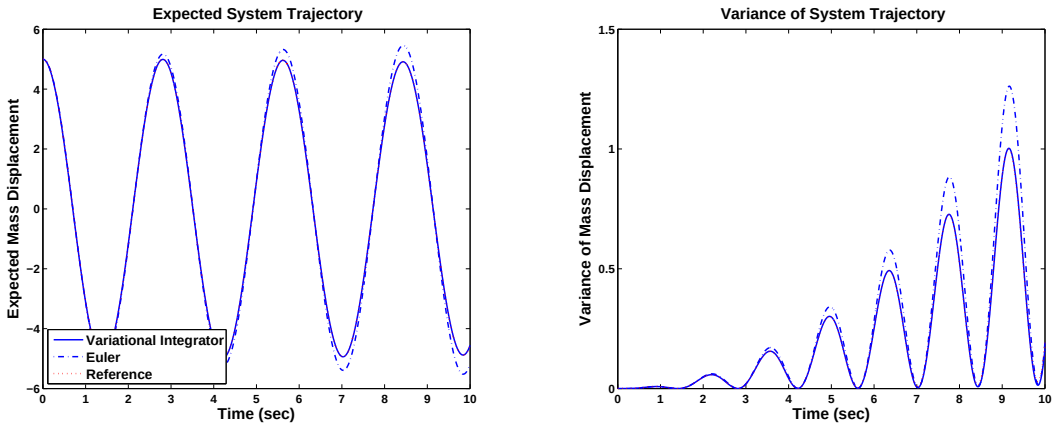
$$\ddot{Q}_{10} = -(KQ_{10} + \sigma Q_{11})/m,$$
$$\ddot{Q}_{11} = -(KQ_{11} + \sigma Q_{10} + 2\sigma Q_{12})/m,$$
$$\ddot{Q}_{12} = -(2KQ_{12} + 2\sigma Q_{11} + 6\sigma Q_{13})/2m,$$
$$\ddot{Q}_{13} = -(6KQ_{13} + 6\sigma Q_{12})/6m.$$

[1]Recall that if $\lambda \sim \mathcal{N}(0,1)$ then $\int_\Gamma \lambda^p \rho(\lambda)\ \mathrm{d}\lambda = \begin{cases} 0 & \text{if } p \text{ is odd} \\ (p-1)!! & \text{if } p \text{ is even} \end{cases}$.

In order to obtain reference trajectories 1500 Monte Carlo simulations where conducted such that each trial was deterministic and the spring stiffness for each trial was sampled from the considered distribution. The mean trajectory and the squared mean error from the mean trajectory are considered the *reference* expected trajectory and the *reference* variance. Figure 4.1 shows the computed expansion coefficients using the Euler method and the presented variational integrator when $\sigma = 0.1$, $\Delta t = 5 \times 10^{-3}$, and $r = 3$. Note that the Euler method results in a trajectory that diverges quicker from the reference trajectory. Furthermore, the variational integrator was able to fairly accurately propagate both the expected mass displacement and its variance.

Figure 4.2 shows the computed expansion coefficients using the presented variational integrator when $\sigma = 0.1$ and $\Delta t = 5 \times 10^{-3}$ over a range of expansion degrees. As expected, reducing the expansion degree deteriorates the accuracy of the propagated trajectories when compared to the reference trajectory. However, even with a relatively small expansion degree accurate trajectories were obtained. Nevertheless, Figure 4.3 shows the effect on the propagated trajectories when σ is increase to 0.25. Notice the accuracy of all the trajectories are deteriorated and there was a much larger effect on the expansion with the smallest degree. Therefore, the amount of uncertainty strongly affects the accuracy of truncated expansions. Furthermore, as shown in Figure 4.4, the accuracy degrades as the trajectory moves further away from the initial time. The error induced by using the truncated expansion accumulates and, as a result, the accuracy of the trajectory continuously degrades. Note that increasing the expansion degree does mitigate this degradation.

Figure 4.5 shows the computed expansion coefficients using the presented variational integrator when $\sigma = 0.1$ and $r = 3$ over a range of discretization time steps. Notice that there is a negligible difference between trajectories obtained with different discretization time steps. As in Chapter 3, the variational integrator is able to maintain its performance despite a change in Δt. This property distinguishes it from



(a) (b)

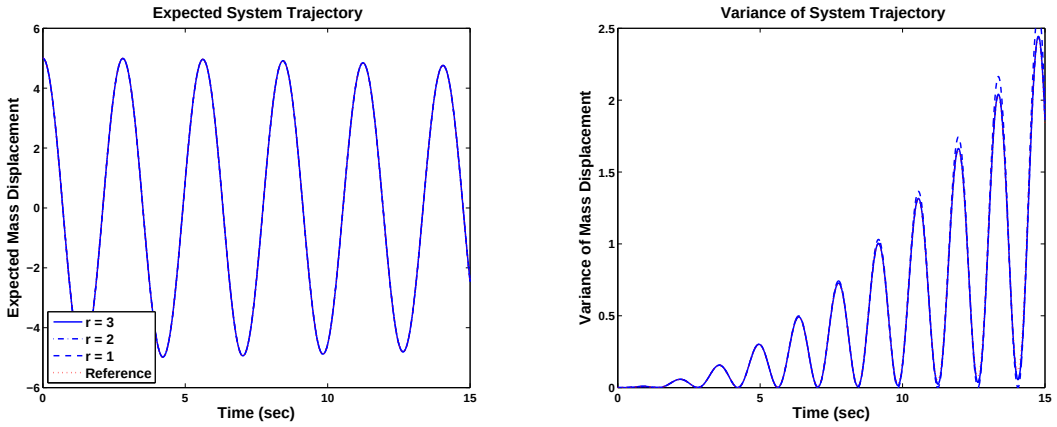
Figure 4.1: (a) and (b): Propagation of the expansion coefficients with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ subject to $\sigma = 0.1$ and $\Delta = 5\times10^{-3}$. Dotted-dashed and solid lines indicate the trajectories were propagated using the Euler method and the variational integrator, respectively, and the dotted line represents the statistics of 1500 Monte Carlo trials.

other integration schemes. As a result, less computational resources should be needed in order to propagate expansion coefficients forward in time.

Figure 4.6 shows the average error in the propagated expansion coefficients when compared to the reference trajectory. It can be concluded that increasing the discretization time and the amount of uncertainty increases the average error in the propagated expansion coefficients. Note that the errors in the expected system configuration all follow a similar trend while the errors in the variance are relatively constant.

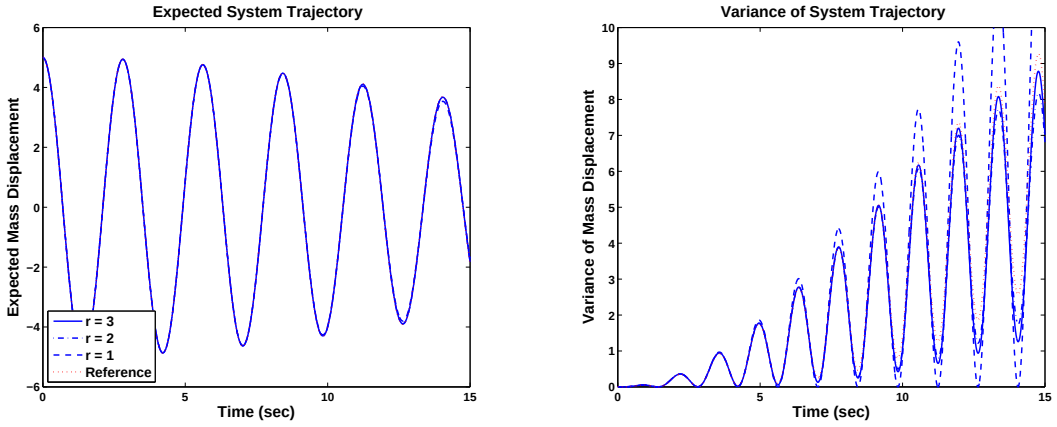
4.3 Conclusion

A variational integrator for polynomial chaos expansion coefficients was presented. It was shown that the presented integrator was able to accurately predict the expectation and variance of an uncertain system. The size of the expansion coefficient vector is a limiting factor of the presented approach. However, it has been shown that variational integrators for an arbitrary mechanical system can be implemented and scales for large state configuration vectors [58]. Therefore, the complexity of the presented approach



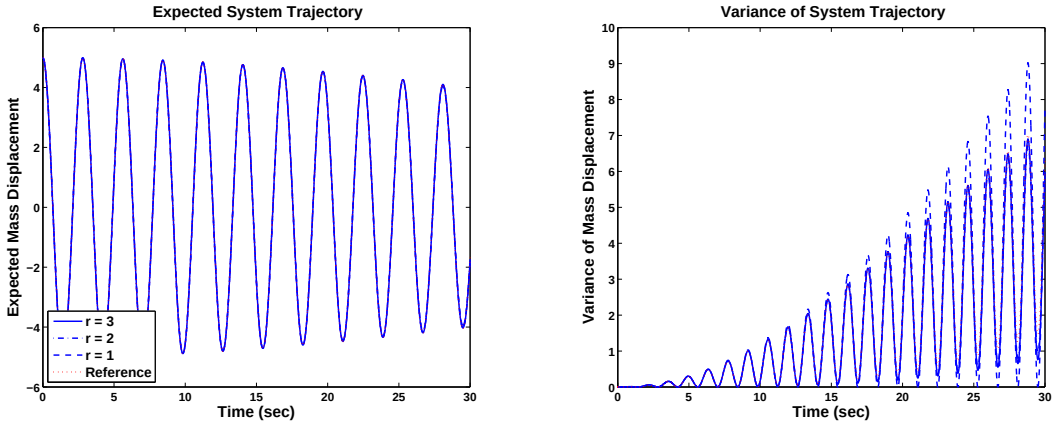
(a) (b)

Figure 4.2: (a) and (b): Propagation of the expansion coefficients using the presented variational integrator, with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ subject to $\sigma = 0.1$ and $\Delta = 5 \times 10^{-3}$. Solid, dotted-dashed, and dotted lines indicate expansion degrees of $r = 1$, $r = 2$ and $r = 3$ were considered, respectively, and the dotted line represents the statistics of 1500 Monte Carlo trials.



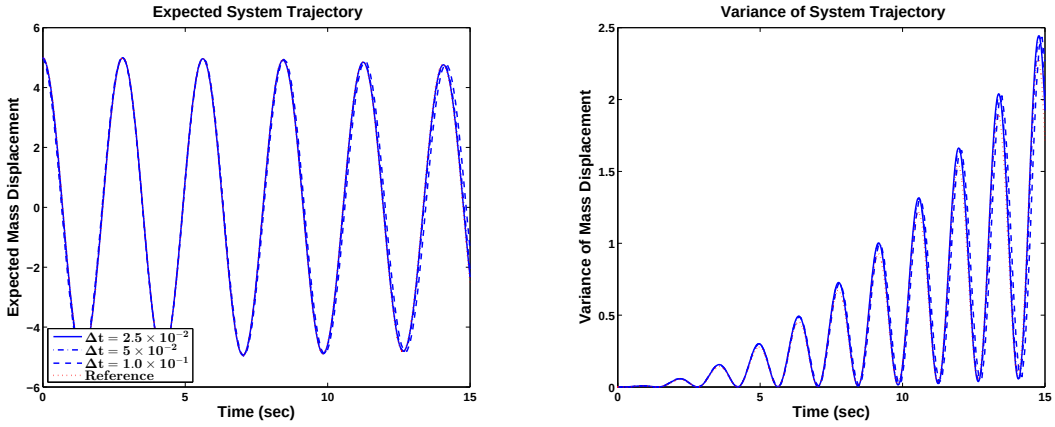
(a) (b)

Figure 4.3: (a) and (b): Propagation of the expansion coefficients using the presented variational integrator, with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ subject to $\sigma = 0.25$ and $\Delta = 5 \times 10^{-3}$. Solid, dotted-dashed, and dotted lines indicate expansion degrees of $r = 1$, $r = 2$, and $r = 3$, were considered, respectively, and the dotted line represents the statistics of 1500 Monte Carlo trials.



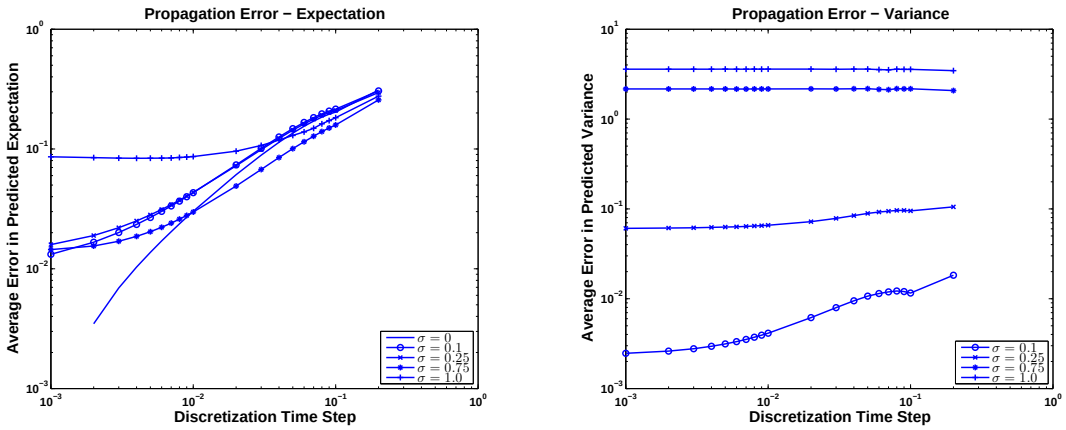
(a) (b)

Figure 4.4: (a) and (b): Propagation of the expansion coefficients using the presented variational integrator, with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ subject to $\sigma = 0.25$ and $\Delta = 5 \times 10^{-3}$. Solid, dotted-dashed, and dotted lines indicate expansion degrees of $r = 1$, $r = 2$, and $r = 3$, were considered, respectively, and the dotted line represents the statistics of 1500 Monte Carlo trials.



(a) (b)

Figure 4.5: (a) and (b): Propagation of the expansion coefficients using the presented variational integrator, with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ subject to $\sigma = 0.25$ and $r = 3$. Solid, dotted-dashed, and dotted lines indicate a discretization time step of $\Delta t = 2.5 \times 10^{-2}$, $\Delta t = 5 \times 10^{-2}$, and $\Delta t = 1 \times 10^{-1}$, were considered, respectively, and the dotted line represents the statistics of 1500 Monte Carlo trials.



(a) (b)

Figure 4.6: (a) and (b): Average error in the propagated expansion coefficients when compared to statistics of 1500 Monte Carlo trials using the presented variational integrator, with initial condition $Q(t_0) = [5, 0, 0, 0, 0]$ and $\dot{Q}(t_0) = [0, 0, 0, 0, 0]$ over a range of discretization times and parameter uncertainty. Circle, X, star, and plus markers indicate that the uncertainty was parameterized as $\sigma = 0.1$, $\sigma = 0.25$, $\sigma = 0.75$, and $\sigma = 1.0$, respectively, and the solid line indicates that no uncertainty was considered.

may be mitigated through software.

V

A TRAJECTORY OPTIMIZATION METHOD FOR AUTONOMOUS SUSPENDED LOAD OPERATIONS

"You promised me Mars colonies. Instead, I got Facebook."
– Dr. Buzz Aldrin

In this chapter a real-time implementable trajectory optimization framework for autonomous suspended load operations is presented. The computational effort required for the optimization frameworks is mitigated by the use of variational integrators and a simplified, but representative, system model. It should be expected that the manner in which an algorithm propagates the modeled system's configuration and linearizes the resultant discrete dynamics has a large effect on its performance. As shown in Chapter 3, utilizing a variational integrator within the DDP algorithm provides significant advantages when compared to utilizing Euler methods. Specifically, the algorithm's performance is far less dependent on the size of the discretization time step. Therefore, the discretization time step can be made larger without sacrificing much performance and enables real-time implementation through the reduction of computational effort. The unmanned vehicle system's existing guidance, navigation, and control architecture is utilized resulting in minimal software reconfiguration. Furthermore, the state of the slung load is estimated via an augmentation to the existing navigation system and utilizes only vision-based measurements of the load. It is shown in simulation studies and a flight test the framework is real-time onboard implementable despite a relatively large time horizon of 12 seconds. The vehicle was able to maneuver the suspended load to track reference trajectories.

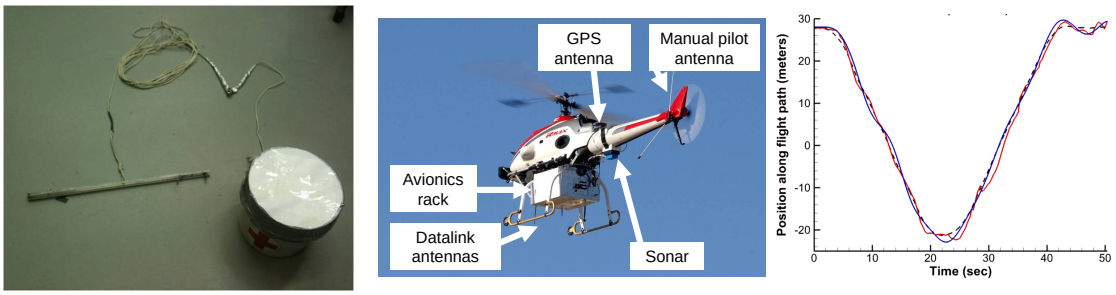
5.1 Flight Test Vehicle and Simulation Environment

Simulation studies and flight tests used in the validation of the proposed trajectory optimization framework were conducted with a modified Yamaha RMAX helicopter UAV, dubbed the GTMax [57]. Custom avionics and the flight control software (GUST) needed for autonomous flight were designed at the Georgia Tech Unmanned Aerial Vehicle Research Facility (UAVRF). The GTMax weighs 157 pounds (with avionics) and has a 10.2 foot main rotor diameter. The vehicle is equipped with an extensive sensor suite including an Inertial Science IMU, short-range sonar, magnetometer, and differential GPS [23, 57]. In addition, the vehicle is instrumented with a Prosilica GC 1380 camera for vision-based operations [17, 74, 117]. Figure 5.1b shows the GTMax test platfrom.

Two separate onboard processes, primary and secondary, are used for all onboard computational requirements and are executed on a single onboard computer with an i7 processor. The primary process performs the basic functionality needed for navigation, guidance, and control of the vehicle. An adaptive neural network model inversion flight controller, the vehicle's baseline controller, was used for all simulation studies and the presented flight test [59]. The secondary process is used to perform image capture and processing needed for the presented load state estimator. In addition, the proposed trajectory optimization framework was executed in the secondary process. The proposed computational architecture is described in detail in Section 5.3.4 (Figure 5.6).

The suspended load weighs 8.4 pounds resulting in a load mass ratio (LMR) of 0.051 and is attached to the vehicle with a 46 foot braided nylon line. The bucket is sealed with a white cover to assist with tracking as discussed in Section 5.2. In addition, the load can be jettisoned by the safety pilot with a radio transmitter. Figure 5.1a shows the suspended load used in the flight tests.

The Georgia Tech UAV Simulation Tool (GUST), developed at the UAVRF, is



(a) (b) (c)

Figure 5.1: (a) Suspended load used during the presented flight tests. (b) Annotated image of the GTMax vehicle test platform. (c) A comparison of simulated and measured responses of the suspended load. Dotted lines indicate the measured position of the vehicle and red and blue lines indicate the measured and simulated position of the load, respectively.

used to simulate flight tests in order to demonstrate the efficacy of the proposed trajectory optimization framework [60,61]. GUST contains a high-fidelity vehicle and environment model, onboard flight control software, and ground station software and is used for rapid development and testing of software and hardware for all aspects of autonomous vehicle operation. The vehicle model is a rigid body six degree of freedom model with additional engine, fuel, and rotor dynamics. In addition, the vehicle model simulates sensor noise, communication delay, real world location, orientation, and actuator dynamics and saturation. It should be noted that the vehicle model is not propagated with a variational integrator. Since accuracy, not computational efficiency, is important during simulation, a 2^{nd} order Euler method is used and integration occurs at 400 Hertz. Therefore, inconsistency between the propagation methods used in the optimization framework and the simulation environment had a negligible affect. Furthermore, a simulated camera capturing rendered graphics is used to predict the real-time performance of vision-based operations. External disturbances such as turbulence and wind are also simulated. The suspended load is simulated by GUST with a rigid body six degree of freedom model with flexible cable dynamics.

In addition, aerodynamic loading is represented with a reduced order model based on a quasi-steady aerodynamics, unsteady effects on body motion, and unsteady effects of vortex shedding [99]. This aerodynamic model significantly improves the fidelity of the suspended load simulation when compared to a simple model that only considers quasi-steady drag. As shown in Figure 5.1c, preliminary flight tests show a good correspondence between the model and the observed behavior of the load. Further details on the high fidelity aerodynamic loading model and the preliminary flight tests results can be found in Reference [99].

5.2 Vision-Based Estimation of Load

The presented vision-based suspended load state estimator is an extension to the suspended load state estimator previously presented in References [7] and [10]. The proposed framework provides estimates of the suspended load state (swing angle and rate) through augmentation of an existing vehicle navigation system. The suspended load is not instrumented with any sensors and only images captured from a downward facing camera provide sensing information. Furthermore, the estimated vehicle state is use to drive the suspended load's process model and an unscented Kalman filter produces estimation updates.

5.2.1 Process Model

The process and sensor model used in the presented framework are the same as those used in the estimator presented in References [7] and [10]. The system model is reviewed here and the cited references are referred to for further details.

A simple pendulum process model is used to described the suspended load system. Specifically, the suspended load is modeled as a point mass where the position of the load is given by generalized coordinates θ_w and ϕ_w (swing angles) which can be considered as a 2-1 Euler angle rotation about the attachment point on the vehicle. The length of the cable is assumed to be constant and the line never becomes slack.

Figure 5.2.a shows the representation of the system. The equations of motion of the system are obtained as

$$\ddot{\theta_w} L = -\cos(\theta_w)\cos(\phi_w)\ddot{x}_v + \sin(\theta_w)\cos(\phi_w)(\ddot{z}_v - g), \tag{5.1}$$

$$\dot{\theta}_w(t_0) = \dot{\theta}_{w0}, \theta_w(t_0) = \theta_{w0},\ t \geq 0,$$

$$\ddot{\phi_w} L = \sin(\theta_w)\sin(\phi_w)\ddot{x}_v + \cos(\theta_w)\sin(\phi_w)(\ddot{z}_v - g) - \cos(\phi_w)\ddot{y}_v, \tag{5.2}$$

$$\dot{\phi}_w(t_0) = \dot{\phi}_{w0}, \phi_w(t_0) = \phi_{w0},$$

where g is the gravitational constant, $\ddot{x}_v$, $\ddot{y}_v$, and $\ddot{z}_v$ are the vehicle's translational accelerations in the inertial frame and L is the total length of the pendulum (attachment point to center of mass of the load). The position of the load is given as

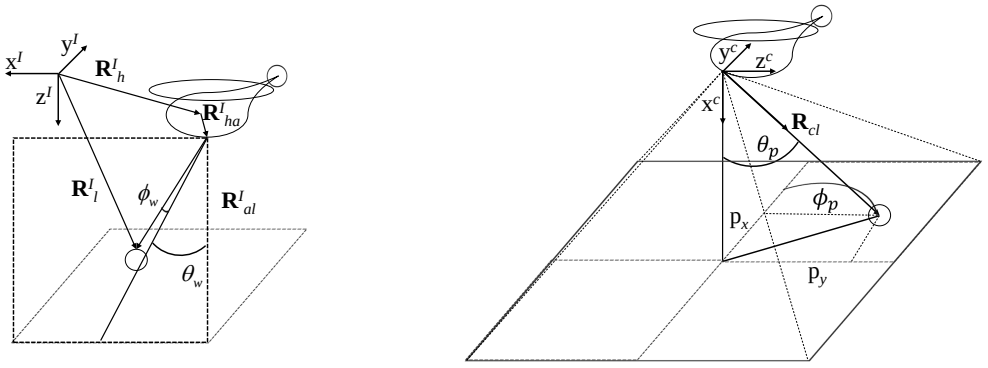
$$\mathbf{R}_l^I = \mathbf{R}_v^I + \mathbf{R}_{ha}^I + \begin{bmatrix} \sin(\theta_w)\cos(\phi_w) \\ \sin(\phi_w) \\ \cos(\theta_w)\cos(\phi_w) \end{bmatrix} L,$$

where $\mathbf{R}_v^I$ is the position of the vehicle in the inertial frame, $\mathbf{R}_l^I$ is the position of the load in the inertial frame, and $\mathbf{R}_{ha}^I$ is the vector from the center of mass of the vehicle to the attachment point in the inertial frame. Furthermore, the velocity of the load

$$\dot{\mathbf{R}}_l^I = \dot{\mathbf{R}}_v^I + \dot{\mathbf{R}}_{ha}^I + \begin{bmatrix} \cos(\theta_w)\cos(\phi_w) & -\sin(\theta_w)\sin(\phi_w) \\ 0 & \cos(\phi_w) \\ -\sin(\theta_w)\cos(\phi_w) & -\cos(\theta_w)\sin(\phi_w) \end{bmatrix} \begin{bmatrix} \dot{\theta}_w \\ \dot{\phi}_w \end{bmatrix} L,$$

5.2.2 Sensor Model

The sensor information extracted by processing the captured images from a downward facing camera is the unit vector, $\mathbf{R}_{cl}^v$, from the camera to the white disk on top of the load in the vehicle frame. The estimate of the load position relative to the attachment



(a) (b)

Figure 5.2: (a) Diagram of the process model used in the vision-based estimator. (b) Diagram of the camera coordinate frame. Figures recreated from those found in Reference [10].

.

point of the helicopter is given by

$$\hat{\mathbf{R}}_{al}^{I} = \begin{bmatrix} \sin(\hat{\theta}_w)\cos(\hat{\phi}_w) \\ \sin(\hat{\phi}_w) \\ \cos(\hat{\theta}_w)\cos(\hat{\phi}_w) \end{bmatrix} L,$$

where $\hat{\theta_w}$ and $\hat{\phi_w}$ are the estimated swing angles. Finally, the predicted measurement needed to implement an unscented Kalman filter is found as [105]

$$\hat{\mathbf{R}}_{cl}^{v} = \frac{\mathbf{T}_{vI}\hat{\mathbf{R}}_{al}^{I} + \mathbf{R}_{ha}^{v} - \mathbf{R}_{hc}^{v}}{|\mathbf{T}_{vI}\hat{\mathbf{R}}_{al}^{I} + \mathbf{R}_{ha}^{v} - \mathbf{R}_{hc}^{v}|}, \tag{5.3}$$

where $\mathbf{R}_{hc}^{v}$ is the vector from the center of mass of the vehicle to the camera in the vehicle frame, $\mathbf{R}_{ha}^{v}$ is the vector from the center of mass of the vehicle to the helicopter attachment point in the vehicle frame, and $\mathbf{T}_{vI}$ is the transformation matrix from the inertial frame to the vehicle frame.

5.2.3 Image Capture and Processing

As previously mentioned, images from a downward facing camera are processed in order to obtain sensor information to aid in the estimation of the suspended load

state. In order to assist in the localizing of the load a white disc is placed on top of the load. A probability-based mapping is then performed on a gray-scaled image in order to identify the likelihood that pixels from that image are centers to a white disk of a specific radius. Specifically, the pixels on the white disc are assumed to be normally distributed parameterized with user-defined mean and variance. Using the cumulative distribution function of a normal distribution the mapping produces a score of how "white" the image around a pixel is:

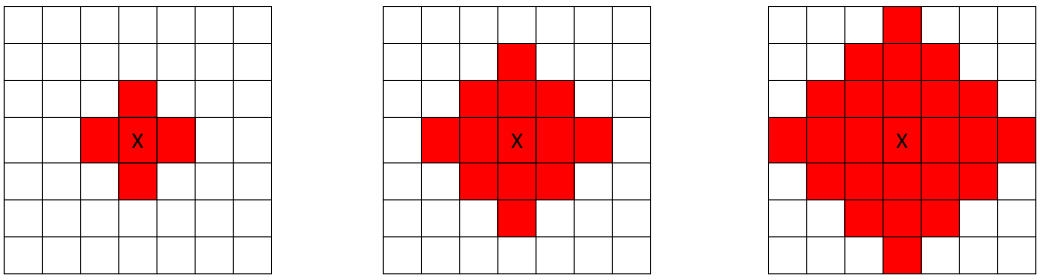
$$S_{(r,s)} = \prod_{(i,j)\in\mathcal{R}} \frac{1}{2}\Big(1 + \operatorname{erf}\left(\frac{P_{(r+i,s+j)} - \mu}{\sigma\sqrt{2}}\right)\Big) \tag{5.4}$$

where $S_{(r,s)} \in [0,1]$ is the pixel score, $\mathcal{R}$ is the set of pixel translations that approximates a disk of a specific radius, and $P_{(r,s)} \in [0,255]$ is the gray-scale value of the (r,s) pixel where $P_{(r,s)} = 0$ if the pixel is black and $P_{(r,s)} = 255$ if the pixel is white. As shown in Figure 5.3, set $\mathcal{R}$ is defined as

$$\mathcal{R} = \{(i,j) : i,j \in \mathbb{N},\ |i| + |j| \leq r\} \tag{5.5}$$

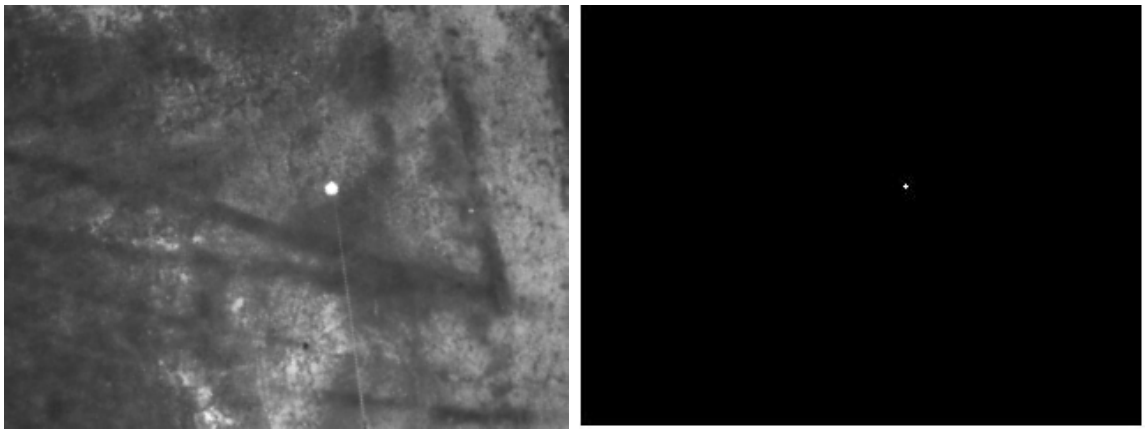
where r is the radius of the disc. As an example, if a disc of unity radius is considered $(r = 1)$ then $\mathcal{R} = \{(0,0),(1,0),(0,1),(-1,0),(0,-1)\}$. In order for a pixel to achieve a score close to unity all of its neighboring pixels and itself should be "whiter than most other pixels" given the specified mean and variance. If the pixel with the highest score meets user-defined minimum requirements it is considered a positive detection of the load. Figure 5.4 gives an example of the mapping produced by (5.4) given $\mu = 220$, $\sigma = 5$ and $r = 2$ (pixel scores scaled to produce a post-processed gray-scaled image). Notice the white pixels in the post-process image correspond to the center of the load. Note that the select of r is determined by the length of the cable and the camera resolution. Furthermore, if possible, σ and μ should be selected such that the environment does not produce scores above the defined threshold.

Note the use of the cumulative distribution function of a normal distribution in equation (5.4) is arbitrary. Other functions that map pixel values to $[0,1]$ can be



(a) (b) (c)

Figure 5.3: Depictions of the set $\mathcal{R}$ for a radius of 1 (a), 2 (b), and 3 (c).



(a) (b)

Figure 5.4: (a) Typical image from a flight test and (b) the post-processed image when $\mu = 220$, $\sigma = 5$ and $r = 2$.

used. However, in order for the function to be effective a pixel should receive a very low score if one of its neighbors is significantly different from the *ideal* pixel. Finally, it should be noted that the target disc can be made a different color depending on the application. For example, the disc can be red or orange in operations with thick snow/ice on the ground. In this case, a pixel will achieve a high score close if its neighbors and itself are "redder than most other pixels". Appropriate changes to equation (5.4) can be made trivially.

Once a pixel is selected, the coordinates of the pixel (p_x, p_y), as shown in Figure 5.2.b, are used to compute the sensed unit vector from the camera to the load. First, the coordinate p_y is scaled based on the image dimensions as $\bar{p}_y = \frac{l_x}{l_y} p_y$ where l_x

and l_y are the width and height of the image in pixels. Next, characterize angles of computed as

$$\theta_p = \rho\sqrt{p_x^2 + \bar{p}_y^2}, \quad \text{and} \quad \phi_p = \text{atan2}(\bar{p}_y, p_x) \tag{5.6}$$

where ρ is ratio of the field of view and the width in pixels of the image. The sensed unit vector from the camera to the load can then be computed as

$$\mathbf{R}_{cl}^{v} = \mathbf{T}_{vc} \begin{bmatrix} \cos(\theta_p) \\ \sin(\theta_p)\cos(\phi_p) \\ \sin(\theta_p)\sin(\phi_p) \end{bmatrix},$$

where $\mathbf{T}_{vc}$ is the transformation matrix from the camera frame to the vehicle frame.

5.3 Trajectory Optimization Framework

5.3.1 System Model and Cost Function

Since guidance (not control) was the goal of the optimization framework a greatly simplified model was used to represent the complex and coupled rotorcraft-suspended load system. The model consist of two point masses under a gravitational field with a holonomic constraint restricting the distance between them to equal a defined length. Furthermore, three control forces can accelerate the mass representing the vehicle. Figure 5.5 shows a depiction of the simplified model. Variational integrators were used to propagate and linearize the system in the DDP and projection-based frameworks. The Lagrangian of the system is given as

$$L(q, \dot{q}) = \frac{1}{2}M(\dot{x}_{\text{v}}^2 + \dot{y}_{\text{v}}^2 + \dot{z}_{\text{v}}^2) + \frac{1}{2}m(\dot{x}_{\text{l}}^2 + \dot{y}_{\text{l}}^2 + \dot{z}_{\text{l}}^2) + mgz_{\text{l}} + mgz_{\text{v}} \tag{5.7}$$

where $q = [x_{\text{v}}, y_{\text{v}}, z_{\text{v}}, x_{\text{l}}, y_{\text{l}}, z_{\text{l}}]^{\text{T}}$, M is the mass of the vehicle and m is the mass of the load. The holonomic constraint is a wire constraint defined as

$$h(q) = (x_{\text{v}} - x_{\text{l}})^2 + (y_{\text{v}} - y_{\text{l}})^2 + (z_{\text{v}} - z_{\text{l}})^2 - L^2 \tag{5.8}$$

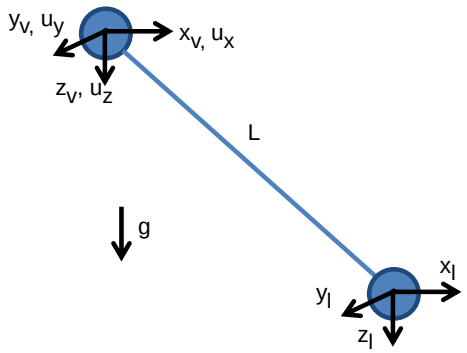


Figure 5.5: Diagram of the modeled suspended load system.

where L is the cable length (slacked lines not considered). Finally, the left and right discrete forces needed to evaluate the constrained-forced DEL equations (2.86) are defined as

$$F_{\mathrm{d}}^{\pm}(q_k, q_{k+1}, u_k) = \begin{bmatrix} \frac{1}{2}u_x\Delta t \\ \frac{1}{2}u_y\Delta t \\ \frac{1}{2}u_z\Delta t \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.9}$$

where Δt is the discretization time step. A variational integrator can now be derived for the system.

The reference trajectory based cost function used in both optimization processes is given as

$$\begin{aligned} v(q,u,t) = \int_{t_0}^{t_{\mathrm{f}}} \left(Q_1(q(\tau) - q_{\mathrm{ref}}(\tau))^2 + Q_2(\dot{q}(\tau) - \dot{q}_{\mathrm{ref}}(\tau))^2 + R(u(\tau) - u_{\mathrm{trim}}(\tau))^2\right) \mathrm{d}\tau \\ + Q_{\mathrm{f}}(q(t_{\mathrm{f}}) - q_{\mathrm{ref}}(t_{\mathrm{f}}))^2. \end{aligned} \tag{5.10}$$

where $(q_{\mathrm{ref}}(t), \dot{q}_{\mathrm{ref}}(t))$ is a reference system trajectory, $Q_1, Q_2 \geq 0$ are the state tracking error cost, $Q_{\mathrm{f}} \geq 0$ is the final state tracking error cost, and $R > 0$ is the control cost.

Though the reference system trajectory should be reasonable (i.e. limited vehicle velocity, acceleration and jerk), it does not have to be dynamically feasible. For example, a reference trajectory with no load swing ($x_{\mathrm{v}}(t) = x_{\mathrm{l}}(t)$ and $y_{\mathrm{v}}(t) = y_{\mathrm{l}}(t)$) can be valid.

The variational integrator allows for reliable propagation of the system configuration and linearization of the discrete dynamics about a given trajectory. Furthermore, as noted in Chapter 3 the performance of iterative optimization algorithms is far less dependent on the discretization time step when a variational integrator is used. Therefore, as shown in the proceeding sections the DDP and projection-based frameworks were real-time feasible due, in part, to a relatively large time step. In addition, modeling the simplified system with a wire constraint allows the reference trajectory for the vehicle and load to be decouple. As a result, the cost associated with the load trajectory is not directly a function of the vehicle trajectory and results in a simple cost function. Lastly, as discussed in Section 2.3.1.2, the variational integration ensures that each discrete system configuration q_k will observe the defined constraint.

5.3.2 Differential Dynamic Programming

The differential dynamic programming (DDP) algorithm outlined in Section 2.2.1 was one of the algorithms implemented to solve the trajectory optimization problem described above (deterministic dynamics were considered). The initial nominal discrete inputs and associated state trajectory were the output of the previous optimization cycle shifted to the current initial time t_0. Furthermore, during each optimization cycle a maximum of 5 iterations were performed before a control and state trajectory were outputted. The time horizon ($t_{\mathrm{f}} - t_0$) was 12 seconds and $\Delta t = 0.1$. In addition,

the cost parameters were defined as

$$Q_1 = \text{diag}([0.5, 0.5, 0.5, 3.0, 3.0, 3.0]), \tag{5.11}$$

$$Q_2 = \text{diag}([0.5, 0.5, 0.5, 3.0, 3.0, 3.0]), \tag{5.12}$$

$$R = \text{diag}([1.0, 1.0, 1.0]), \tag{5.13}$$

$$Q_\text{f} = \text{diag}([0.5, 0.5, 0.5, 2.0, 2.0, 2.0]). \tag{5.14}$$

The Armijo search parameters were set as $\alpha = 0.00001$ and $\beta = 0.7$ and the maximum number of possible Armijo iterations was set to 15. If the Armijo search failed (i.e. the Armijo cost criteria was not met after 15 iterations) the current optimization cycle was terminated and the current nominal input and state trajectory were outputted. Furthermore, if the maximum possible change in the cost for a computed descent direction, $\delta x^\text{T} \nabla_x v(x, u, t) + \delta u^\text{T} \nabla_u v(x, u, t)$, was less than 1×10^{-6} the optimization cycle was terminated and the current nominal input and state trajectory were outputted.

5.3.3 Projection-Based Optimization

The projection-based optimization methodology outlined in Section 2.2.2 was also used to solve the considered trajectory optimization problem. Parameters associated with the cost function, time horizon and discretization, and Armijo search were the same as those in the DDP algorithm. In addition, the LQR cost used to find the time-varying feedback gain $K(t)$ needed to define the projection operator was selected as:

$$G = \text{I}_{12} \tag{5.15}$$

$$G_\text{f} = \text{I}_{12} \tag{5.16}$$

$$F = \text{I}_3 \tag{5.17}$$

During each optimization cycle the descent direction during the first 6 iterations was found using a quasi-Newton method (2.55). In the remaining iterations the Newton

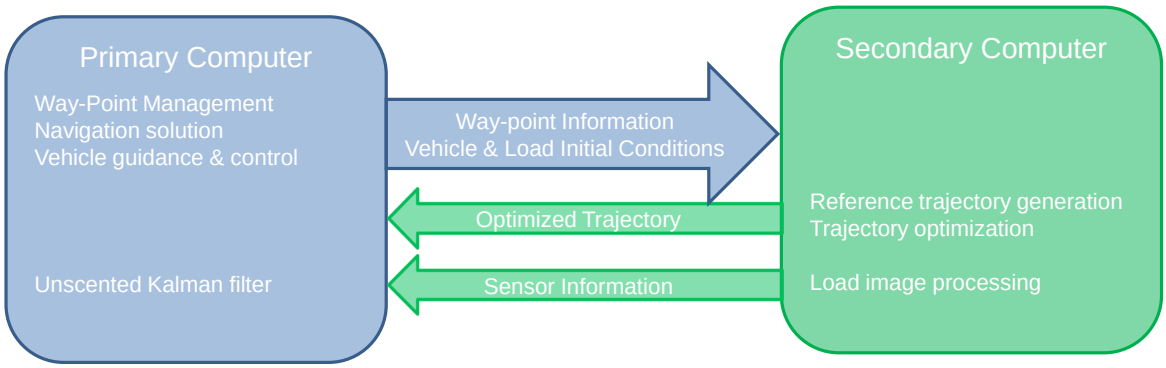


Figure 5.6: Overview of the implemented communication and computer architecture.

descent method (2.53) was used. A maximum of 10 iterations were performed. Like in the DDP algorithm the cycle was terminated if the Armijo search failed or if the descent direction could only yield a very small change in the cost function.

5.3.4 Integration

The optimization process described in the previous section can be implemented into an existing autonomous vehicle system. Figures 5.6 and 5.7 give an overview of how the proposed optimization framework can be integrated into an existing on-board computer. It is assumed the existing guidance, navigation, and control system is able to follow reasonable reference trajectories relatively accurately. A second thread in the secondary computer is used exclusively to solve the selected optimization algorithm. As a result, other processes contained in the secondary computer (i.e. image processing, communication to primary computer, etc.) are not affected by the computationally heavy optimization process. Simulation studies and flight tests have shown that the stored optimization solution using either the DDP or projection-based algorithm is updated at an average rate faster the 10 Hertz. Note that computational time is much smaller than the optimization horizon of 12 seconds. Therefore, as confirm in simulation and flight tests, real-time trajectory optimization is achieved by the proposed framework.

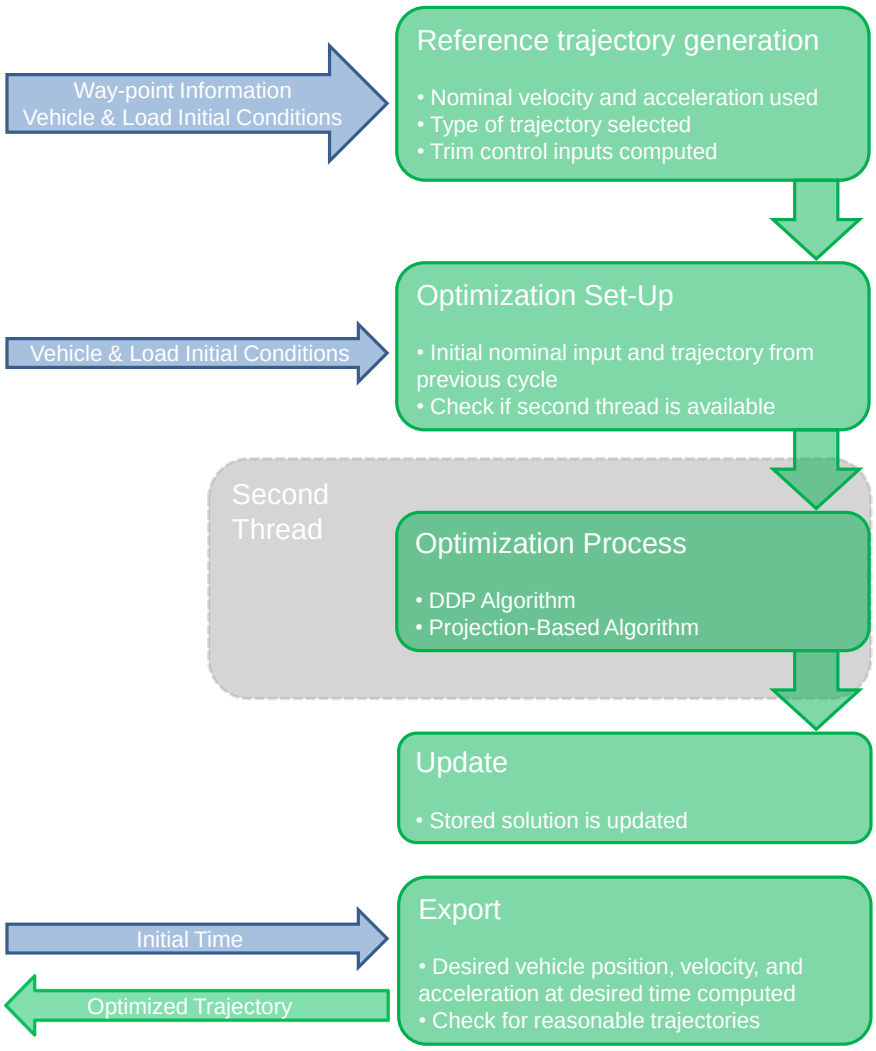


Figure 5.7: Overview of the optimization trajectory optimization process.

5.4 Simulation Implementation Results

In order to test the effectiveness of the proposed framework a set of 4 basic maneuvers were performed. First, it was ensured that the vehicle could maintain the load at a fixed location (Maneuver 1: Hover). In order to test for robustness in simulation, the response to a instant change in load velocity was also investigated. The second and third reference trajectories were constructed to go from two way-points with a given prescribed maximum velocity and acceleration (Maneuver 2a: Way-point and Maneuver 2b: Way-point). Maneuver 2a reference trajectory was generated using

a maximum velocity and acceleration of 10 feet/sec and 2 feet/sec^2, respectively, and traveled about 46 feet and 115 feet along the x-axis and y-axis, respectively. Maneuver 2b reference trajectory was generated using a maximum velocity and acceleration of 10 feet/sec and 2 feet/sec^2, respectively, and traveled about 50 feet and 1 foot along the x-axis and y-axis, respectively. The final reference trajectory was constructed to inscribed a circle with a 50 foot radius every 75 seconds such that its tangential velocity is 4.19 feet/sec (Maneuver 3: Circle). Table 5.1 gives a summary of maneuvers.

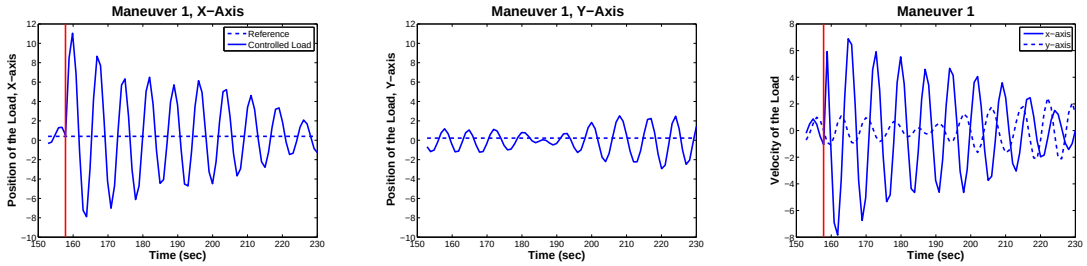
The reference trajectory for the load and the vehicle were identical except for a offset in the altitude. Therefore, the generated reference trajectories are not dynamically feasible, but are idealized (and reasonable) trajectories. Note that trajectories near the idealized trajectories are sought. The cost matrices, Q_1, Q_2, and Q_f, are designed to prioritize which states should be closer to their idealized values. Note that the consider cost function (5.11) - (5.14) penalizes deviations associated with the load more than vehicle deviations. Therefore, in this case the load's trajectory is prioritized. As a result, the optimization process favors adjusting the vehicle's trajectory in order to maintain the load closer to the reference trajectory. Furthermore, specifying dynamically feasible trajectories would require more on-line computation and, furthermore, prescribing non-ideal reference trajectories may lead to sub-optimal performance.

5.4.1 Unoptimized Response

This section presents results obtained from simulations studies where no optimized trajectory was computed and the reference trajectory was used in its place. Therefore, the vehicle simply followed the generated reference trajectory without any consideration of the response of the load or a cost function.

Table 5.1: Summary of maneuvers were performed.

Maneuver	Designation	Description
1	Hover	Maintains a fixed position.
2a	Way-point	Travels 46 and 115 feet along the x-axis and y-axis, respectively, with a maximum velocity and acceleration of 10 feet/sec and 2 feet/sec^2, respectively.
2b	Way-point	Travels 50 and 1 feet along the x-axis and y-axis, respectively, with a maximum velocity and acceleration of 10 feet/sec and 2 feet/sec^2, respectively.
3	Circle	Inscribes a 50 foot radius circle every 75 seconds.

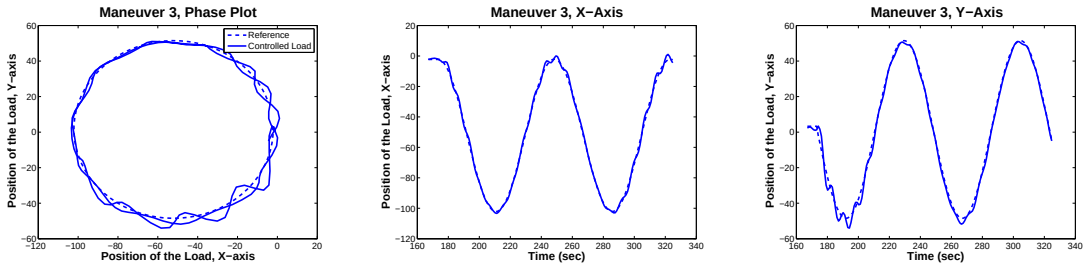


(a) (b) (c)

Figure 5.8: Simulation-Unoptimized Maneuver 1 (a) and (b): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Load velocity as a function of time. Solid and dotted lines represent the velocity along the x-axis and y-axis, respectively.

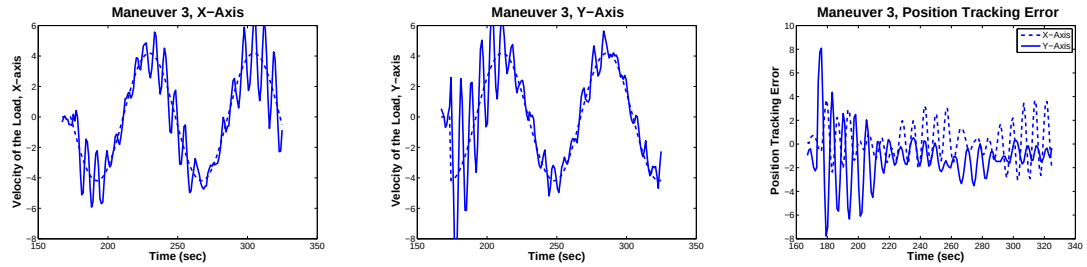
Figure 5.8 shows the trajectory history of the load when Maneuver 1 was performed. At $t = 157.80$ the velocity of the load along the x-axis was instantaneously change to 10 feet/sec. Note that the velocity of the load slowly decays and the load retains a 12 foot oscillation about 30 seconds after the induce velocity disturbance.

Figures 5.9 and 5.10 shows the trajectory history of the load when Maneuver 3 was performed. As seen in Maneuver 1, large load oscillations are found in the system response. Note the large oscillations along the y-axis at the beginning of the maneuver and those along the x-axis at the end of the maneuver.



(a) (b) (c)

Figure 5.9: Simulation-Unoptimized Maneuver 3 (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively.



(a) (b) (c)

Figure 5.10: Simulation-Unoptimized Maneuver 3 (a) and (b): Load velocity as a function of time. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the tracking error along the x-axis and y-axis, respectively.

5.4.2 Differential Dynamic Programming Implementation

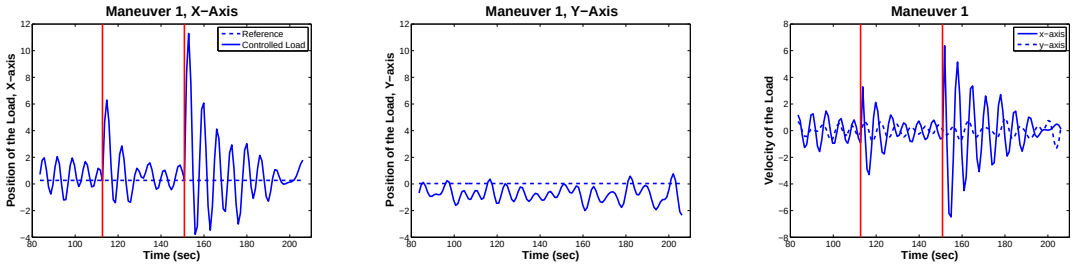
This section presents results obtained from simulations studies where the DDP algorithm was used for optimization.

Figure 5.11 shows the trajectory history of the load when Maneuver 1 was performed. At $t = 112.72$ and $t = 150.88$ the velocity of the load along the x-axis was instantaneously change to 5 feet/sec and 10 feet/sec, respectively. Note that the vehicle is able to maintain the load close to the desire position and at "steady state" the load tracking error is always less than 2.5 feet. Furthermore, the system is able to decrease the velocity of the load after the induced disturbance. Figure 5.12 shows the load position estimation errors (or residuals). Note there is a bias in both directions. These biases are caused by errors in the vehicle navigation solution. In particular, in the estimation of the vehicle's attitude and accelerometer bias terms. Any error in the estimate of the vehicle's state affects the sensor model used to update the estimate of the load's state. As shown in Figure 5.11, these biases are manifested in the trajectory history of the load.

Figures 5.13 and 5.14 shows the trajectory history of the load when Maneuver 2a was performed. Though experiencing large tracking errors during transitional portions of the reference trajectory the suspended load is able to track the reference trajectory generally well. As shown in Figure 5.14 some of the tracking error is caused by a delayed response.

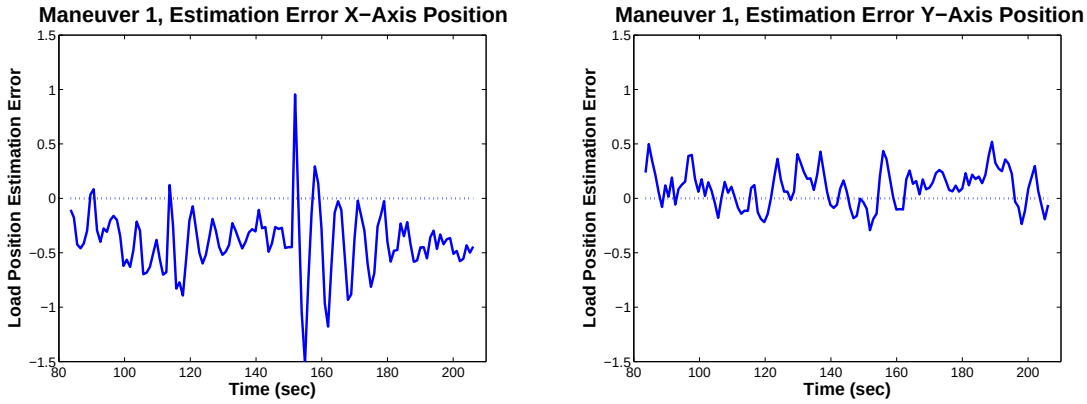
Figures 5.15 and 5.16 shows the trajectory history of the load when Maneuver 2b was performed. As in Maneuver 2a the load was able to track the reference trajectory well and there is a noticeable delayed response.

Figures 5.17 and 5.18 shows the trajectory history of the load when Maneuver 3 was performed. Note that following an initial transient phase the load is able to track the reference load with a relatively small error.



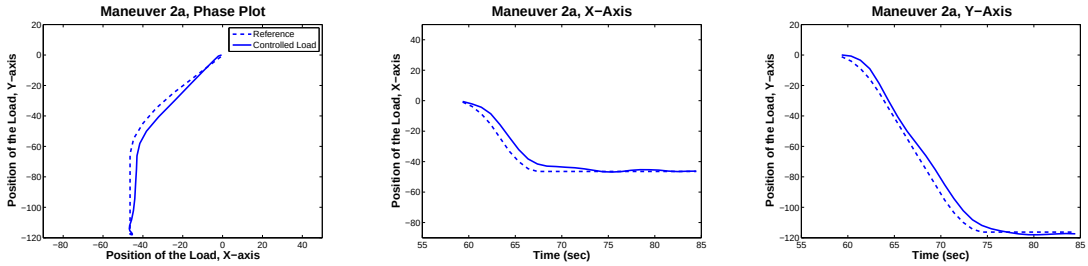
(a) (b) (c)

Figure 5.11: Simulation-DDP Maneuver 1 (a) and (b): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Load velocity as a function of time. Solid and dotted lines represent the velocity along the x-axis and y-axis, respectively.



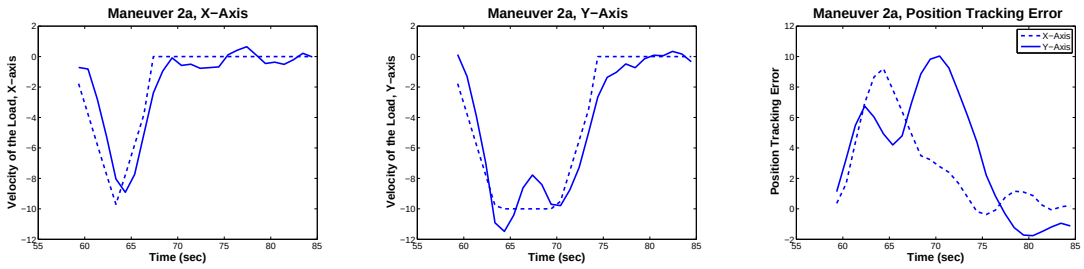
(a) (b)

Figure 5.12: Simulation-DDP Maneuver 1 (a) and (b): The estimation error of the load's position along the x-axis and y-axis, respectively



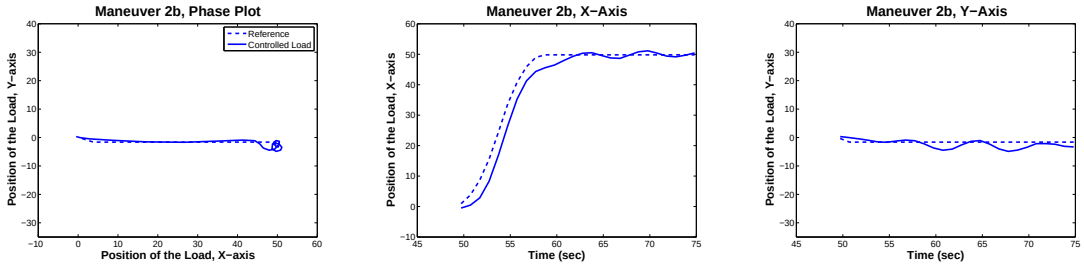
(a) (b) (c)

Figure 5.13: Simulation-DDP Maneuver 2a (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively.



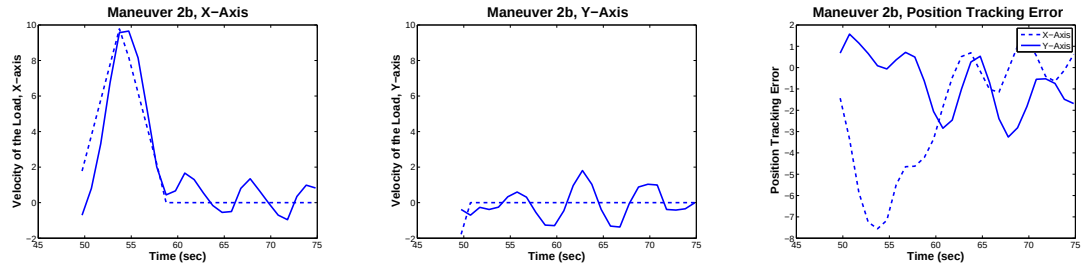
(a) (b) (c)

Figure 5.14: Simulation-DDP Maneuver 2a (a) and (b): Load velocity as a function of time. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the tracking error along the x-axis and y-axis, respectively.



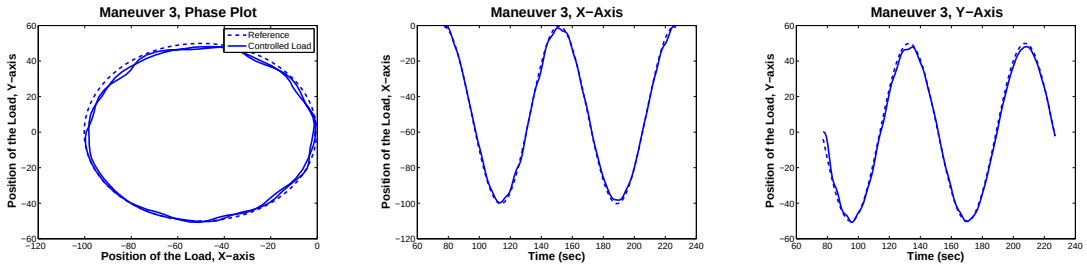
(a) (b) (c)

Figure 5.15: Simulation-DDP Maneuver 2b (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively.



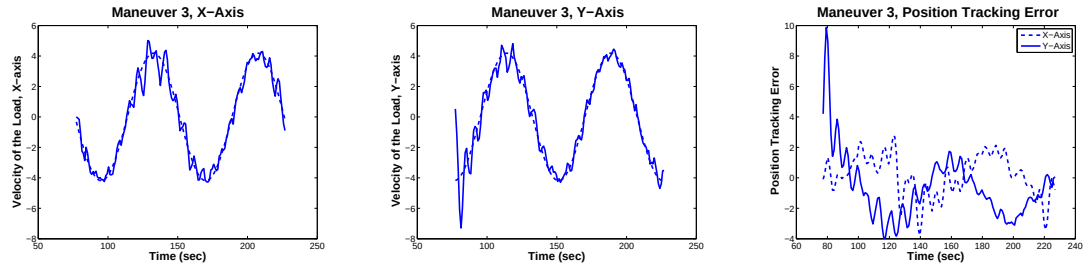
(a) (b) (c)

Figure 5.16: Simulation-DDP Maneuver 2b (a) and (b): Load velocity as a function of time. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the tracking error along the x-axis and y-axis, respectively.



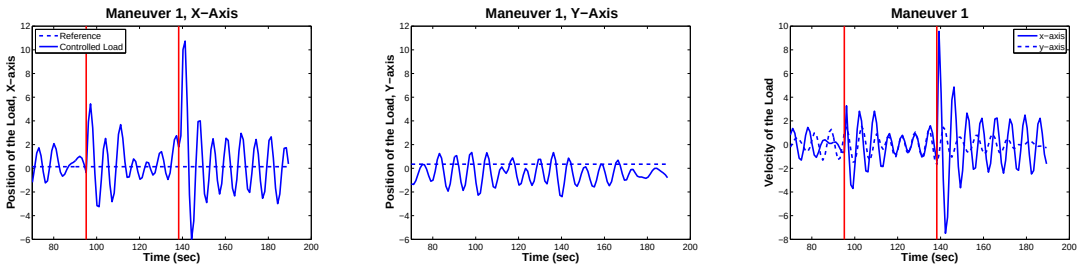
(a) (b) (c)

Figure 5.17: Simulation-DDP Maneuver 3 (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively.



(a) (b) (c)

Figure 5.18: Simulation-DDP Maneuver 3 (a) and (b): Load velocity as a function of time. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the tracking error along the x-axis and y-axis, respectively.



(a) (b) (c)

Figure 5.19: Simulation-Projection-Based Optimization Maneuver 1 (a) and (b): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Load velocity as a function of time. Solid and dotted lines represent the velocity along the x-axis and y-axis, respectively.

5.4.3 Projection-Based Optimization Implementation

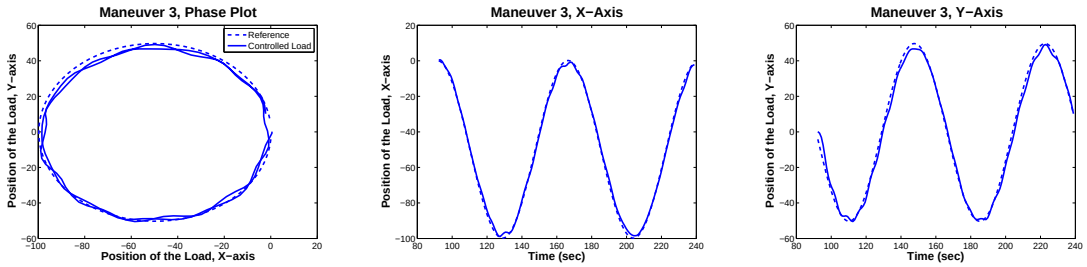
This section presents results obtained from simulations studies where the projection-based optimization algorithm was used for optimization.

Figure 5.19 shows the trajectory history of the load when Maneuver 1 was performed. At $t = 95.14$ and $t = 138.24$ the velocity of the load along the x-axis was instantaneously change to 5 feet/sec and 10 feet/sec, respectively. The proposed framework was able to reduced the induced velocity and maintain the load relatively near the desired position. Note that the response of the system is very similar to that shown in Figure 5.11.

Figures 5.20 and 5.21 shows the trajectory history of the load when Maneuver 3 was performed. Again, the performance of the framework is quite similar to that displayed in Figures 5.15 and 5.16

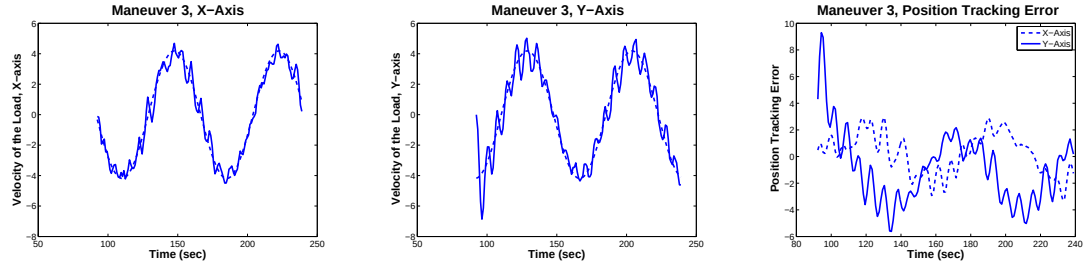
5.4.4 Effect of State Estimator Error

This section presents results obtained from simulations studies where the DDP algorithm was used for optimization and when the system's navigation solution was error free (estimated state equaled simulation state). Maneuver 1 was performed in order



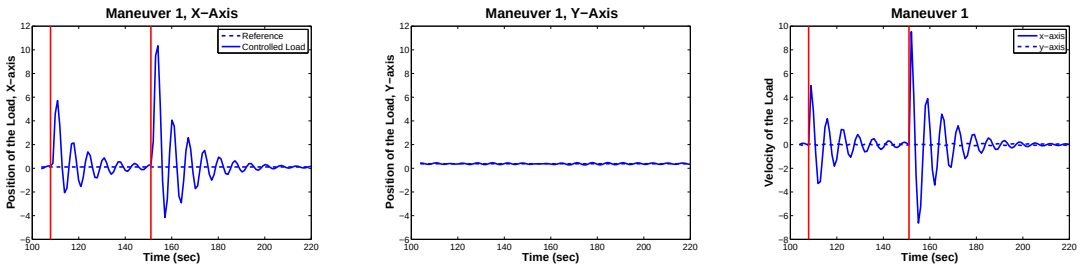
(a) (b) (c)

Figure 5.20: Simulation-Projection-Based Optimization Maneuver 3 (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively.



(a) (b) (c)

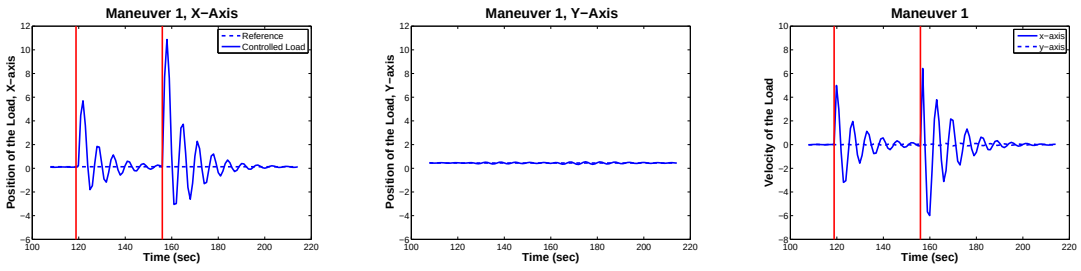
Figure 5.21: Simulation-Projection-Based Optimization Maneuver 3 (a) and (b): Load velocity as a function of time. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the tracking error along the x-axis and y-axis, respectively.



(a) (b) (c)

Figure 5.22: Simulation-DDP (with error free navigation) Maneuver 1 (a) and (b): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Load velocity as a function of time. Solid and dotted lines represent the velocity along the x-axis and y-axis, respectively.

to investigate if a significant performance improvement could be obtained if the navigation solution was error free. Figure 5.22 shows the trajectory history of the load. At $t = 107.87$ and $t = 151.04$ the velocity of the load along the x-axis was instantaneously change to 5 feet/sec and 10 feet/sec, respectively. Note that the vehicle is able to maintain the load close to the desire position and at "steady state" the load tracking error is negligible. Furthermore, the system is able to decrease the velocity of the load at a faster rate when compared to the trajectory shown in Figure 5.11. Figure 5.23 shows the trajectory history of the load if a more aggressive cost function parameterized as $Q_1 = \text{diag}([1.0, 1.0, 1.0, 6.0, 6.0, 6.0])$, $Q_2 = \text{diag}([1.0, 1.0, 1.0, 6.0, 6.0, 6.0])$, $R = \text{diag}([1.0, 1.0, 1.0])$, and $Q_\text{f} = \text{diag}([1.0, 1.0, 1.0, 4.0, 4.0, 4.0])$, is used. At $t = 118.82$ and $t = 155.93$ the velocity of the load along the x-axis was instantaneously change to 5 feet/sec and 10 feet/sec, respectively. The velocity of the load decays at a slightly faster rate when compared Figure 5.22. Furthermore, it has been observed in simulation studies not presented here that the performance of the system when the navigation solution is *not* error free deteriorates if the aggressive cost function is selected.



(a) (b) (c)

Figure 5.23: Simulation-DDP (with error free navigation and aggressive cost function) Maneuver 1 (a) and (b): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dotted lines represent the load state and reference trajectory, respectively. (c): Load velocity as a function of time. Solid and dotted lines represent the velocity along the x-axis and y-axis, respectively.

5.5 Flight Test Results

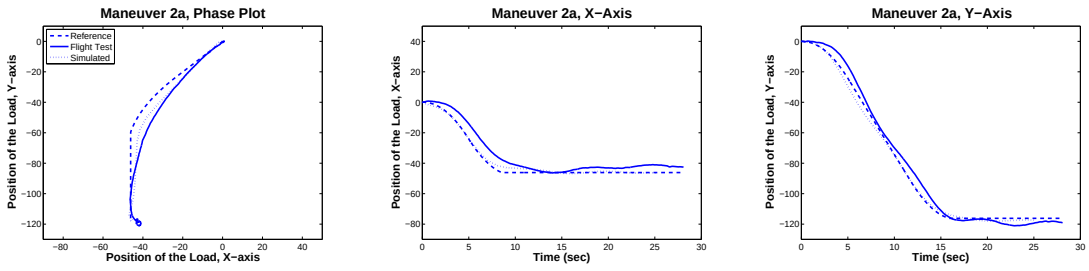
This section presents results obtained from a flight test where the DDP algorithm was used for optimization. The flight test was perform on March 18, 2015 at around 10am. An air temperature of 63.3 degrees Fahrenheit and a wind speed of 2.3 miles per hour in varied directions was reported[1].

Figures 5.24 and 5.25 shows the *estimated* trajectory history of the load when Maneuver 2a was performed. Note that the response of the system is very similar to the response seen in simulation (Figures 5.13 and 5.14). As expected, the tracking errors in the flight test are larger than the ones seen in simulation. However, given that the length of the line is 46 feet and the system is subject to environmental factors like wind a tracking error of about 6 feet is reasonable. Furthermore, the estimation bias shown in simulation may also have contributed to the tracking errors.

Figures 5.26 and 5.27 shows the trajectory history of the load when Maneuver 2b was performed. As seen in Maneuver 2b, the response of the system is very similar to the response seen in simulation (Figures 5.15 and 5.16).

Figures 5.28 and 5.29 shows the trajectory history of the load when Maneuver 3

[1]Weather report for area code 31805 and the specified date provided by Weather Undergound.



(a) (b) (c)

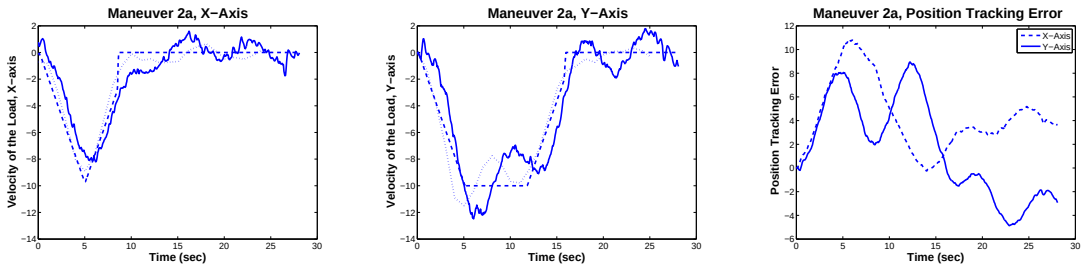
Figure 5.24: Flight Test-DDP Maneuver 2a (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.13), and reference trajectory, respectively.

was performed. As seen in the other two maneuvers performed, the results obtained through simulation studies are very similar to those obtained in the flight test. Figure shows the histograms of the computational times related to image processing and optimization when Maneuver 3 was performed. In the recorded data set the computational time required to perform an optimization cycle (maximum of 5 iterations) had a mean of 0.0965 seconds and a standard deviation of 0.0167. This relatively small computational time allows for real-time trajectory optimization and is possible due to the large discretization time step facilitated by the used of a variational integrator. Furthermore, the computational time required to process an image required for load state estimation had a mean of 0.1329 seconds and a standard deviation of 0.005. When compared to other sensors used on-board for vehicle navigation this "sensor" has a relatively slow update rate.

5.6 Analysis of Results

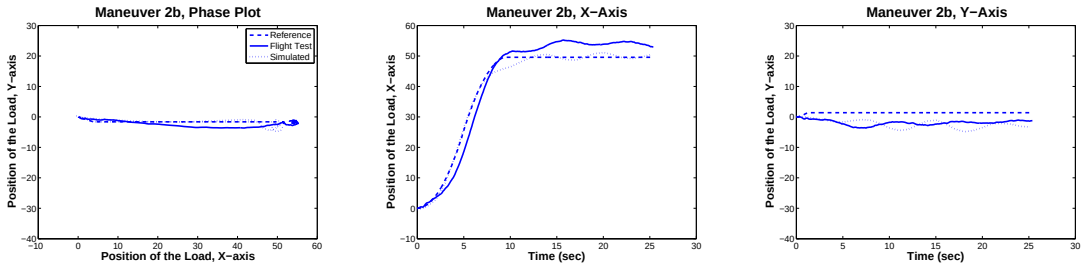
The following observations can be made from the presented data:

- The load tracking error is significantly reduced when the vehicle trajectory is optimized. Furthermore, if Figures 5.10 and 5.18 are compared it can be seen that the trajectory of the load is more oscillatory in the unoptimized case.



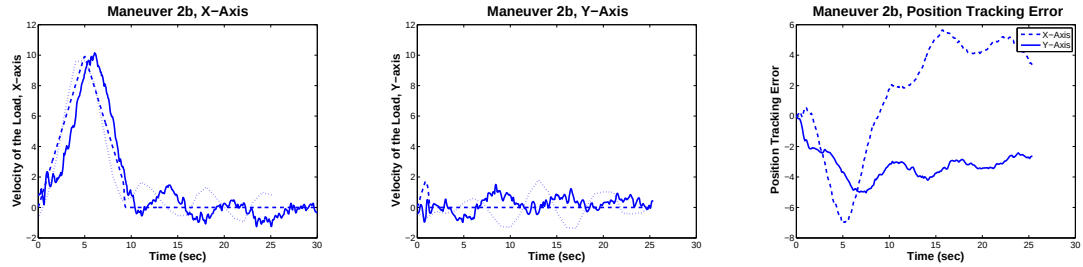
(a) (b) (c)

Figure 5.25: Flight Test-DDP Maneuver 2a (a) and (b): Load velocity as a function of time. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.14), and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the estimated tracking error along the x-axis and y-axis, respectively.



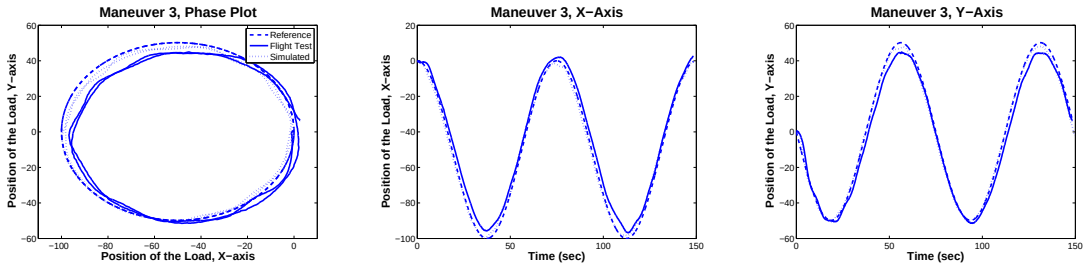
(a) (b) (c)

Figure 5.26: Flight Test-DDP Maneuver 2b (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.15), and reference trajectory, respectively.



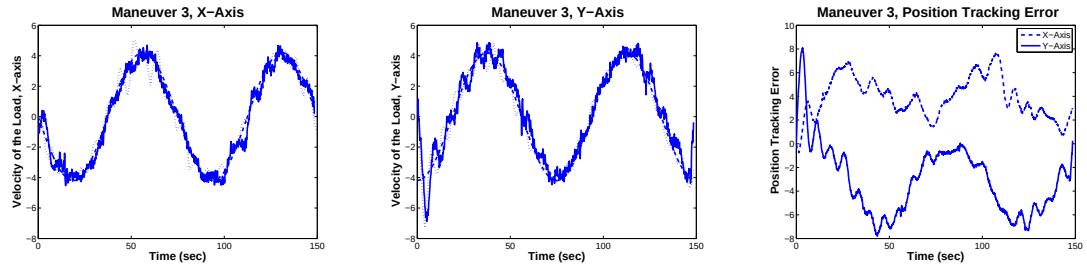
(a) (b) (c)

Figure 5.27: Flight Test-DDP Maneuver 2b (a) and (b): Load velocity as a function of time. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.16), and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the estimated tracking error along the x-axis and y-axis, respectively.



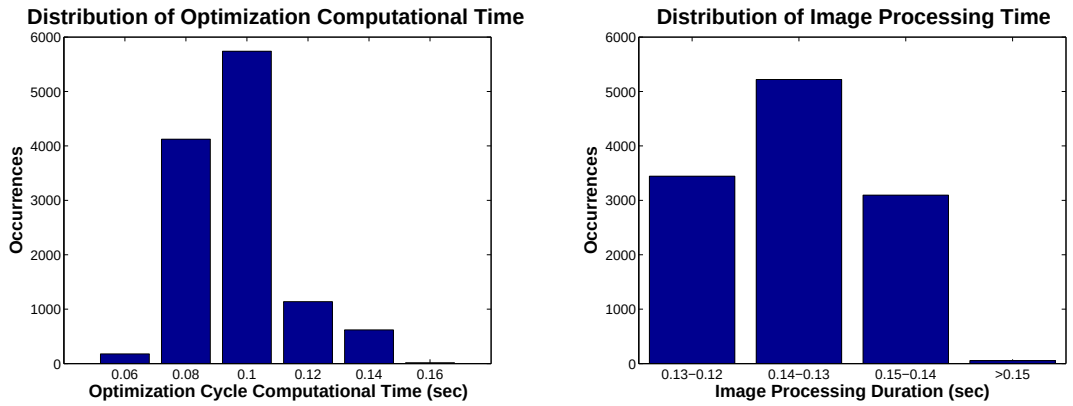
(a) (b) (c)

Figure 5.28: Flight Test-DDP Maneuver 3 (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.17), and reference trajectory, respectively.



(a) (b) (c)

Figure 5.29: Flight Test-DDP Maneuver 3 (a) and (b): Load velocity as a function of time. Solid, dotted, and dashed lines represent the estimated load state, simulated load state (Figure 5.18), and reference trajectory, respectively. (c): Position tracking error as functions of time. Solid and dotted lines represent the estimated tracking error along the x-axis and y-axis, respectively.



(a) (b)

Figure 5.30: Maneuver 1 (a): Histogram of the optimization cycle computational times. The recorded data set had a mean of 0.0965 seconds and a standard deviation of 0.0167. (b): Histogram of image processing computational times. The recorded data set had a mean of 0.1329 seconds and a standard deviation of 0.005.

- The response of the system when the DDP algorithm is utilized is nearly identical to the response when the projection-based optimization method is used. Appendix B shows computed optimized vehicle trajectories using both optimization processes. It is shown that nearly identical solutions are computed.

- The simulation studies accurately predicted data collected during the flight test. It can be expected that further simulation studies will predict future flight test. Appendix B presents additional results obtained from simulation in which more aggressive (larger nominal velocity and acceleration) reference trajectories were considered.

- The results presented in Section 5.4.4 suggest that a limiting factor of the presented framework is the vehicle and suspended load navigation solution. Furthermore, since the vehicle navigation solution is fairly accurate it can be concluded that instrumenting the load will significantly improve the performance of the framework. This is not too surprising since any inaccuracy in the initial conditions given to the optimization process will be manifested in the effectiveness

of the optimized trajectory.

5.7 Conclusion

A trajectory optimization framework for suspended load operations was proposed. The effectiveness of the approach was demonstrated through a series of simulation studies and a flight test. Several directions for future research are outlines in the next chapter.

VI

CONCLUSION AND FUTURE WORK

"...you must go on, I can't go on, I'll go on."
– Samuel Beckett, *The Unnamable*

This dissertation focused on the optimization and propagation of dynamical systems. Using autonomous rotorcraft with suspended load operations as the motivating example the presented optimization framework was demonstrated to be effective and real-time feasible. To the author's knowledge, the presented flight test is the first time a suspended load operation was conducted using iterative optimization techniques in an outdoor/uncontrolled environment. Furthermore, accurate methods for propagating a system's configuration in the presence of uncertainty and stochasticity were also developed.

In Chapter 3, a stochastic variational integrator and its linearization were presented. The stochastic differential dynamical programming and extended Kalman filter algorithms were used to compare the performance of the variational integrator to that of the standard Euler method. It was demonstrated that the algorithms were far less dependent on the discretization time step when the variational integrator was used to propagate system trajectories and linearize system dynamics. It is stressed that these benefits are not limited to the S-DDP and EKF algorithms and similar improvements can be expected in other control and estimation frameworks. The use of variational integrators may enable real-time implementation of nonlinear optimal control algorithms and reduce the computational, power, and sensing requirements of control and estimation technologies.

In Chapter 4, a variational integrator for polynomial chaos expansion coefficients was presented. It was shown that the presented integrator was able to accurately

predict the expectation and variance of an uncertain system. The size of the expansion coefficient vector is a limiting factor of the presented approach and, as a result, further work is needed to mitigate the integrator's complexity.

In Chapter 5, a real-time implementable trajectory optimization framework for autonomous suspended load operations was presented. The computational effort required for the differential dynamic programming (DDP) framework was reduced by the use of variational integrators and a simplified, but representative, system model. Successful simulation studies and a flight test displayed the effectiveness of the proposed framework. Note that this work verifies real-time implementations of trajectory optimization algorithms on complex aerospace systems. Furthermore, the framework can, with necessary changes, be enacted in other robotic guidance or control systems.

The following contributions were presented in this dissertation:

- **Development of a stochastic variational integrator.**
- **Demonstration of benefits when the proposed variational integrator is used in the stochastic differential dynamic programming and extended Kalman filter algorithms.**
- **Derivation of a polynomial chaos variational integrator.**
- **Development of a trajectory optimization framework for autonomous rotorcraft suspended load operations.**
- **Demonstration of the proposed trajectory optimization framework in simulation studies and a flight test.**

In addition, the following research directions have the potential to make significant contributions:

- **Implementation of the stochastic differential dynamic programming algorithm in the presented trajectory optimization framework.** In

Chapter 3, the stochastic differential dynamic programming algorithm was shown to effectively mitigate against stochasticity. Incorporating the S-DDP algorithm into the existing framework may be useful in handling the stochasticity found in the suspended load system. In particular, disturbances due to wind and estimation errors caused by the inherent nature of pixel-based processing.

- **Consideration of input and state constraints.** Due to safety considerations and vehicle limits it is critical to ensure that the optimized trajectory respects given constraints. As more demanding reference trajectories and operations are performed constraints will become even more crucial. Furthermore, state constraints can be used to incorporate obstacle avoidance and multi-vehicle lift in the presented framework.

- **Identification of wind, cable length, and other uncertainties.** On-line methods to identify uncertainty and persistent disturbances can greatly improve the performance and usability of the proposed framework. Existing learning or adaptive based approaches can be implemented separately from the presented framework. In between iterations the vehicle and environment models can be easily updated to incorporate new information. The interaction between the processes (learning and control/guidance) is complex and an area of research in multiple communities.

- **Continuation of flight tests.** More aggressive maneuvers, like those shown in Appendix B, and the research directions described above should be demonstrated. Furthermore, other operational objectives such as precision load drop off need to be performed.

- **System identification using polynomial chaos expansion.** Suppose that an uncertain system is represented with a polynomial chaos expansion. Is it possible to use sensor information in conjunction with this representation to

reduce system uncertainty? That is, given the computed distribution of the trajectory and sensor information gathered in operation can uncertain system parameters be identified? If so, what are the limits in identifying a system in this manner?

- **Optimal control for a polynomial chaos expansion dynamical system.** Integrating a polynomial chaos expansion representation into an iterative optimization algorithm can allow for probabilistic control. That is, the expectation *and* the variance of the optimized trajectory can both be optimized. This allows for risk sensitivity, for example, to be incorporated into the cost function.

- **Development of a variational integrator for an uncertain *and* stochastic system.** The developed stochastic and polynomial chaos variational integrators both consider non-deterministic system dynamics. However, stochasticity and uncertainty are fundamentally different. Consider that stochasticity arises from random events external to the considered modeling "domain" while uncertainty captures what is unknown inside this domain. Nevertheless, in any system both representations are useful and neither one can capture a complete picture of the studied system.

- **Investigate the connection between the projection-based optimization framework and the DDP algorithm.** Note that both methodologies provide the optimal control deviation, but do so in fundamentally different ways. In the DDP algorithm, system dynamics are used to approximate the backward propagating value function. In the projection-based framework, the initial descent direction can be found without explicit consideration of the underlying dynamics. However, the projection operator ensures valid trajectories. It seems probable that a projection operator can be constructed such that the DDP algorithm is recovered within the projection-based optimization framework. Such

an operator may utilize the value function to project trajectories optimally to the desired manifold. The results presented in Appendix B give motivation for this line of research.

APPENDIX A

VARIATIONAL INTEGRATORS OF CONSIDERED DYNAMICAL SYSTEMS

This chapter derives variational integrators (VIs) and their linearizations for a mass-spring-damper system and a 3-link planar manipulator. The reader is referred to Chapters 2 and 3 for additional details concerning the stochastic variational integrator and its linearization.

A.1 Mass-Spring-Damper

The Lagrangian of the considered one-dimensional mass-spring-damper system is given by

$$T(q,\dot{q}) = \frac{1}{2}m\dot{q}^2(t), \tag{A.1}$$

$$V(q) = \int_0^q f_s(\xi(t))d\xi, \tag{A.2}$$

$$L(q,\dot{q}) = T(q,\dot{q}) - V(q), \tag{A.3}$$

where m is the system's mass and $f_s(\cdot)$ gives the force of the spring as a function of mass displacement such that $f_s(q)$ is finite and positive for all $q \neq 0$ and $f_s(0) = 0$. In addition, the system is subject to a damping force and an input such that

$$F_0(q,\dot{q},u) = u(t) - b\dot{q}(t), \tag{A.4}$$

where $b > 0$, and a stochastic disturbance parameterized by (see (3.4))

$$F_1(q,\dot{q},u) = q^2(t). \tag{A.5}$$

The discrete Lagrangian (using the midpoint approximation) is given as

$$L_{\mathrm{d}}(q_k,q_{k+1}) = \frac{1}{2}m\frac{(q_{k+1}-q_k)^2}{\Delta t} - \Big(\int_0^{\frac{q_{k+1}+q_k}{2}} f_s(\xi(t))\ \mathrm{d}\xi\Big)\Delta t, \tag{A.6}$$

and the derivatives necessary to define the variational integrator and its linearization are given as

$$\frac{\partial}{\partial q}L(q,\dot{q}) = -f_s(q(t)), \quad \frac{\partial}{\partial \dot{q}}L(q,\dot{q}) = m\dot{q}, \quad \frac{\partial^2}{\partial q \partial q}L(q,\dot{q}) = -\frac{\partial}{\partial q}f_s(q(t)),$$
$$\frac{\partial^2}{\partial \dot{q}\partial \dot{q}}L(q,\dot{q}) = m, \quad \frac{\partial^2}{\partial \dot{q}\partial q}L(q,\dot{q}) = 0, \quad \text{and} \quad \frac{\partial^2}{\partial q \partial \dot{q}}L(q,\dot{q}) = 0.$$

The left, right, and stochastic discrete forces are given as

$$F^-_{k+1}(q_k, q_{k+1}, u_k) = -\frac{b}{2}(q_{k+1} - q_k) + \frac{1}{2}\Big(\frac{q_{k+1}+q_k}{2}\Big)^3 \Delta t + u_k \Delta t, \tag{A.7}$$
$$F^+_{k+1}(q_k, q_{k+1}, u_k) = -\frac{b}{2}(q_{k+1} - q_k) + \frac{1}{2}\Big(\frac{q_{k+1}+q_k}{2}\Big)^3 \Delta t, \tag{A.8}$$
$$F^s_k(q_{k-1}, q_k, u_k) = q_k^2, \tag{A.9}$$

respectively, and the associated derivatives are computed as

$$\begin{aligned}
D_1F^-_{k+1}(q_k, q_{k+1}, u_k) &= \frac{b}{2} - \frac{3}{4}\Big(\frac{q_{k+1}+q_k}{2}\Big)^2 \Delta t,\\
D_2F^-_{k+1}(q_k, q_{k+1}, u_k) &= -\frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1}+q_k}{2}\Big)^2 \Delta t,\\
D_3F^-_{k+1}(q_k, q_{k+1}, u_k) &= \Delta t,\\
D_1F^+_{k+1}(q_k, q_{k+1}, u_k) &= \frac{b}{2} - \frac{3}{4}\Big(\frac{q_{k+1}+q_k}{2}\Big)^2 \Delta t,\\
D_2F^+_{k+1}(q_k, q_{k+1}, u_k) &= -\frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1}+q_k}{2}\Big)^2 \Delta t,\\
D_3F^+_{k+1}(q_k, q_{k+1}, u_k) &= 0,\\
D_1F^s_k(q_{k-1}, q_k, u_k) &= 0,\\
D_2F^s_k(q_{k-1}, q_k, u_k) &= 2q_k,\\
D_3F^s_k(q_{k-1}, q_k, u_k) &= 0.
\end{aligned}$$

The integrator equation (3.13) for the mass-spring-damper system is computed as

$$\begin{aligned}
f(q_{k+1}) = p_k &- \frac{1}{2}f_s((q_{k+1}+q_k)/2)\Delta t - \frac{m}{\Delta t}(q_{k+1} - q_k) - \frac{b}{2}(q_{k+1} - q_k)\\
&+ \frac{1}{2}\Big(\frac{q_{k+1}+q_k}{2}\Big)^3 \Delta t + u_k \Delta t + q_k^2 \Delta \omega_k,
\end{aligned} \tag{A.10}$$

where

$$p_k = -\frac{1}{2}f_s((q_k + q_{k-1})/2)\Delta t + \frac{m}{\Delta t}(q_k - q_{k-1}) - \frac{b}{2}(q_k - q_{k-1}) + \frac{1}{2}\Big(\frac{q_k + q_{k-1}}{2}\Big)^3\Delta t.$$

The derivative of the integrator equation is calculated as

$$Df(q_{k+1}) = -\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t - \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t. \qquad \text{(A.11)}$$

Therefore, given q_{k-1}, q_k, and u_k the next system configuration q_{k+1} can be found by using (A.10), (A.11), and Algorithm 3. The first-order linearization of the discrete dynamics (3.16) is computed as

$$\begin{aligned}
\frac{\partial q_{k+1}}{\partial q_k} &= -M_{k+1}^{-1}\Big(-\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t + \frac{m}{\Delta t} + \frac{b}{2} - \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t + 2q_k\Delta\omega_k\Big),\\
\frac{\partial q_{k+1}}{\partial p_k} &= -M_{k+1}^{-1},\\
\frac{\partial q_{k+1}}{\partial u_k} &= -M_{k+1}^{-1}\Delta t,\\
\frac{\partial q_{k+1}}{\partial \Delta\omega_k} &= -M_{k+1}^{-1}q_k^2,\\
\frac{\partial p_{k+1}}{\partial q_k} &= \Big(-\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t + \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t\Big)\frac{\partial q_{k+1}}{\partial q_k},\\
&\quad -\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t - \frac{m}{\Delta t} + \frac{b}{2} - \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t,\\
\frac{\partial p_{k+1}}{\partial p_k} &= -\Big(-\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t + \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t\Big)M_{k+1}^{-1},\\
\frac{\partial p_{k+1}}{\partial u_k} &= -\Big(-\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t + \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t\Big)M_{k+1}^{-1}\Delta t,\\
\frac{\partial p_{k+1}}{\partial \Delta\omega_k} &= -\Big(-\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t + \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t\Big)M_{k+1}^{-1}q_k^2,
\end{aligned}$$

where

$$M_{k+1} = -\frac{1}{4}\frac{\partial}{\partial q}f_s((q_{k+1} + q_k)/2)\Delta t - \frac{m}{\Delta t} - \frac{b}{2} + \frac{3}{4}\Big(\frac{q_{k+1} + q_k}{2}\Big)^2\Delta t. \qquad \text{(A.12)}$$

As an example, consider the case where there is no stochastic disturbance, $u(t) = 0$, $b = 0$, $m = 1$, and $f_s(q(t)) = 4q(t)$. The linearization of the system for this case is given as

$$\begin{bmatrix} \delta q_{k+1} \\ \delta p_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{1-\Delta t^2}{1+\Delta t^2} & \frac{\Delta t}{1+\Delta t^2} \\ \frac{-4\Delta t}{1+\Delta t^2} & \frac{1-\Delta t^2}{1+\Delta t^2} \end{bmatrix}\begin{bmatrix} \delta q_k \\ \delta p_k \end{bmatrix}. \qquad \text{(A.13)}$$

If $\Delta t << 1$ then

$$\begin{bmatrix} \delta q_{k+1} \\ \delta p_{k+1} \end{bmatrix} \approx \begin{bmatrix} 1 & \Delta t \\ -4\Delta t & 1 \end{bmatrix} \begin{bmatrix} \delta q_k \\ \delta p_k \end{bmatrix}. \tag{A.14}$$

For comparison purposes consider the explicit integrator equation obtained by using the Euler method

$$x_{k+1} = (\mathbf{I} + \frac{\partial}{\partial x_k} f(x_k)\Delta t)x_k. \tag{A.15}$$

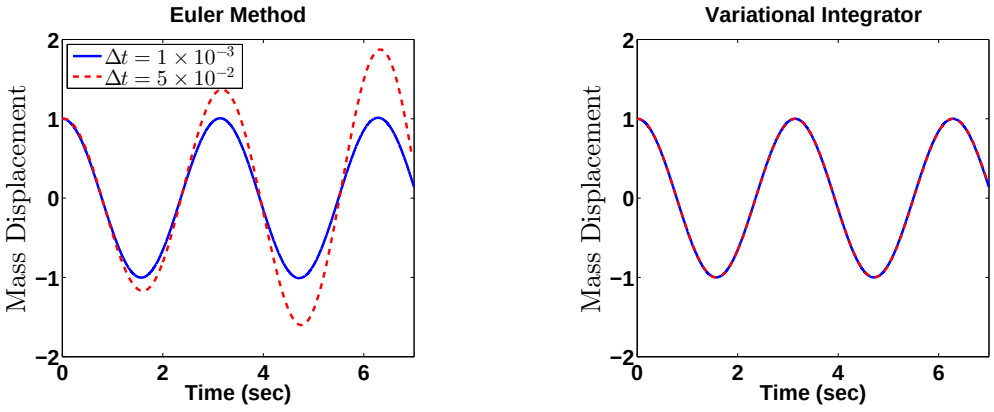
where $x_k = [q_k, \dot{q}_k]^{\mathrm{T}}$, $\dot{q}_k = mp_k$ (for this case and not in general), and

$$f(x) = \begin{bmatrix} \delta \dot{q}_k \\ -4q_k \end{bmatrix}. \tag{A.16}$$

Note that if Δt is sufficiently small the linearizations approximate each other. However as shown in Chapters 3 and 4, the variational integrator is far less dependent on Δt than the Euler method. To further highlight this point, Figure A.1 shows how the two methods propagate $q(t)$ for the case considered above. It is clear the the variational integrator is able to accurately propagate the system state for both time steps while the Euler method quickly becomes unreliable when the larger time step is used.

A.2 3-Link Manipulator

This section considers a dynamical system representing a human finger (3-link planar manipulator) (see [71] for system model and parameters). As shown in Figure A.2 the dynamical system is described by three coordinates given by the relative angles between adjacent links, $q(t) = [\theta_1(t), \theta_2(t), \theta_3(t)]$, and three control inputs, $u(t) = [u_1(t), u_2(t), u_3(t)]$. The mass of the links is assumed to be concentrated at the end of each link (3 linked pendulums). The potential energy of the system arises from the



(a) (b)

Figure A.1: (a): Propagation of mass displacement, $q(t)$, with initial condition $q(t_0) = 1$ and $\dot{q}(t_0) = 0$ subjected to input $u(t) = 0$. Solid lines indicate a step size of $\Delta t = 1 \times 10^{-3}$ while dotted lines indicate $\Delta t = 5 \times 10^{-2}$.

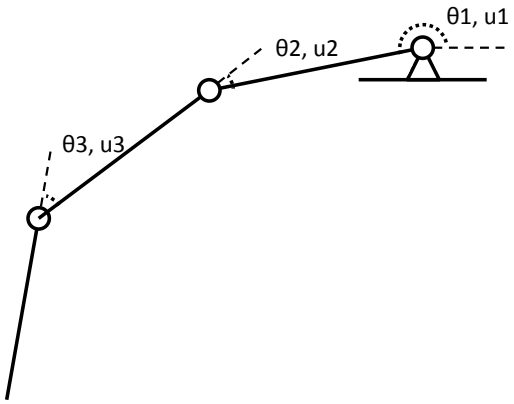


Figure A.2: Diagram of the studied 3-link planar manipulator.

gravitational field present and is computed as

$$V(q) = -m_1 g l_1 \sin(\theta_1) - m_2 g\big(l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)\big) \\ - m_3 g\big(l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)\big). \tag{A.17}$$

The kinetic energy of the system is given as

$$T(q, \dot{q}) = T_1(q, \dot{q}) + T_2(q, \dot{q}) + T_3(q, \dot{q}), \tag{A.18}$$

$$T_1(q, \dot{q}) = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2, \tag{A.19}$$

$$T_2(q, \dot{q}) = \frac{1}{2} m_2 \big(l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_1 l_2 \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2)\big), \tag{A.20}$$

$$T_3(q, \dot{q}) = \frac{1}{2} m_3 \big(l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_3^2 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + l_1 l_2 \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \\ + l_1 l_3 \cos(\theta_2 + \theta_3) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 l_3 \cos(\theta_3) (\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\big). \tag{A.21}$$

The system is subject to damping forces and inputs such that

$$F_0(q, \dot{q}, u) = [u_1(t) - b\dot{\theta}_1(t), u_2(t) - b\dot{\theta}_2(t), u_3(t) - b\dot{\theta}_3(t)]^{\mathrm{T}} \tag{A.22}$$

where $b > 0$, and stochastic disturbances parameterized by (see (3.4))

$$F_1^s = [C_1(\theta_1(t) - \theta_1(t_0))^2, 0, 0]^{\mathrm{T}}, \quad F_2^s = [C_2 u_1^2(t), 0, 0]^{\mathrm{T}}, \quad F_3^s = [C_3, 0, 0]^{\mathrm{T}},$$

$$F_4^s = [0, C_4(\theta_2(t) - \theta_2(t_0))^2, 0]^{\mathrm{T}}, \quad F_5^s = [0, C_5 u_2^2(t), 0]^{\mathrm{T}}, \quad F_6^s = [0, C_6, 0]^{\mathrm{T}},$$

$$F_7^s = [0, 0, C_7(\theta_3(t) - \theta_3(t_0))^2]^{\mathrm{T}}, \quad F_8^s = [0, 0, C_8 u_3^2(t)]^{\mathrm{T}}, \quad F_9^s = [0, 0, C_9]^{\mathrm{T}},$$

$C_i > 0$ for all i. The partial derivatives necessary to define the variational integrator and its linearization are computed as

$$
\begin{aligned}
[\frac{\partial}{\partial q}L(q,\dot{q})]_1 &= m_1 g l_1 \cos(\theta_1) + m_2 g\big(l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2)\big) \\
&\quad + m_3 g\big(l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) + l_3\cos(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial}{\partial q}L(q,\dot{q})]_2 &= m_2 g l_2 \cos(\theta_1+\theta_2) + m_3 g\big(l_2\cos(\theta_1+\theta_2) + l_3\cos(\theta_1+\theta_2+\theta_3)\big) \\
&\quad - m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2) - m_3 l_1 l_2 \sin(\theta_2)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2) \\
&\quad - m_3 l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3), \\
[\frac{\partial}{\partial q}L(q,\dot{q})]_3 &= m_3 g l_3 \cos(\theta_1+\theta_2+\theta_3) - m_3 l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) \\
&\quad - m_3 l_2 l_3 \sin(\theta_3)(\dot{\theta}_1+\dot{\theta}_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3), \\
[\frac{\partial}{\partial \dot{q}}L(q,\dot{q})]_1 &= m_1 l_1^2\dot{\theta}_1 + m_2\big(l_1^2\dot{\theta}_1 + l_2^2(\dot{\theta}_1+\dot{\theta}_2) + l_1 l_2\cos(\theta_2)(2\dot{\theta}_1+\dot{\theta}_2)\big) \\
&\quad + m_3\big(l_1^2\dot{\theta}_1 + l_2^2(\dot{\theta}_1+\dot{\theta}_2) + l_3^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + l_1 l_2\cos(\theta_2)(2\dot{\theta}_1+\dot{\theta}_2) \\
&\quad\quad + l_1 l_3\cos(\theta_2+\theta_3)(2\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + l_2 l_3\cos(\theta_3)(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)\big), \\
[\frac{\partial}{\partial \dot{q}}L(q,\dot{q})]_2 &= m_2\big(l_2^2(\dot{\theta}_1+\dot{\theta}_2) + l_1 l_2\cos(\theta_2)\dot{\theta}_1\big) \\
&\quad + m_3\big(l_2^2(\dot{\theta}_1+\dot{\theta}_2) + l_3^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + l_1 l_2\cos(\theta_2)\dot{\theta}_1 \\
&\quad\quad + l_1 l_3\cos(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3\cos(\theta_3)(2\dot{\theta}_1+2\dot{\theta}_2+\dot{\theta}_3)\big), \\
[\frac{\partial}{\partial \dot{q}}L(q,\dot{q})]_3 &= m_3\big(l_3^2(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + l_1 l_3\cos(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3\cos(\theta_3)(\dot{\theta}_1+\dot{\theta}_2)\big) \\
,[\frac{\partial^2}{\partial q\partial q}L(q,\dot{q})]_{1,1} &= -m_1 g l_1\sin(\theta_1) - m_2 g\big(l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2)\big) \\
&\quad - m_3 g\big(l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial q\partial q}L(q,\dot{q})]_{1,2} &= -m_2 g l_2\sin(\theta_1+\theta_2) - m_3 g\big(l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial q\partial q}L(q,\dot{q})]_{1,3} &= -m_3 g l_3\sin(\theta_1+\theta_2+\theta_3), \\
[\frac{\partial^2}{\partial q\partial q}L(q,\dot{q})]_{2,1} &= -m_2 g l_2\sin(\theta_1+\theta_2) - m_3 g\big(l_2\sin(\theta_1+\theta_2) + l_3\sin(\theta_1+\theta_2+\theta_3)\big),
\end{aligned}
$$

$$
\begin{aligned}
[\frac{\partial^2}{\partial q \partial q} L(q,\dot{q})]_{2,2} &= -m_2\big(l_1 l_2 \cos(\theta_2)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2) + g l_2 \sin(\theta_1+\theta_2)\big) \\
&\quad - m_3\big(l_1 l_2 \cos(\theta_2)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2) + l_1 l_3 \cos(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) \\
&\quad + g l_2 \sin(\theta_1+\theta_2) + g l_3 \sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial q \partial q} L(q,\dot{q})]_{2,3} &= -m_3\big(l_1 l_3 \cos(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + g l_3 \sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial q \partial q} L(q,\dot{q})]_{3,1} &= -m_3 g l_3 \sin(\theta_1+\theta_2+\theta_3), \\
[\frac{\partial^2}{\partial q \partial q} L(q,\dot{q})]_{3,2} &= -m_3\big(l_1 l_3 \cos(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + g l_3 \sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial q \partial q} L(q,\dot{q})]_{3,3} &= -m_3\big(l_1 l_3 \cos(\theta_2+\theta_3)\dot{\theta}_1(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) \\
&\quad + l_2 l_3 \cos(\theta_3)(\dot{\theta}_1+\dot{\theta}_2)(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) + g l_3 \sin(\theta_1+\theta_2+\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{1,1} &= m_1 l_1^2 + m_2\big(l_1^2 + l_2^2 + 2 l_1 l_2 \cos(\theta_2)\big) \\
&\quad + m_3\big(l_1^2 + l_2^2 + l_3^2 + l_1 l_2 \cos(\theta_2)2 + l_1 l_3 \cos(\theta_2+\theta_3)2 + 2 l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{1,2} &= m_2\big(l_2^2 + l_1 l_2 \cos(\theta_2)\big) \\
&\quad + m_3\big(l_2^2 + l_3^2 + l_1 l_2 \cos(\theta_2) + l_1 l_3 \cos(\theta_2+\theta_3) + 2 l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{1,3} &= m_3\big(l_3^2 + l_1 l_3 \cos(\theta_2+\theta_3) + l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{2,1} &= m_2\big(l_2^2 + l_1 l_2 \cos(\theta_2)\big) \\
&\quad + m_3\big(l_2^2 + l_3^2 + l_1 l_2 \cos(\theta_2) + l_1 l_3 \cos(\theta_2+\theta_3) + 2 l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{2,2} &= m_2 l_2^2 + m_3\big(l_2^2 + l_3^2 + 2 l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{2,3} &= m_3\big(l_3^2 + l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{3,1} &= m_3\big(l_3^2 + l_1 l_3 \cos(\theta_2+\theta_3) + l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{3,2} &= m_3\big(l_3^2 + l_2 l_3 \cos(\theta_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial \dot{q}} L(q,\dot{q})]_{3,3} &= m_3 l_3^2, \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{1,1} &= [\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{2,1} = [\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{3,1} = 0,
\end{aligned}
$$

$$
\begin{aligned}
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{1,2} &= -m_2 l_1 l_2 \sin(\theta_2)(2\dot{\theta}_1 + \dot{\theta}_2) \\
&\quad - m_3\big(l_1 l_2 \sin(\theta_2)(2\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_3 \sin(\theta_2+\theta_3)(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{1,3} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 l_3 \sin(\theta_3)(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{2,2} &= -m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1 - m_3\big(l_1 l_2 \sin(\theta_2)\dot{\theta}_1 + l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1\big), \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{2,3} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3 \sin(\theta_3)(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{3,2} &= -m_3 l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1, \\
[\frac{\partial^2}{\partial q \partial \dot{q}} L(q,\dot{q})]_{3,3} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3 \sin(\theta_3)(\dot{\theta}_1 + \dot{\theta}_2)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{1,1} &= [\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{1,2} = [\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{1,3} = 0, \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{2,1} &= -m_2 l_1 l_2 \sin(\theta_2)(2\dot{\theta}_1 + \dot{\theta}_2) \\
&\quad - m_3\big(l_1 l_2 \sin(\theta_2)(2\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_3 \sin(\theta_2+\theta_3)(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{2,2} &= -m_2 l_1 l_2 \sin(\theta_2)\dot{\theta}_1 - m_3\big(l_1 l_2 \sin(\theta_2)\dot{\theta}_1 + l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{2,3} &= -m_3 l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1, \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{3,1} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 l_3 \sin(\theta_3)(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{3,2} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3 \sin(\theta_3)(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)\big), \\
[\frac{\partial^2}{\partial \dot{q} \partial q} L(q,\dot{q})]_{3,3} &= -m_3\big(l_1 l_3 \sin(\theta_2+\theta_3)\dot{\theta}_1 + l_2 l_3 \sin(\theta_3)(\dot{\theta}_1 + \dot{\theta}_2)\big),
\end{aligned}
$$

The left, right, and stochastic discrete forces are given as

$$F^-_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} -\frac{b}{2}(\theta_{1,k+1} - \theta_{1,k}) + \frac{C_1^2}{2}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^3 \Delta t + u_{1,k}\Delta t \\ -\frac{b}{2}(\theta_{2,k+1} - \theta_{2,k}) + \frac{C_4^2}{2}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^3 \Delta t + u_{2,k}\Delta t \\ -\frac{b}{2}(\theta_{3,k+1} - \theta_{3,k}) + \frac{C_7^2}{2}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^3 \Delta t + u_{3,k}\Delta t \end{bmatrix},$$

$$F^+_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} -\frac{b}{2}(\theta_{1,k+1} - \theta_{1,k}) + \frac{C_1^2}{2}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^3 \Delta t \\ -\frac{b}{2}(\theta_{2,k+1} - \theta_{2,k}) + \frac{C_4^2}{2}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^3 \Delta t \\ -\frac{b}{2}(\theta_{3,k+1} - \theta_{3,k}) + \frac{C_7^2}{2}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^3 \Delta t \end{bmatrix},$$

$$F^s_k(q_{k-1}, q_k, u_k) = \begin{bmatrix} C_1(\theta_{1,k} - \theta_{1,0})^2 & C_2 u_{1,k}^2 & C_3 & \dots & \dots \\ \dots & C_4(\theta_{2,k} - \theta_{2,0})^2 & C_5 u_{2,k}^2 & C_6 & \dots \\ \dots & \dots & C_7(\theta_{3,k} - \theta_{3,0})^2 & C_8 u_{3,k}^2 & C_9 \end{bmatrix},$$

respectively, and the associated derivatives are computed as

$$D_1 F^-_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} \frac{b}{2} + \frac{3C_1^2}{4}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^2 \Delta t \\ \frac{b}{2} + \frac{3C_4^2}{4}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^2 \Delta t \\ \frac{b}{2} + \frac{3C_7^2}{4}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^2 \Delta t \end{bmatrix},$$

$$D_2 F^-_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} -\frac{b}{2} + \frac{3C_1^2}{4}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^2 \Delta t \\ -\frac{b}{2} + \frac{3C_4^2}{4}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^2 \Delta t \\ -\frac{b}{2} + \frac{3C_7^2}{4}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^2 \Delta t \end{bmatrix},$$

$$D_3 F^+_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} \Delta t \\ \Delta t \\ \Delta t \end{bmatrix},$$

$$D_1 F^+_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} \frac{b}{2} + \frac{3C_1^2}{4}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^2 \Delta t \\ \frac{b}{2} + \frac{3C_4^2}{4}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^2 \Delta t \\ \frac{b}{2} + \frac{3C_7^2}{4}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^2 \Delta t \end{bmatrix},$$

$$D_2F^+_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} -\frac{b}{2} + \frac{3C_1^2}{4}\left(\frac{\theta_{1,k+1}+\theta_{1,k}}{2} - \theta_{1,0}\right)^2 \Delta t \\ -\frac{b}{2} + \frac{3C_4^2}{4}\left(\frac{\theta_{2,k+1}+\theta_{2,k}}{2} - \theta_{2,0}\right)^2 \Delta t \\ -\frac{b}{2} + \frac{3C_7^2}{4}\left(\frac{\theta_{3,k+1}+\theta_{3,k}}{2} - \theta_{3,0}\right)^2 \Delta t \end{bmatrix},$$

$$D_3F^-_{k+1}(q_k, q_{k+1}, u_k) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

$D_1F^s_k(q_{k-1}, q_k, u_k)$, $D_2F^s_k(q_{k-1}, q_k, u_k)$, and $D_3F^s_k(q_{k-1}, q_k, u_k)$ can be calculated in a similar manner. The integrator equation (3.13), its derivative (3.15), and first-order linearization (3.16) can now be computed from the quantities presented above.

APPENDIX B

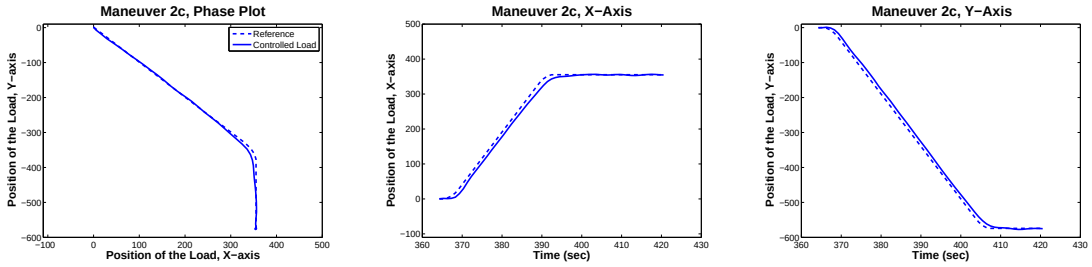
ADDITIONAL SIMULATION RESULTS

The chapter provides additional results from simulation studies. First, results when the reference trajectories are made more aggressive (i.e. larger maximum velocities and accelerations) are shown. Next, a possible method to incorporate input constraints into the proposed framework is discussed and preliminary simulation results are shown. Finally, outputs from the two optimization techniques implemented (differential dynamic programming and projection-based optimization) are compared.

B.1 Aggressive Maneuvers

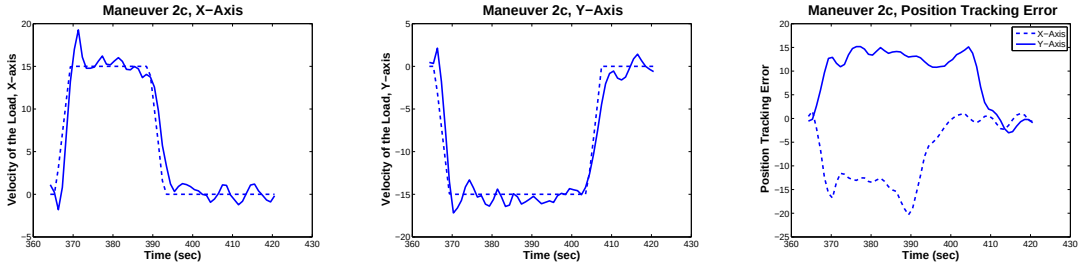
Three new maneuvers are introduced for the purposes of this section. Maneuver 2c reference trajectory was generated using a maximum velocity and acceleration of 15 feet/sec and 4 feet/sec^2, respectively, and traveled about 355 feet and -575 feet along the x-axis and y-axis, respectively. Maneuver 2d reference trajectory was generated using a maximum velocity and acceleration of 25 feet/sec and 5 feet/sec^2, respectively, and traveled about -347 feet and 246 foot along the x-axis and y-axis, respectively. Maneuver 3a reference trajectory was constructed to inscribed a circle with a 100 foot radius every 75 seconds such that its tangential velocity is 8.38 feet/sec.

Figures B.1, B.2, B.3, and B.4 show the trajectory histories when Maneuvers 2c and 2d were performed. Note that in both responses the general shape of the resultant trajectories were very similar in both position and velocity to the given reference. However as seen in Chapter 5, the responses lag behind the reference trajectory (see, for example, Figure B.3.b). Limiting the reference trajectory's jerk and snap should alleviate this problem. Currently, the reference trajectory has infinite jerk and this



(a) (b) (c)

Figure B.1: Simulation-DDP Maneuver 2c (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dashed lines represent the load state and reference trajectory, respectively.

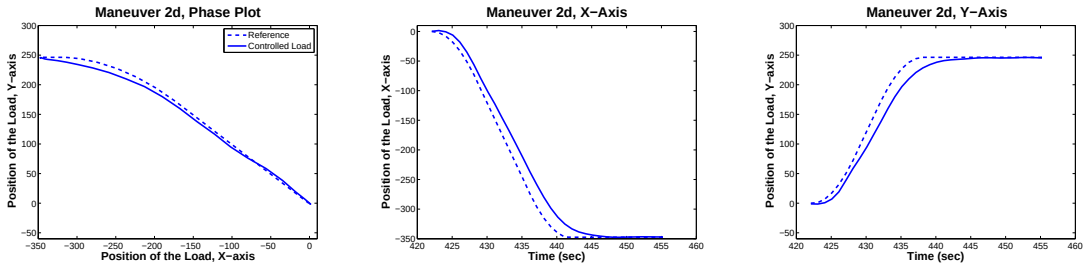


(a) (b) (c)

Figure B.2: Simulation-DDP Maneuver 2c (a) and (b): Load velocity as a function of time. Solid and dashed lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Dashed and solid lines represent the tracking error along the x-axis and y-axis, respectively.

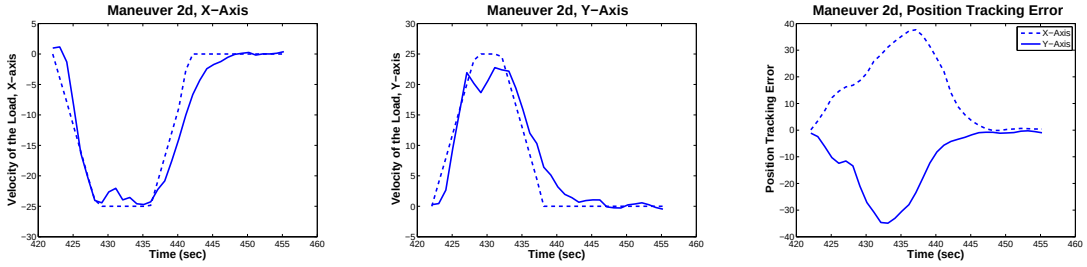
results in a rapid increase of the tracking error at the beginning of the maneuver (see, for example, Figures 5.18.c and B.2.c). Note that the vehicle (if initially at hover) cannot instantaneously accelerate the load forward. Furthermore, the acceleration of the vehicle is directly related to its attitude and cannot be changed instantaneously. Therefore, placing limits on the jerk and snap of the reference trajectory corresponding to the maneuverability of the vehicle is the next logical step forward.

Figures B.5 and B.6 show the trajectory history when Maneuvers 3a was performed. After an initial large tracking error (see paragraph above) the vehicle was able to maintain the load within 5 feet of the reference trajectory.



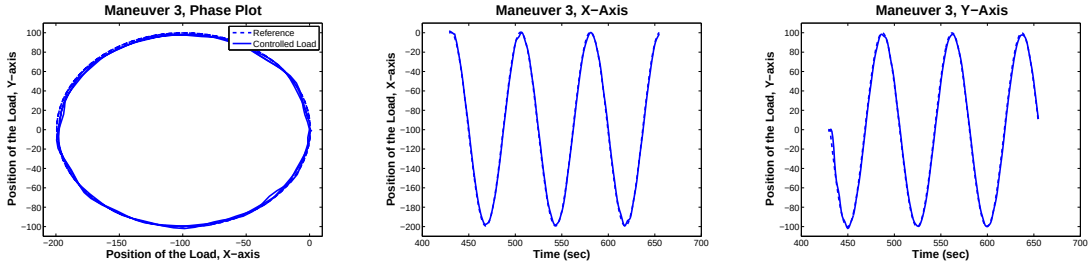
(a) (b) (c)

Figure B.3: Simulation-DDP Maneuver 2d (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dashed lines represent the load state and reference trajectory, respectively.



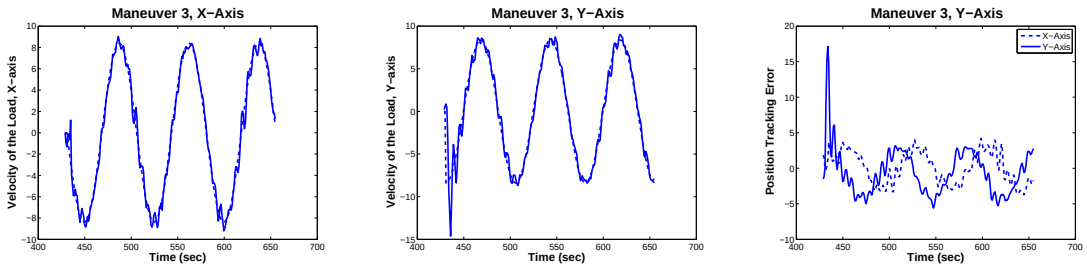
(a) (b) (c)

Figure B.4: Simulation-DDP Maneuver 2d (a) and (b): Load velocity as a function of time. Solid and dashed lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Dashed and solid lines represent the tracking error along the x-axis and y-axis, respectively.



(a) (b) (c)

Figure B.5: Simulation-DDP Maneuver 3a (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dashed lines represent the load state and reference trajectory, respectively.



(a) (b) (c)

Figure B.6: Simulation-DDP Maneuver 3a (a) and (b): Load velocity as a function of time. Solid and dashed lines represent the load state and reference trajectory, respectively. (c): Position tracking error as functions of time. Dashed and solid lines represent the tracking error along the x-axis and y-axis, respectively.

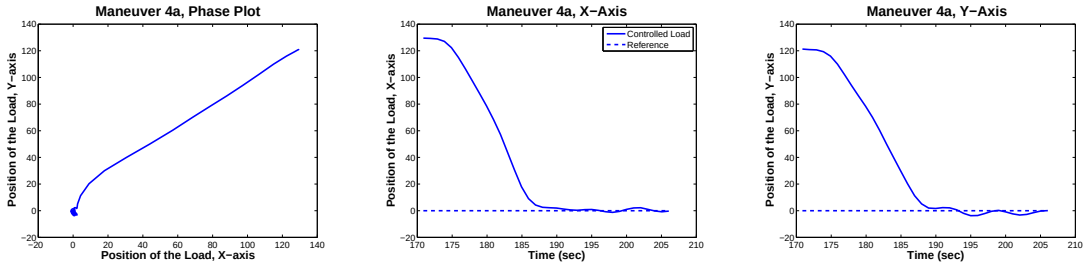
B.2 Input Constraints

In this section we investigate the effectiveness of imposing given input constraints with "clamping". Specifically, the Armijo line search is altered in Algorithm 1 such that the proposed input respects given constraints:

$$u_{\mathrm{p}} = \max(\min(u + \beta^j \delta u^\star, \bar{u}), \underline{u}) \tag{B.1}$$

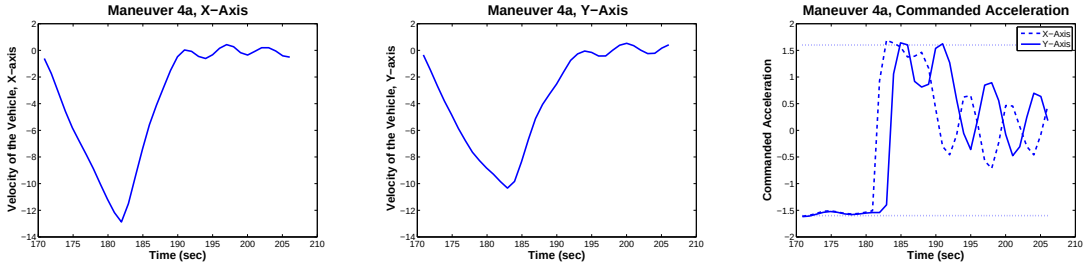
where $\bar{u}$ and $\underline{u}$ are the upper and lower input constraints, respectively. Note that the proposed input u_{p}, and not the optimal descent direction $u^\star$, is altered. Additionally, reference trajectories were not used in this investigation. Instead, step commands were given. That is, the desired position of the load was translated instantaneously to a new position.

Figures B.7, B.8, B.9, and B.10 show the trajectory histories when Maneuvers 4a and 4b were performed. The vehicle was able to place the load at the desired positions. As expected, the velocity of the vehicle generally follows "a rapid increase then a rapid decrease" shape. The input constraints were generally respected. Note that the initial acceleration of the vehicle need not respect the imposed constraints. As a result, the initial system configuration vector may cause a slight violation of the constraints since the commanded position, velocity, and acceleration of the vehicle



(a) (b) (c)

Figure B.7: Simulation-DDP Maneuver 4a (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dashed lines represent the load state and reference trajectory, respectively.



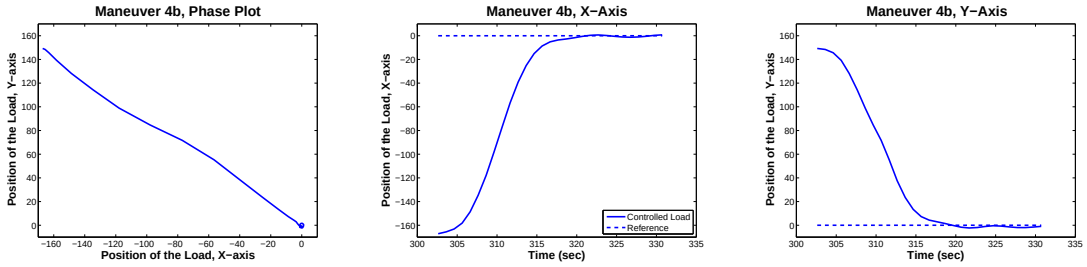
(a) (b) (c)

Figure B.8: Simulation-DDP Maneuver 4a (a) and (b): Velocity of the vehicle as a function of time along the x-axis and y-axis, respectively. (c): Commanded acceleration of the vehicle. Dashed and solid lines represent the commanded acceleration along the x-axis and y-axis, respectively, and the dotted lines indicate the imposed input constraints.

are found through interpolation. It should be noted that state constraints cannot be handled in this manner.

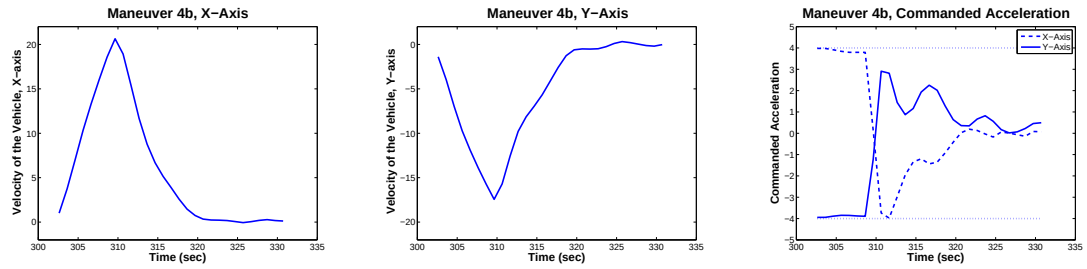
B.3 Comparing Optimization Techniques

In this section outputs from the two optimization techniques implemented (differential dynamic programming and projection-based optimization) are compared. The initial conditions were set such that the vehicle was at hover, there is no swing angle, and the load has a small forward velocity (1.5 feet/sec). Figures B.11 ,B.12, and B.13 show the optimized trajectories (at $t = 0$) when Maneuvers 1, 2e, and 3 were selected,



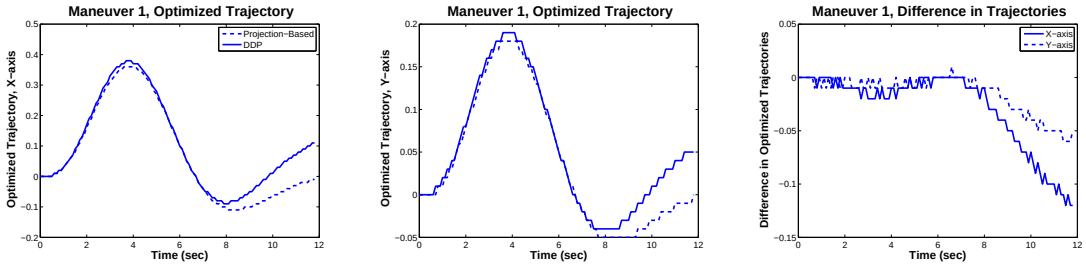
(a) (b) (c)

Figure B.9: Simulation-DDP Maneuver 4b (a): Phase plot of the maneuver, (b) and (c): Load position as a function of time along the x-axis and y-axis, respectively. Solid and dashed lines represent the load state and reference trajectory, respectively.



(a) (b) (c)

Figure B.10: Simulation-DDP Maneuver 4b (a) and (b): Velocity of the vehicle as a function of time along the x-axis and y-axis, respectively. (c): Commanded acceleration of the vehicle. Dashed and solid lines represent the commanded acceleration along the x-axis and y-axis, respectively, and the dotted lines indicate the imposed input constraints.



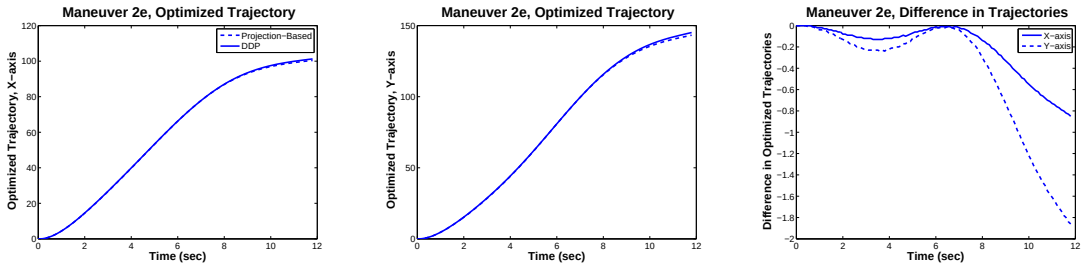
(a) (b) (c)

Figure B.11: Simulation-Maneuver 1 (a) and (b): Optimized trajectory as a function of time along the x-axis and y-axis, respectively. The dashed and solid lines correspond to the Projection-Based and DDP algorithms, respectively. (c): The difference between the optimized trajectories as a function of time. The solid and dashed lines correspond to the difference along the x-axis and y-axis, respectively.

respectively. Maneuver 2e reference trajectory was generated using a maximum velocity and acceleration of 10 feet/sec and 2 feet/sec^2, respectively, and traveled about 100 feet and 150 feet along the x-axis and y-axis, respectively. (see Section 5.4 for a description of Maneuvers 1 and 3). The DDP algorithm produced a lower cost for each maneuver:

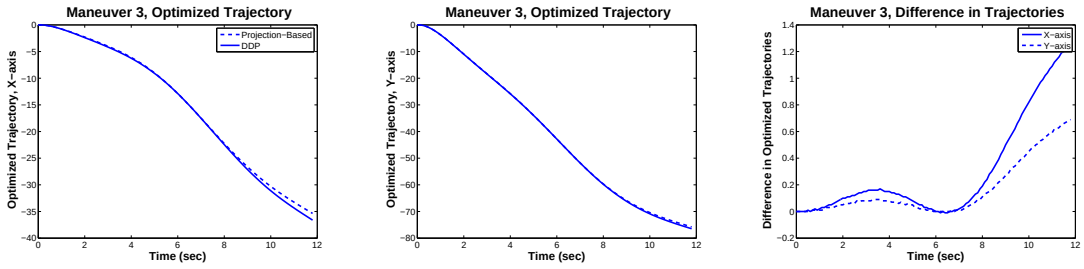
DDP	Man. 1 : 9.72, Man. 2e : 21353.33, Man. 3 : 5528.76
Projection-Based	Man. 1 : 9.81, Man. 2e : 21462.01, Man. 3 : 5697.67

However, the differences between the costs were all relatively small. Furthermore, the optimized trajectories were very similar (particularly before $t = 7$). The minor differences can be a result of how the projection operator was selected. It is reasonable to suspect that changing G, G_f, and F will produce different, albeit very similar, results.



(a) (b) (c)

Figure B.12: Simulation-Maneuver 2e (a) and (b): Optimized trajectory as a function of time along the x-axis and y-axis, respectively. The dashed and solid lines correspond to the Projection-Based and DDP algorithms, respectively. (c): The difference between the optimized trajectories as a function of time. The solid and dashed lines correspond to the difference along the x-axis and y-axis, respectively.



(a) (b) (c)

Figure B.13: Simulation-Maneuver 3 (a) and (b): Optimized trajectory as a function of time along the x-axis and y-axis, respectively. The dashed and solid lines correspond to the Projection-Based and DDP algorithms, respectively. (c): The difference between the optimized trajectories as a function of time. The solid and dashed lines correspond to the difference along the x-axis and y-axis, respectively.

REFERENCES

[1] Ascher, U. M. and Petzold, L. R., *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

[2] Asseo, S. J. and Whitbeck, R. F., "Control requirements for sling-load stabilization in heavy lift helicopters," *Journal of the American Helicopter Society*, vol. 18, no. 3, pp. 23–31, 1973.

[3] Banavar, R., Kazi, F., Ortega, R., and Manjarekar, N. S., "The ida-pbc methodology applied to a gantry crane," in *Proceedings of the Mathematical Theory of Networks and Systems*, 2006.

[4] Bernard, M., Kondak, K., and Hommel, G., "A slung load transportation system based on small size helicopters," in *Autonomous Systems - Self-Organization, Management, and Control* (Mahr, B. and Huanye, S., eds.), 2008.

[5] Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., and Bertsekas, D. P., *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, 1995.

[6] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[7] Bisgaard, M., la Cour-Harbo, A., and Bendtsen, J., "Full state estimation for helicopter slung load system," in *AIAA Guidance, Navigation and Control Conference*, 2007.

[8] Bisgaard, M., la Cour-Harbo, A., and Bendtsen, J. D., "Input shaping for helicopter slung load swing reduction," in *AIAA Guidance, Navigation and Control Conference*, 2008.

[9] Bisgaard, M., la Cour-Harbo, A., and Bendtsen, J. D., "Swing damping for helicopter slung load systems using delayed feedback," in *AIAA Guidance, Navigation and Control Conference*, 2009.

[10] Bisgaard, M., la Cour-Harbo, A., Johnson, E. N., and Bendtsen, J. D., "Vision aided state estimator for helicopter slung load system," *Automatic Control in Aerospace*, vol. 17, no. 1, pp. 425–430, 2007.

[11] Bou-Rabee, N. and Owhadi, H., "Stochastic variational integrators," *IMA Journal of Numerical Analysis*, vol. 29, no. 2, pp. 421–443, 2009.

[12] Brenner, H., "Helicopter sling load pendulum damping," in *European Rotorcraft Forum*, 2009.

[13] Brenner, H., "Helicopter sling load positioning," in *European Rotorcraft Forum*, 2009.

[14] Bryson, E. and Ho, Y.-C., *Applied Optimal Control*. Hemisphere Publishing Corporation, 1975.

[15] Camacho, E. F. and Alba, C. B., *Model predictive control*. Springer, 2013.

[16] Cameron, R. H. and Martin, W. T., "The orthogonal development of nonlinear functionals in series of fourier-hermite functionals," *Annals of Mathematics*, pp. 385–392, 1947.

[17] Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A., "Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013.

[18] Colucci, F., "Hummingbird hunts for a home," *Vertiflite*, 2008.

[19] Colucci, F., "Evolving autonomy unmanned k-max points the way," *Vertiflite*, 2013.

[20] Conway, B. A., "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012.

[21] de Crousaz, C., Farshidian, F., and Buchli, J., "Aggressive optimal control for agile flight with a slung load," in *IROS Workshop on Machine Learning in Planning and Control of Robot Motion*, 2014.

[22] de Crousaz, C., Farshidian, F., Neunert, M., and Buchli, J., "Unified motion control for dynamic quadrotor maneuvers demonstrated on slung load and rotor failure tasks," in *IEEE International Conference on Robotics and Automation*, 2015.

[23] Dittrich, J. S. and Johnson, E. N., "Multi-sensor navigation system for an autonomous helicopter," in *Digital Avionics Systems Conference*, 2002.

[24] Dukes, T. A., "Maneuvering heavy sling loads near hover part i: Damping the pendulous motion," *Journal of the American Helicopter Society*, vol. 18, no. 2, pp. 2–11, 1973.

[25] Dunbar, W. B., Milam, M. B., Franz, R., and Murray, R. M., "Model predictive control of a thrust-vectored flight control experiment," in *IFAC World Congress*, 2002.

[26] Dutta, P. and Bhattacharya, R., "Nonlinear estimation with polynomial chaos and higher order moment updates," in *American Control Conference*, 2010.

[27] Ernst, O. G., Mugler, A., Starkloff, H.-J., and Ullmann, E., "On the convergence of generalized polynomial chaos expansions," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 46, no. 02, pp. 317–339, 2012.

[28] Faust, A., Malone, N., and Tapia, L., "Preference-balancing motion planning under stochastic disturbances," in *IEEE International Conference on Robotics and Automation*, 2015.

[29] Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L., "Automated aerial suspended cargo delivery through reinforcement learning," *Artificial Intelligence*, 2014.

[30] Fetecau, R. C., Marsden, J. E., Ortiz, M., and West, M., "Nonsmooth lagrangian mechanics and variational collision integrators," *SIAM Journal on Applied Dynamical Systems*, vol. 2, no. 3, pp. 381–416, 2003.

[31] Fisher, J. and Bhattacharya, R., "On stochastic lqr design and polynomial chaos," in *American Control Conference*, 2008.

[32] Fisher, J. and Bhattacharya, R., "Stability analysis of stochastic systems using polynomial chaos," in *American Control Conference*, 2008.

[33] Fisher, J. and Bhattacharya, R., "Linear quadratic regulation of systems with stochastic parameter uncertainties," *Automatica*, vol. 45, no. 12, pp. 2831–2841, 2009.

[34] Flasskamp, K. and Murphey, T. D., "Variational integrator in linear optimal control and filtering," in *American Control Conference*, 2015.

[35] Fleming, W. and Rishel, R., *Deterministic and stochastic optimal control*. Springer, 1975.

[36] Garnett, T., Smith, J., and Lane, R., "Design and flight test of the active arm external load stabilization system," in *American Helicopter Society Forum*, 1976.

[37] Gelb, A., *Applied optimal estimation*. MIT press, 1974.

[38] Gera, J. and Farmer Jr., S., "A method of automatically stabilizing helicopter sling loads," Tech. Rep. D-7593, NASA, 1974.

[39] Gupta, N. K. and Ryson, A., "Near-hover control of a helicopter with a hanging load," *Journal of Aircraft*, vol. 13, no. 3, pp. 217–222, 1976.

[40] Hairer, E., Lubich, C., and Wanner, G., *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer Science & Business Media, 2006.

[41] Hamers, M., von Hinüber, E., and Richter, A., "Ch53g experiences with a flight director for slung load handling," in *American Helicopter Society Forum*, 2008.

[42] Hauser, J., "A projection operator approach to the optimization of trajectory functionals," in *IFAC World Congress*, 2002.

[43] Hauser, J., "On the controllability of the pendubot," in *IFAC Symposium on Nonlinear Control Systems*, 2010.

[44] Hauser, J., "Projection operator strategies in the optimization of trajectory functionals," Tech. Rep. AFRL-OSR-VA-TR-2012-1090, University of Colorado, Boulder, CO, 2012.

[45] Hauser, J. and Meyer, D. G., "The trajectory manifold of a nonlinear control system," in *IEEE Conference on Decision and Control*, 1998.

[46] Häusler, A. J., Saccon, A., Aguiar, A. P., Hauser, J., and Pascoal, A. M., "Cooperative motion planning for multiple autonomous marine vehicles," in *Conference on Maneuvering and Control of Marine Craft*, 2012.

[47] Head, E., "Production potential," *Vertical Magazine*, 2014.

[48] Hover, F. S. and Triantafyllou, M. S., "Application of polynomial chaos in stability and control," *Automatica*, vol. 42, no. 5, pp. 789–795, 2006.

[49] Huang, G., Lu, Y., and Nan, Y., "A survey of numerical algorithms for trajectory optimization of flight vehicles," *Science China Technological Sciences*, vol. 55, no. 9, pp. 2538–2560, 2012.

[50] Hull, D. G., *Optimal control theory for applications*. Springer, 2003.

[51] Hutto, A., "Flight-test report on the heavy-lift helicopter flight-control system," *Journal of the American Helicopter Society*, vol. 21, no. 1, pp. 32–40, 1976.

[52] Iserles, A., *A first course in the numerical analysis of differential equations*. Cambridge University Press, 2009.

[53] Ivler, C., *Design and Flight Test of a Cable Angle Feedback Control System for Improving Helicopter Slung Load Operations at Low Speed*. PhD thesis, Stanford University, 2012.

[54] Ivler, C. M., Tischler, M. B., and Powell, J. D., "Cable angle feedback control systems to improve handling qualities for helicopters with slung loads," in *AIAA Guidance, Navigation, and Control Conference*, 2011.

[55] Jacobson, D. H. and Mayne, D. Q., "Differential dynamic programming," 1970.

[56] Johnson, E., Schultz, J., and Murphey, T., "Structured linearization of discrete mechanical systems for analysis and optimal control," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 140–152, 2015.

[57] Johnson, E. N. and Schrage, D. P., "The georgia tech unmanned aerial research vehicle: Gtmax," in *AIAA Guidance, Navigation, and Control Conference*, 2003.

[58] Johnson, E. and Murphey, T., "Scalable variational integrators for constrained mechanical systems in generalized coordinates," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1249–1261, 2009.

[59] Johnson, E. N. and Kannan, S. K., "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 524–538, 2005.

[60] Johnson, E. N. and Mishra, S., "Flight simulation for the development of an experimental uav," in *AIAA Modeling and Simulation Technologies Conference*, 2002.

[61] Johnson, E. N. and Schrage, D. P., "System integration and operation of a research unmanned aerial vehicle," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, no. 1, pp. 5–18, 2004.

[62] Kang, K., *Online optimal obstacle avoidance for rotary-wing autonomous unmanned aerial vehicles*. PhD thesis, Georgia Institute of Technology, 2012.

[63] Kelley, C. T., *Iterative methods for optimization*. Siam, 1999.

[64] Khalid, A., Huey, J., Singhose, W., Lawrence, J., and Frakes, D., "Human operator performance testing using an input-shaped bridge crane," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 4, pp. 835–841, 2006.

[65] Kirk, D. E., *Optimal control theory: an introduction*. Prentice Hall, 1970.

[66] Kloeden, P. E. and Platen, E., *Numerical Solution of Stochastic Differential Equations*. Springer Science & Business Media, 1992.

[67] Lantoine, G. and Russell, R. P., "A hybrid differential dynamic programming algorithm for robust low-thrust optimization," in *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*, 2008.

[68] Lew, A., Marsden, J. E., Ortiz, M., and M., W., "An overview of variational integrators," in *Finite Element Methods: 1970's and Beyond*, pp. 98–115, 2004.

[69] Lew, A., Marsden, J. E., Ortiz, M., and West, M., "Variational time integrators," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 1, pp. 153–212, 2004.

[70] Lew, A., *Variational time integrators in computational solid mechanics*. PhD thesis, California Institute of Technology, 2003.

[71] Li, W. and Valero-Cuevas, F., "Linear quadratic optimal control of contact transition with fingertip," in *American Control Conference*, 2009.

[72] Li, W. and Todorov, E., "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *International Conference on Informatics in Control, Automation, and Robotics*, 2004.

[73] Liu, D. T., "In-flight stabilization of exernally slung helicopter loads.," Tech. Rep. NORT-72-39, Northrop Corp, 1973.

[74] Ludington, B., Johnson, E. N., and Vachtsevanos, G. J., "Vision based navigation and target tracking for unmanned aerial vehicles," in *Advances in Unmanned Aerial Vehicles*, pp. 245–266, 2007.

[75] Lure, A. I., *Analytical mechanics*. Springer Science & Business Media, 2002.

[76] Lusardi, J. A., Blanken, C. L., Braddom, S. R., Cicolani, L. S., and Tobias, E. L., "Development of external load handling qualities criteria for rotorcraft," in *American Helicopter Society Forum*, 2010.

[77] Marsden, J. E. and West, M., "Discrete mechanics and variational integrators.," *Acta Numerica*, vol. 10, pp. 357–514, 2001.

[78] Marsden, J. E. and Ratiu, T. S., *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Springer, 1999.

[79] McGonagle, J. G., "The design, test, and development challenges of converting the k-max helicopter to a heavy lift rotary wing uav," in *American Helicopter Society Forum*, 2001.

[80] Micale, E. C. and Poli, C., "Dynamics of slung bodies utilizing a rotating wheel for stability," *Journal of Aircraft*, vol. 10, no. 12, pp. 760–763, 1973.

[81] Milam, M., Mushambi, K., and Murray, R., "Ntg - a library for real-time trajectory generation," 2002.

[82] Milam, M. B., *Real-time optimal trajectory generation for constrained dynamical systems*. PhD thesis, California Institute of Technology, 2003.

[83] Milam, M. B., Mushambi, K., and Murray, R. M., "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *IEEE Conference on Decision and Control*, 2000.

[84] Milstein, G., Repin, Y. M., and Tretyakov, M. V., "Symplectic integration of hamiltonian systems with additive noise," *SIAM Journal on Numerical Analysis*, vol. 39, no. 6, pp. 2066–2088, 2002.

[85] Morari, M. and Lee, J. H., "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 45, pp. 667–682, 1999.

[86] Morimoto, J. and Atkeson, C. G., "Minimax differential dynamic programming: An application to robust biped walking," in *Intelligent Robots and Systems*, 2003.

[87] Nagy, Z. and Braatz, R., "Distributional uncertainty analysis using power series and polynomial chaos expansions," *Journal of Process Control*, vol. 17, no. 3, pp. 229–240, 2007.

[88] Naidu, D. S., *Optimal control systems*. CRC press, 2002.

[89] Notarstefano, G. and Hauser, J., "Modeling and dynamic exploration of a tilt-rotor vtol aircraft," in *IFAC Symposium on Nonlinear Control Systems*, 2010.

[90] Ober-Blöbaum, S., Junge, O., and Marsden, J. E., "Discrete mechanics and optimal control: an analysis," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 17, no. 2, pp. 322–352, 2011.

[91] Oladyshkin, S. and Nowak, W., "Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion," *Reliability Engineering & System Safety*, vol. 106, pp. 179–190, 2012.

[92] Ortega, R., Spong, M. W., Gómez-Estern, F., and Blankenstein, G., "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1218–1233, 2002.

[93] Ottander, J. A. and Johnson, E. N., "Precision slung cargo delivery onto a moving platform," in *AIAA Modeling and Simulation Technologies Conference*, 2010.

[94] Palunko, I., Cruz, P., and Fierro, R., "Agile load transportation : Safe and efficient load manipulation with aerial robots," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[95] Palunko, I., Fierro, R., and Cruz, P., "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *IEEE International Conference on Robotics and Automation*, 2012.

[96] Palunko, I. and Fierro, R., "Adaptive control of a quadrotor with dynamic changes in the center of gravity," in *IFAC World Congress*, 2011.

[97] Pasini, J. M. and Sahai, T., "Polynomial chaos based uncertainty quantification in hamiltonian and chaotic systems," in *IEEE Conference on Decision and Control*, 2013.

[98] Press, W. H., *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[99] Prosser, D. T. and Smith, M. J., "A novel, high fidelity 6-dof simulation model for tethered load dynamics," in *American Helicopter Society Forum*, 2013.

[100] Rao, A. V., "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[101] Rosen, A., Ronen, T., and Raz, R., "Active aerodynamic stabilization of a helicopter/sling-load system," *Journal of Aircraft*, vol. 26, no. 9, pp. 822–828, 1989.

[102] Saccon, A., Hauser, J., and Aguiar, A. P., "Optimal control on non-compact lie groups: A projection operator approach.," in *IEEE Conference on Decision and Control*, 2010.

[103] Sahasrabudhe, V., Faynberg, A., Pozdin, M., Cheng, R., Tischler, M., Stumm, A., and Lavin, M., "Balancing ch-53k handling qualities and stability margin requirements in the presence of heavy external loads," in *American Helicopter Society Forum*, 2007.

[104] Schultz, J. and Murphey, T., "Trajectory generation for underactuated control of a suspended mass," in *IEEE International Conference on Robotics and Automation*, 2012.

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

[106] Singer, N., Singhose, W., and Kriikku, E., "An input shaping controller enabling cranes to move without sway," in *ANS Topical Meeting on Robotics and Remote Systems*, 1997.

[107] Singhose, W. E., *Command Generation for Flexible Systems*. PhD thesis, Massachusetts Institute of Technology, 1997.

[108] Sreenath, K., Lee, T., and Kumar, V., "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *IEEE Conference on Decision and Control*, 2013.

[109] Sreenath, K., Michael, N., and Kumar, V., "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *IEEE International Conference on Robotics and Automation*, 2013.

[110] Templeton, B. A., *A Polynomial Chaos Approach to Control Design*. PhD thesis, Virginia Polytechnic Institute and State University, 2009.

[111] Theodorou, E., Tassa, Y., and Todorov, E., "Stochastic differential dynamic programming," in *American Control Conference*, 2010.

[112] Todorov, E. and Li, W., "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*, 2005.

[113] Von Stryk, O. and Bulirsch, R., "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, no. 1, pp. 357–373, 1992.

[114] Wang, L., Hong, J., Scherer, R., and Bai, F., "Dynamics and variational integrators of stochastic hamiltonian systems," *International Journal of Numerical Analysis and Modeling*, vol. 6, no. 4, pp. 586–602, 2009.

[115] West, M., *Variational integrators*. PhD thesis, California Institute of Technology, 2004.

[116] Wiener, N., "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[117] Wu, A. D., Johnson, E. N., and Proctor, A. A., "Vision-aided inertial navigation for flight control," *Journal of Aerospace Computing, Information, and Communication*, vol. 13, no. 3, pp. 63–71, 2006.

[118] Xie, X., Huang, J., and Liang, Z., "Using continuous function to generate shaped command for vibration reduction," *Journal of Systems and Control Engineering*, vol. 227, no. 6, pp. 523–528, 2013.

[119] Xiu, D., *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.

[120] Yakowitz, S., "The stagewise kuhn-tucker condition and differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 31, no. 1, pp. 25–30, 1986.