State Estimation Techniques for Hybrid Dynamical Systems

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Mechanical Engineering

J. Joe Payne

B.S., Mechanical Engineering, University of Michigan

B.S., Computer Science Engineering, University of Michigan

Carnegie Mellon University
Pittsburgh, PA
December, 2024

Acknowledgements

There are innumerable thanks that need to be given for all of the support I've received during this process. But in this section I suppose I have to do my best to enumerate them.

The research presented within this thesis was supported by the National Science Foundation under grants #CMMI-1943900 and #IIS-1704256, the U.S. Army Research Office under grant #W911NF-19-1-0080, and Palladyne AI.

Firstly, I would like to thank all of the people who have advised me throughout this process: Prof. Aaron Johnson – I couldn't have asked for a better advisor. You provided the right balance of guidance and freedom to pursue what I found exciting. Your focus on ensuring your students are taking ample breaks and achieving work life balance made this whole process a joy. Dr. Daniel Hagen – on top of joining my committee, you have acted almost as a second advisor to me during the final year of my program. Your hands on guidance has been a great help. Prof. Zac Manchester – your class inspired an entire section of this thesis, thank you for guiding me toward optimization-based estimation and for being a member of my committee. Prof. Mark Bedillion – thank you for always being a supportive face in the audience during stressful presentations and for being a member of my committee.

Beyond advising, I have received massive support from both coauthors and labmates. As such, I would like to thank all of my coauthors during this process: Dr. Nathan Kong, Dr. James Zhu, Hans Kumar, and Dr. George Council. Working with each of you has been wonderful. Thank you to all of my labmates for creating such great lab culture together: Cat, Joe, Ankit, Nathan, Sean, James, Paul, Ardalan, Stutt, David, Steven, Diana, and Naomi. Thank you to all of the postdocs I have worked with for the guidance they have provided: Vahid, George, Justin, Aja and Vivek.

There were many people involved in putting me on this academic track, but there are three I would like to call out in particular. Ryan Parrott for encouraging my love of math and physics in high school despite being a class clown. Prof. David R. Dowling for encouraging me to consider graduate school when it was the last thing on my mind. Prof. C. David Remy for giving me an early opportunity to work in a robotics lab as an undergrad.

I have so many other people to thank for their support up to and throughout my Ph.D. Firstly, thanks to my parents for everything, specifically for nurturing my intellectual curiosity while growing up and being people I know I can always call for support or just to vent to. I don't think I would be here if it weren't for you two encouraging me throughout my whole academic journey. Jack, thanks for putting up with me growing up & sending me countless entertaining messages. To my grandparents: thank you for believing in me even more than I believe in myself at times.

I have had the support of so many friends during this thesis. Ethan Brownell, thanks for being my first friend (and roommate!) in Pittsburgh, I couldn't imagine having a better start to our program. Connor Forsythe, thank you for being part of my A-List Entourage and for all of the movie conversations. I hope we can continue to support one another as well as we did during the miserable time we were on the job hunt. Isabel Osgood, for being one of my oldest friends, keeping my ego in check at times, and always being open to an hour long phone call just because I'm bored. Aaron Lowe for organizing countless game nights and maintaining an amazing, friendly, supportive community – thank you to everyone in the OGCord as well. We're going to get so many games in after I submit this document. Nathan Kong, beyond being an excellent coauthor, you're one of my closest friends, and I can't picture a world where we aren't chatting almost every day. Catherine Pavlov, you'll always be my favorite neighbor and I know we can always stay connected over the newest silly videos.

In a truly stereotypical fashion, rock climbing has been my hobby of choice to de-stress. I've had the opportunity to meet so many amazing friends through rock climbing, but I'll do my best to cover as many as possible. Michael Brown, you've pushed me to be a better and more confident outdoor climber – love you Mikeeeeey. Jeremy Lipshaw, you helped get me into climbing initially and have always been down to climb when I'm in town. Jon, Cat, Arjun, Aria, Jackie, Jay, Ben, Christian, Mike, Audrey, you have all helped to make weeknight climbing sessions a great escape from the stresses of the day. I also have to thank the rest of the Iron City Boulders community generally for being such a fun group of people.

Lastly, I want to thank Dina Kaganer, you've been a true source of love and joy during this time. You offered me immeasurable support during some of the most stressful parts of this process. You are slowly but surely nudging me out of my comfort zone and I'm excited to see where that takes us and the happy dances we'll share!

Abstract

Intermittent contact is ubiquitous in robotic systems. It can take the form of legged robots making and breaking contact with the ground while walking, running, or locomoting in general. This also occurs in manipulator robots while grasping, reconfiguring, and placing objects. For robots to be useful, it is necessary for them to physically interact with their environments.

To perform meaningful tasks, robots need an accurate estimate of their current position within that environment as well as their current contact state. This would include knowledge of whether a given foot is on the ground or whether a finger is currently touching an object. Without knowledge of this state, a robot will have difficulty determining reasonable control actions to take. If a foot is in contact with the ground, it can exert large contact forces on the robot, however if that same foot is even slightly above the ground, it won't be able to exert any contact forces. Without this knowledge, a robot can easily be induced to trip and fall. Similarly, this can mean the difference between properly grasping an object or failing to do so like pinching thin air or making an insecure grasp for a manipulator robot.

Most existing state estimation methods assume that the underlying system has smooth dynamics. The nonsmooth or even discontinuous nature of contact systems causes issues for these existing state estimation methods as their assumptions do not hold. This work addresses the problem of state estimation for contact systems by analyzing them as hybrid dynamical systems and improving upon existing algorithms to operate on this class of systems.

One aspect of this problem that this thesis solves is when we don't have access to accurate system models. Some examples of this include unknown ground height for legged locomotion or uncertainty in object dimensions for manipulation tasks. We handle this by explicitly considering model uncertainty in state estimation algorithms. To do this, we use probability distributions to represent uncertainties and calculate how variations in guard and reset parameters affect gradients through transition events to create an augmented form of the saltation matrix. This work is presented as the uncertainty aware Salted Kalman Filter (ua-SKF).

Another part of the problem this work solves is when ground truth information isn't available to test online estimation algorithms. In these cases where ground truth is not readily available directly through sensors such as motion capture cameras, we optimize over the available measurements to create a proxy for ground truth to compare against for online algorithm evaluation. Using system dynamics and measurement information from the entire trajectory rather than only information prior to each timestep results in more accurate estimates at each time. This work is presented as Hybrid iterative Linear Quadratic Estimation (HiLQE).

Lastly, we estimate the current contact conditions using methods specifically designed for bipedal robots without foot contact sensors. This is a particularly difficult problem as existing algorithms for state estimation either utilize contact sensors or make assumptions that do not generally hold for bipedal robots, such as the assumption that the robot's legs represent significantly less mass than the body. This work uses multiple momentum observers to detect touchdown events and maintain an estimate of the current contact conditions. By utilizing momentum observers with the assumption that the current contact feet are effectively fixed to the ground, we can avoid assumptions about the mass distribution of the robot.

The work presented in this thesis greatly improves the performance of state estimation on hybrid dynamical systems in a variety of situations. Through the use of these techniques, many contact systems including legged robots and manipulators will perform better when handling contact events. By considering environmental uncertainty, robots will be better able to operate in less structured environments such as in the wilderness or through the imperfections of urban settings. With access to offline optimization methods, online algorithms can be tuned to offer greater performance. Access to a better estimate of hybrid modes enables more dynamic motions as control decisions that rely on knowledge of the current contact mode can be made at a faster pace.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Legged robotic systems have recently started moving out of labs and other controlled environments into more unpredictable environments like wilderness and cities. The use of robots in these contexts has the potential to remove humans from dangerous or repetitive tasks. Some tasks which we hope to use robots in place of humans include search and rescue in wilderness or disaster conditions which could lead to physical harm of people, as well as environmental and system monitoring in remote locations which could prove hazardous for chemical or other reasons. Tasks like these require robots to interact with their environments through contact. Contact can be required for a variety of actions, whether it is for locomotion due to foot touchdown and liftoff, or for manipulation tasks such as performing pick and place in warehouse environments, using tools to sample environments or adjusting the controls of systems they seek to monitor.

To complete their tasks, robots need reliable planning, estimation, and control. For contact systems, these can present additional difficulties as the states of contact systems can be non-smooth and potentially discontinuous. Traditional algorithms for planning [1], estimation [2], and control [3] are not designed with contact events in mind, so they cannot be directly applied to hybrid systems.

To handle the discontinuous nature of these contact systems, we examine them within the framework of hybrid dynamical systems [4, 5, 6]. Hybrid dynamical systems simultaneously represent the continuous and discrete states of a system. The continuous states typically considered for robots are position, velocity, orientation, angular velocity, and joint angles. Typically, the discrete states that are considered are whether a given foot is in contact with the ground or if a hand/finger is making contact with an object. The framework of hybrid dynamical systems also enables the consideration of instantaneous events to update the state, both discrete and continuous, when certain conditions are met such as the height of a foot reaching ground level.

The goal of this work is to improve estimation on these contact driven systems since estimation is critical to both planning and control. Without reliable state estimation, the starting point for planning may be inaccurate, which can cause infeasible plans to be selected as optimal. Similarly, without an accurate estimate of the current state, control actions may be selected that rely on inaccurate contact information, leading the system to expect reaction forces where there won't be any, and conversely expect to be able to move freely when contact may restrict that motion.

To improve performance around contact events, this thesis addresses the problem of state estimation for this class of hybrid dynamical systems. We address this problem with algorithms which seek to estimate state with system structural uncertainty, perform offline algorithm evaluation, and tackle issues specific to bipedal robotic systems. We utilize the saltation matrix and other analysis methods for hybrid dynamical systems to redesign traditional state estimation algorithms for contact systems. The algorithms presented either improve performance over existing algorithms or adapt algorithms that would not work on hybrid dynamical systems previously. In combination, these algorithms will greatly improve the performance of robotic systems with intermittent contact. The rest of the thesis is organized as follows:

1.2 Hybrid Systems and the Saltation Matrix

Chapter 2 introduces the definition of a hybrid dynamical system and provides an overview of the saltation matrix, a tool for linearizing these systems. The work presented in this chapter was performed with Dr. Nathan Kong and is intended to serve as a survey of the field as well as a tutorial on the use of the saltation matrix in a variety of engineering contexts. The paper this chapter is sampled from appears in the Proceedings of the IEEE [6].

1.3 Uncertainty Aware Kalman Filtering

Chapter 3 presents the uncertainty aware Salted Kalman Filter (ua-SKF). This algorithm builds off of the salted Kalman filter (SKF) [7] designed to be applied to hybrid systems, which I developed in conjunction with Dr. Nathan Kong, though he led the project, so it does not appear in this thesis. The modifications presented in this thesis are designed to consider structural uncertainty. This includes uncertainty both in guard location as well as uncertainty in reset map parameters. The explicit consideration of the guard and reset uncertainty at hybrid events in addition to the saltation terms results in better estimation performance around impact events for uncertain hybrid systems. This can be used in situations such as walking on uneven ground where the exact height isn't known ahead of time, or if the system parameters, such as coefficient of restitution or ground slope, are unknown. By explicitly considering the uncertainty in parameters, a better estimate of the covariance of a distribution can be achieved. This improved estimate results in better performance in the filtering algorithm. This paper was presented at IROS 2022 [8].

1.4 Optimization Based Estimation

In Chapter 4, we use iterative linear quadratic (iLQ) methods for state estimation on hybrid dynamical systems. The initial work for this is to perform an offline batch estimation algorithm to create a pseudo-ground truth for online algorithms to use as a comparison point for evaluation.

When we consider the entire trajectory of measurements and dynamics, we are able to achieve more accurate state estimates than simply using online information. An advantage of this method is the ability to achieve better knowledge of impact times, which notably reduces errors around hybrid events. This could potentially be extended as an online moving horizon estimation algorithm.

1.5 Hybrid Mode Detection for Bipedal Robots

The issue of current contact mode estimation is of specific interest for bipedal robotic systems due to their limited contact points and need for stability. To address this issue, Chapter 5 estimates the current contact state of a bipedal robot without relying on common assumptions made in existing contact mode estimators. Existing estimation algorithms for legged robots often assume the robot has contact sensors at the feet or ground reaction force sensors. For systems lacking these sensors, another assumption that is often made is that the legs represent significantly less mass than the floating base. These assumptions often do not hold for bipeds as contact sensors may not be durable enough for long term use and legs often represent a significant portion of the robot's mass. In place of these assumptions, we take the assumption that flight phases will not occur. We then utilize momentum observers for each contact mode to detect touchdown and liftoff events of the other leg.

1.6 Additional Work

While they will not appear in this thesis, I have worked on a variety of other projects during my time in this program. With Hans Kumar, I worked on periodic SLAM [9], which seeks to solve the state estimation problem for hybrid systems (specifically legged robots) from a different direction. This work leverages the periodicity of the gait cycles of many legged robots to improve the estimation on these systems. In this work, we exploited this periodicity to examine camera frames that are close to one another in orientation rather than close to one another in time. Periodic SLAM was able to provide accurate state estimates in scenarios where many state of the art algorithms were induced to fail. Additionally, I worked with Dr. James Zhu on the dual problem of planning and control of hybrid systems [10]. In this work, we utilize an iLQR algorithm to simultaneously solve a trajectory planning problem and optimize the resulting trajectory for convergence properties.

Chapter 2

The Saltation Matrix

This chapter defines the saltation matrix and the broad class of hybrid systems where the saltation matrix applies (Sec. 2.1), derives the expression of the saltation matrix using a geometric approach (Sec. 2.2), and demonstrates the use of saltation matrices in linear (Sec. 2.3) and quadratic forms (Sec. 2.4).

2.1 Hybrid systems and the saltation matrix

While there are many definitions of hybrid dynamical systems, e.g. [11, 12, 13, 5], this treatment of the saltation matrix is based on the definition from [6].

Definition 1 *A C^r **hybrid dynamical system**, for continuity class $r \in \mathbb{N}_{>0} \cup \{\infty, \omega\}$, is a tuple $\mathcal{H} := (\mathcal{J}, \Gamma, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$ where the parts are defined as:*

1. *$\mathcal{J} := \{\mathrm{I}, \mathrm{J}, ...\} \subset \mathbb{N}$ is the finite set of discrete **modes**.*

2. *$\Gamma \subseteq \mathcal{J} \times \mathcal{J}$ is the set of discrete **transitions** forming a directed graph structure over $\mathcal{J}$.*

3. *$\mathcal{D} := \amalg_{\mathrm{I} \in \mathcal{J}} D_{\mathrm{I}}$ is the collection of **domains**, where D_{I} is a C^r manifold and the state $x \in D_{\mathrm{I}}$ while in mode* I.

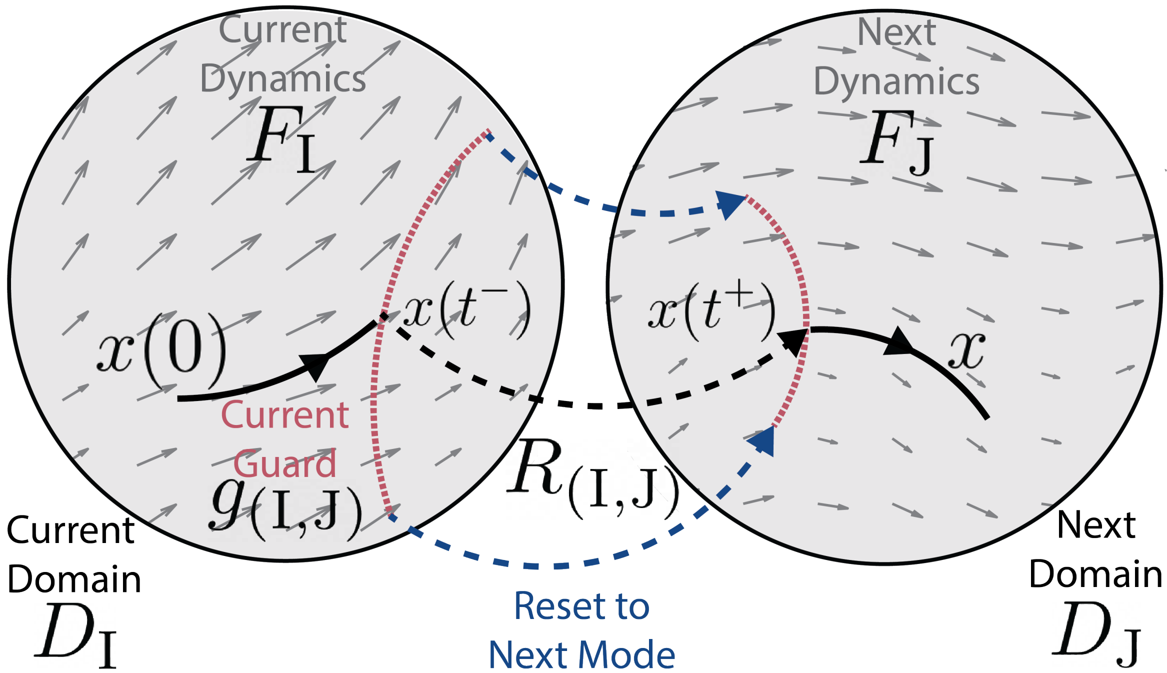


Figure 2.1: An example 2 mode hybrid system where the domains are shown in black circles D, the dynamics are shown with gray arrows F, the guard for the current domain is shown in red dotted lines g, and the reset from the current mode to the next mode is shown in blue dashed lines R.

4. *$\mathcal{F} : \mathbb{R} \times \mathcal{D} \to \mathcal{T}\mathcal{D}$ is a collection of C^r time-varying **vector fields**, $F_{\mathrm{I}} := \mathcal{F}|_{D_I} : \mathbb{R} \times D_{\mathrm{I}} \to \mathcal{T}D_{\mathrm{I}}$, for each $\mathrm{I} \in \mathcal{J}$.*

5. *$\mathcal{G} := \amalg_{(\mathrm{I,J})\in\Gamma}\, G_{(\mathrm{I,J})}(t)$ is the collection of **guard sets**, where $G_{(\mathrm{I,J})}(t) \subseteq D_{\mathrm{I}}$ for each $(\mathrm{I,J}) \in \Gamma$ is defined as a regular sublevel set of a C^r guard function, i.e. $G_{(\mathrm{I,J})}(t) = \{x \in D_{\mathrm{I}} | g_{(\mathrm{I,J})}(t,x) \le 0\}$ and $\mathrm{D}_x g_{(\mathrm{I,J})}(t,x) \neq 0\ \forall\ g_{(\mathrm{I,J})}(t,x) = 0$.*

6. *$\mathcal{R} : \mathbb{R} \times \mathcal{G} \to \mathcal{D}$ is a C^r map called the **reset** that restricts as $R_{(\mathrm{I,J})} := \mathcal{R}|_{G_{(\mathrm{I,J})(t)}} : G_{(\mathrm{I,J})}(t) \to D_{\mathrm{J}}$ for each $(\mathrm{I,J}) \in \Gamma$.*

*Note that this definition incorporates the **control input** $u(t,x)$ into the dynamics $\mathcal{F}$ as $\mathcal{F}(t,x,u(t,x))$, which we simplify as $\mathcal{F}(t,x)$ going forward.*

Fig. 2.1 shows an example hybrid system with a hybrid execution consisting of a starting point $x(0)$ in D_{I} flowing with dynamics F_{I} and reaching the guard condition $g_{(\mathrm{I,J})}(t,x) = 0$ at time t, applying the reset map $R_{(\mathrm{I},\mathrm{J})}(t,x)$ resetting into D_J and then flowing with the new dynamics F_J. Denote t^- as the instant before a hybrid event occurs while the system is still in domain I, t^+ the instant after the reset map is applied following the hybrid event where the system has transitioned into domain J, and $x(t^{\pm}) = x^{\pm}$ the limiting value of the signal x from the left $(-)$ or right $(+)$.

The goal in this chapter is to understand how variations about a nominal trajectory evolve over time. For smooth systems, it is well known that variations about a nominal trajectory, δx ,can be approximated to first order using the derivative of the dynamics $F(t,x)$ with respect to state D_x:

$$\frac{d}{dt}\delta x(t) = \mathrm{D}_x F(t,x)\delta x + \text{h.o.t.} \tag{2.1}$$

where h.o.t. represents higher order terms. Hybrid systems with time triggered reset maps can be similarly analyzed using the Jacobian of the reset map, $\delta x^+ = \mathrm{D}_x R(t,x)\delta x^-$. However, the Jacobian of the reset map does not account for differences that are introduced from time-to-impact variations in systems with event driven resets, where the differences in dynamics in the two hybrid modes must be considered. The saltation matrix, e.g. [14, Eq. 3.5], [15, Pg. 118 Eq. 6], or [16, Eq. 7.65], accounts for these terms to capture how variations are mapped through event-driven hybrid transitions to the first order. From here on, the term hybrid transition/system refers to this event-driven class.

For notational simplicity, the following shorthands are made for the terms in the saltation matrix:

$$F_\mathrm{I}^- := F_\mathrm{I}(t^-, x(t^-)) \tag{2.2}$$

$$F_\mathrm{J}^+ := F_\mathrm{J}(t^+, x(t^+)) \tag{2.3}$$

$$x(t^+) := R_{(\mathrm{I},\mathrm{J})}(t^-, x(t^-)) \tag{2.4}$$

$$\mathrm{D}_x R^- := \mathrm{D}_x R_{(\mathrm{I},\mathrm{J})}(t^-, x(t^-)) \tag{2.5}$$

$$\mathrm{D}_t R^- := \mathrm{D}_t R_{(\mathrm{I},\mathrm{J})}(t^-, x(t^-)) \tag{2.6}$$

$$\mathrm{D}_x g^- := \mathrm{D}_x g_{(\mathrm{I},\mathrm{J})}(t^-, x(t^-)) \tag{2.7}$$

$$\mathrm{D}_t g^- := \mathrm{D}_t g_{(\mathrm{I},\mathrm{J})}(t^-, x(t^-)) \tag{2.8}$$

Note that D_t in (2.6) and (2.8) refers to the derivative with respect to the first coordinate (and not the time dependence of x, which is captured by other terms). Now, we can define the saltation matrix as follows.

Definition 2 *The* ***saltation matrix*** *for transition from mode I to mode J is the first order approximation of the variational update at hybrid transitions from mode I to J, defined as*

$$\boxed{\Xi_{(\mathrm{I},\mathrm{J})} := \mathrm{D}_x R^- + \frac{\left(F_{\mathrm{J}}^+ - \mathrm{D}_x R^- F_{\mathrm{I}}^- - \mathrm{D}_t R^-\right)\mathrm{D}_x g^-}{\mathrm{D}_t g^- + \mathrm{D}_x g^- F_{\mathrm{I}}^-}} \tag{2.9}$$

In the saltation matrix, the first term, $\mathrm{D}_x R^-$, captures the variations due to the reset map being applied at different states. The second term accounts for the variations caused by a trajectory being subject to differing dynamics for a small amount of time due to the displacement.

Note that the matrix multiplication in (2.9) *results in an outer-product between the terms in the parentheses and $\mathrm{D}_x g^-$ to get a rank-1 correction to the Jacobian of the reset map. The saltation matrix is an $n_{\mathrm{J}} \times n_{\mathrm{I}}$ matrix, where n_{I} is the dimension of the states in domain D_{I} and n_{J} is the dimension of the states in domain D_{J}.*

The saltation matrix maps variations to the first order from pre-transition $\delta x(t^-)$ to post-transition $\delta x(t^+)$ as

$$\boxed{\delta x(t^+) = \Xi_{(\mathrm{I},\mathrm{J})}\delta x(t^-) + \textit{h.o.t.}} \tag{2.10}$$

The saltation matrix in (2.9) is well defined when the following assumptions are true:

1. Guards and resets are differentiable

2. Trajectories must be transverse to the guard at an event:

$$\frac{d}{dt} g_{(\mathrm{I},\mathrm{J})}(t, x(t)) = \mathrm{D}_t g^- + \mathrm{D}_x g^- F_{\mathrm{I}}^- < 0 \tag{2.11}$$

In addition, it is often taken that trajectories cannot undergo an infinite number of resets in finite time (no Zeno) in order to ensure trajectories can be analyzed without needing to determine the behavior in limit conditions.

The saltation matrix relies on differentiating the guards and reset maps so they must be differentiable. Transversality ensures that neighboring trajectories impact the same guard unless the impact point lies on any other guard surface, in which case the Bouligand derivative is the appropriate analysis tool [17, 18, 19, 20, 21]. Transversality also ensures the denominator in (2.9) does not approach zero.

These assumptions also indicate the main limitations of the saltation matrix. On top of the limitations inherent to the linearization of nonlinear systems, the saltation matrix assumes that all neighboring trajectories undergo the same transition sequence as the nominal trajectory. This is unable to capture situations where the nominal trajectory transitions transversely to the guard (i.e. grazing impact) or near the intersection of two guard surfaces (i.e. simultaneous touchdown of feet).

In some cases, the saltation matrix for a hybrid transition can become an identity transformation. Knowing when the saltation matrix is identity is useful to simplify computation and analysis. The most common reason for a saltation matrix to become identity is if both of these conditions are true:

1. The reset map is an identity transformation in the neighborhood of the center of approximation, $R(x) = I_{n\times n}x$, where n is the dimension of the state x in both D_I and D_J, additionally this means that $\mathrm{D}_x R = I$.

2. The dynamics in both modes are the same before and after impact, $F_\mathrm{I}^- = F_\mathrm{J}^+$.

With these conditions, we can see that the saltation matrix becomes the identity map:

$$\boxed{\left.\begin{aligned} \mathrm{D}_x R_{(\mathrm{I},\mathrm{J})} = I_{n\times n} \\ F_\mathrm{I}^- = F_\mathrm{J}^+ \end{aligned}\right\} \implies \Xi_{(\mathrm{I},\mathrm{J})} = I_{n\times n}} \tag{2.12}$$

An example of such a transition is a foot lifting off from the ground, since there is no abrupt change in forces, the dynamics are equal at the mode transition. If the reset map is an identity transformation, then $\mathrm{D}_x R$ is also identity and $\mathrm{D}_t R$ is zero. Using these conditions to simplify the expression in (2.9) gives

$$\Xi_{(\mathrm{I},\mathrm{J})} = I_{n\times n} + \frac{\left(F_{\mathrm{J}}^{+} - I_{n\times n}F_{\mathrm{I}}^{-} - 0_{n\times n}\right)\mathrm{D}_x g^{-}}{\mathrm{D}_t g^{-} + \mathrm{D}_x g^{-} F_{\mathrm{I}}^{-}} = I_{n\times n} \tag{2.13}$$

Lastly, in the case that the transition is triggered by time rather than state, the saltation matrix is exactly equal to the Jacobian of the reset map $\mathrm{D}_x R$. This is because there is no longer a variation in the time to impact, and $\mathrm{D}_x g^{-} = 0_{1\times n}$, thus

$$\Xi_{(\mathrm{I},\mathrm{J})} = \mathrm{D}_x R^{-} + \frac{\left(F_{\mathrm{J}}^{+} - \mathrm{D}_x R^{-} F_{\mathrm{I}}^{-} - \mathrm{D}_t R^{-}\right) 0_{1\times n}}{\mathrm{D}_t g^{-} + 0_{1\times n} F_{\mathrm{I}}^{-}} = \mathrm{D}_x R^{-} \tag{2.14}$$

Therefore, in this case, it is safe to use the Jacobian of the reset map instead of the saltation matrix because they are equivalent.

2.2 Saltation matrix derivation

In this section, the derivation of the saltation matrix (2.9) is presented, following the geometric derivation from [16] with the addition of reset maps. There are many alternate ways to derive (2.9), one method is a derivation using the chain rule, which is included in [6].

Suppose the nominal trajectory of interest is $x(t)$ as shown in Fig. 2.2. The trajectory starts in mode I and goes through a hybrid transition to mode J at time t. The saltation matrix is a first-order approximation, so the flow is treated as a constant in each mode, evaluated at time $t^{\pm}$ as in (2.2) and (2.3) such that for an infinitesimal timestep δt:

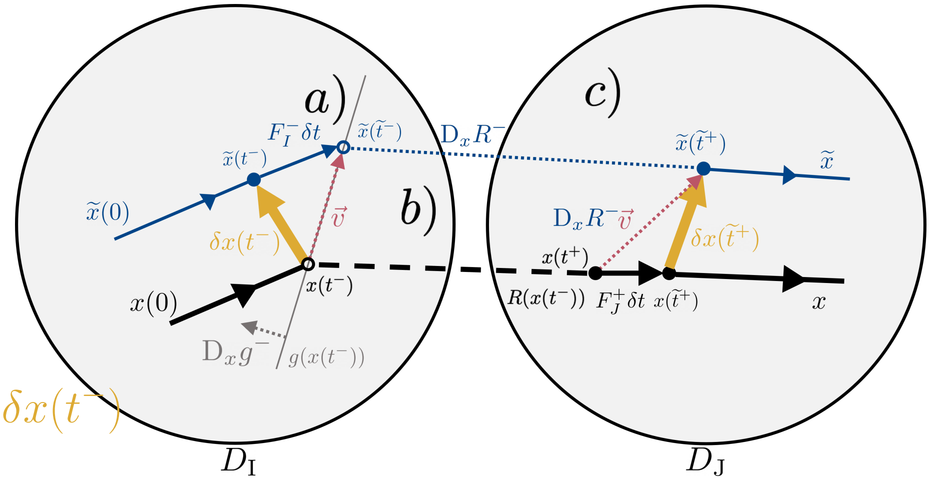
$$x(t^{-}) \approx x(t^{-} - \delta t) + F_{\mathrm{I}}^{-}\delta t \qquad \text{in mode I} \tag{2.15}$$

$$x(t^{+} + \delta t) \approx x(t^{+}) + F_{\mathrm{J}}^{+}\delta t \qquad \text{in mode J} \tag{2.16}$$

The reset and guard are also linearized at t^{-} as in (2.5) and (2.7), such that

$$\bar{R}(t^{-} + \delta t, x + \delta x(t^{-})) = R_{(\mathrm{I},\mathrm{J})}(t^{-}, x(t^{-})) + \mathrm{D}_x R^{-}\delta x(t^{-}) + \mathrm{D}_t R^{-}\delta t \tag{2.17}$$

$$\bar{g}(t^{-} + \delta t, x + \delta x(t^{-})) = g_{(\mathrm{I},\mathrm{J})}(t^{-}, x(t^{-})) + \mathrm{D}_x g^{-}\delta x(t^{-}) + \mathrm{D}_t g^{-}\delta t \tag{2.18}$$



$a)\quad \vec{v} = F_\mathrm{I}^- \delta t + \delta x(t^-)$

$b)\quad 0 = \mathrm{D}_x g^- \vec{v}$

$c)\quad \delta x(\widetilde{t}^+) = \mathrm{D}_x R^- \vec{v} - F_\mathrm{J}^+ \delta t$

Figure 2.2: Linearizations made about the nominal trajectory shown in black where a perturbation is shown in yellow and the perturbed trajectory is shown in blue. At $a)$ describes $\vec{v} = F_\mathrm{I}^- \delta t + \delta x(t^-)$. At $b)$ the guard condition is $0 = \mathrm{D}_x g^-(\delta x(t^-) + F_\mathrm{I}^- \delta t)$. At $c)$ $\delta x(\widetilde{t}^+)$ is $\mathrm{D}_x R^- \vec{v} - F_\mathrm{J}^+ \delta t$. Here δt is positive (late transition) and for the purposes of this figure it is assumed that the system is autonomous, so the $\mathrm{D}_t g$ and $\mathrm{D}_t R$ terms drop out.

where $\bar{R}$ and $\bar{g}$ are the linearization of the reset map and guard function about a nominal trajectory.

Trajectories that are perturbed δx away are labeled as $\widetilde{x}$. Variations can lead to changes in the impact time, which we describe with the infinitesimal time difference $\delta t := \widetilde{t} - t$ where t is the original impact time and $\widetilde{t}$ is the perturbed impact time. If $\delta t > 0$ then the perturbed transition occurs after the nominal solution and the perturbed solution stays in the previous hybrid mode longer, while if $\delta t < 0$ then the perturbed solution transitions early. For simplicity of notation, assume the perturbed trajectory reaches the guard surface late, but the analysis also works for early transitions, resulting in the same expression (2.9), which is shown in [6].

Define the perturbation at the pre-impact time of the nominal trajectory t^- and the post-impact time of the perturbed trajectory $\widetilde{t}^+$, which we will solve for based on the initial state perturbation, as:

$$\delta x(t^-) := \widetilde{x}(t^-) - x(t^-) \tag{2.19}$$

$$\delta x(\widetilde{t}^{+}) := \widetilde{x}(\widetilde{t}^{+}) - x(\widetilde{t}^{+}) \tag{2.20}$$

where $\widetilde{x}(t^{-})$ is the perturbed trajectory following the previous mode dynamics until time t^{-}. Next, we can write (2.20) in terms of the nominal trajectory at time of impact $x(t^{-})$ and just after impact $x(t^{+})$.

Using (2.15) and (2.19), $\widetilde{x}(\widetilde{t}^{-})$ can be written in terms of the flow before impact $F_{\mathrm{I}}^{-}\delta t$ and the perturbation before impact $\delta x(t^{-})$:

$$\widetilde{x}(\widetilde{t}^{-}) = x(t^{-}) + \delta x(t^{-}) + F_{\mathrm{I}}^{-}\delta t + \text{h.o.t.} \tag{2.21}$$

Note that we denote the expression $\delta x(t^{-}) + F_{\mathrm{I}}^{-}\delta t$ as $\vec{v}$ in Fig. 2.2 Eq. a. For brevity, we will drop the higher order terms for the rest of this chapter.

By using the linearized reset map (2.17) and the perturbation expressed in terms of the nominal trajectory (2.21), the reset at $\widetilde{x}(\widetilde{t}^{-})$ can be expressed in terms of the nominal state $x(t^{-})$, the pre-transition perturbation $\delta x(t^{-})$, and the difference in impact time δt

$$\widetilde{x}(\widetilde{t}^{+}) = R(t^{-}, x(t^{-})) + \mathrm{D}_x R^{-}\left(\delta x(t^{-}) + F_{\mathrm{I}}^{-}\delta t\right) + \mathrm{D}_t R^{-}\delta t \tag{2.22}$$

The final term in (2.20) is obtained using the constant flow after the reset (2.15) to calculate $x(\widetilde{t}^{+})$:

$$x(\widetilde{t}^{+}) = R(t^{-}, x(t^{-})) + F_{\mathrm{J}}^{+}\delta t \tag{2.23}$$

By combining (2.20), (2.22), and (2.23), $\delta x(\widetilde{t}^{+})$ is now a linear function of $\delta x(t^{-})$ and δt :

$$\begin{aligned} \delta x(\widetilde{t}^{+}) &= R(t^{-}, x(t^{-})) + \mathrm{D}_x R^{-}\left(\delta x(t^{-}) + F_{\mathrm{I}}^{-}\delta t\right) \qquad (2.24)\\ &\quad + \mathrm{D}_t R^{-}\delta t - \left(R(t^{-}, x(t^{-})) + F_{\mathrm{J}}^{+}\delta t\right) \\ &= \mathrm{D}_x R^{-}\delta x(t^{-}) + \left(\mathrm{D}_x R^{-} F_{\mathrm{I}}^{-} + \mathrm{D}_t R^{-} - F_{\mathrm{J}}^{+}\right)\delta t \qquad (2.25) \end{aligned}$$

This step is highlighted by the vector addition in Fig. 2.2 Eq. c.

Next, we solve for δt as a function of $\delta x(t^-)$. The linearization of the guard (2.18) and the perturbation expressed in terms of the nominal trajectory (2.21) are used to rewrite the guard evaluated at $\widetilde{x}(\widetilde{t}^-)$ as a function of the nominal (and noting that $g(t^-, x(t^-)) = 0$):

$$0 = g(t^-, x(t^-)) + \mathrm{D}_x g^- (\delta x(t^-) + F_{\mathrm{I}}^- \delta t) + \mathrm{D}_t g^- \delta t \tag{2.26}$$

$$= \mathrm{D}_x g^- \delta x(t^-) + (\mathrm{D}_x g^- F_{\mathrm{I}}^- + \mathrm{D}_t g^-) \delta t \tag{2.27}$$

This expansion shows up in Fig. 2.2 as Eq. b. Using 2.18 to write δt as a function of $\delta x(t^-)$ gives:

$$\delta t = -\frac{\mathrm{D}_x g^-}{\mathrm{D}_x g^- F_{\mathrm{I}}^- + \mathrm{D}_t g^-} \delta x(t^-) \tag{2.28}$$

Substituting this δt into (2.25) and solving for $\delta x(\widetilde{t}^+)$ in terms of $\delta x(t^-)$ gives

$$\delta x(\widetilde{t}^+) = \mathrm{D}_x R^- \delta x(t^-) + \frac{\left(F_{\mathrm{J}}^+ - \mathrm{D}_x R^- F_{\mathrm{I}}^- - \mathrm{D}_t R^-\right) \mathrm{D}_x g^-}{\mathrm{D}_x g^- F_{\mathrm{I}}^- + \mathrm{D}_t g^-} \delta x(t^-) \tag{2.29}$$

$$= \Xi_{(\mathrm{I},\mathrm{J})} \delta x(t^-) \tag{2.30}$$

where Ξ is the saltation matrix, as in (2.10).

2.3 Linear forms for the saltation matrix

Understanding how perturbed trajectories behave near a trajectory of interest is crucial for many algorithms which rely on linearizations. The sensitivity equation describes how these variations evolve over time. For a hybrid system, the time evolution simply applies the standard smooth sensitivity equation based on the Jacobian of the dynamics with respect to state, $A_{\mathrm{I}}(t, x) := \mathrm{D}_x F_{\mathrm{I}}(t, x)$ (2.1), and the saltation matrix equation when a hybrid transition occurs (2.10). For a

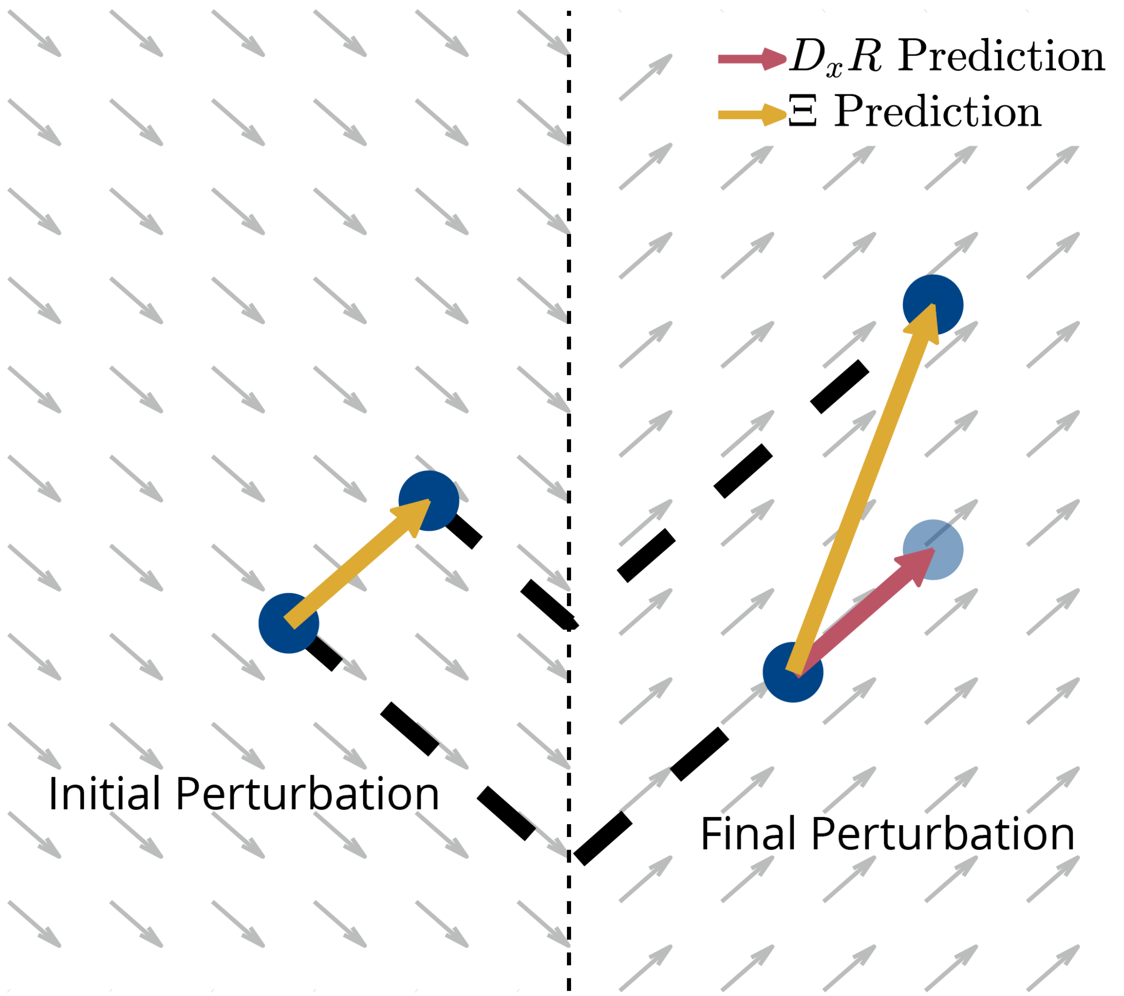


Figure 2.3: Constant flow hybrid system with identity reset map. The Jacobian of the reset map $\mathrm{D}_x R$ predicts no variational changes whereas using the saltation matrix Ξ predicts the correct variational changes.

transition from mode I to mode J at time t^-, the perturbation dynamics are described by

$$\frac{d}{dt}\delta x(t) = A_{\mathrm{I}}\delta x(t) \qquad s.t.\ \ t \leq t^- \tag{2.31}$$

$$\delta x(t^+) = \Xi_{(\mathrm{I},\mathrm{J})}\delta x(t^-) \qquad s.t.\ \ t = t^- \tag{2.32}$$

$$\frac{d}{dt}\delta x(t) = A_{\mathrm{J}}\delta x(t) \qquad s.t.\ \ t \geq t^+ \tag{2.33}$$

An example is shown in Fig. 2.3, where the sensitivity is updated only by the saltation matrix because the flows are constant in both modes (A is zero). Instead, it is the difference in mode timing that determines the change in sensitivity from the initial to final state. If the Jacobian of the reset (which in this case is identity) is used instead of the saltation matrix, the prediction is incorrect. Sensitivity of hybrid systems is extensively analyzed in [22] and [23].

Many algorithms consider finite, discrete timesteps. This makes the analysis slightly different, since the hybrid transition will most likely not occur exactly at the boundary of a discrete timestep. In this case, a "sandwich" method is utilized, where three (or more) smaller discrete updates are applied during a timestep in which a hybrid transition occurs. Consider a time interval from t_k to $t_{k+1} := t_k + \Delta$ over which a single reset occurs at time $t_k + \Delta_1$. The system spends Δ_1 time in the first mode and $\Delta_2 := \Delta - \Delta_1$ in the second mode. In practice, Δ may be chosen based on a desired control update rate, while Δ_1 can be solved for with a zero-crossing algorithm in an event-driven hybrid simulator or similar method. Let $A_{\mathrm{I},\Delta}$ be the Jacobian of the dynamics A_{I} discretized to time duration Δ. Then a discrete approximation of the forward dynamics is

$$\boxed{\delta x(t_{k+1}) = A_{\mathrm{J},\Delta_2}\Xi_{(\mathrm{I},\mathrm{J})}A_{\mathrm{I},\Delta_1}\delta x(t_k)} \tag{2.34}$$

which holds to first order. This result comes from the fundamental matrix solution [16, Eq. 7.22]. Note for the example in Fig. 2.3, the constant flow in each mode means that $A_{\mathrm{I},\Delta_1} = A_{\mathrm{J},\Delta_2} = I$. If multiple (but finitely many) hybrid transitions occur over a time interval, additional A_Δ and Ξ terms can be appended to (2.34) as necessary.

2.4 Quadratic forms for the saltation matrix

Similar to the linear gradient forms from the last section, quadratic forms are often used in algorithms which rely on linearizations. Examples of such algorithms include the well-known Kalman filter and LQR controller, where quadratic forms are used to propagate the covariance distribution and value function approximation, respectively. More formally, these equate to the propagation of the quadratic forms of vectors and co-vectors, respectively, [24, Ch. 3]. This propagation allows for accurate updates to state estimates and control laws along a trajectory. In this context, the vector in question is the state vector, with the corresponding quadratic form being the covariance distribution.

The co-vector, which lies in the space of linear functions of the vector, does not have an explicit representation here, but the quadratic form of the co-vector is the value function approximation of LQR [25].

For covariances, recall that the update law for covariance Σ through a discretized smooth system, with timesteps Δ, is:

$$\Sigma(t_{k+1}) = A_\Delta \Sigma(t_k) A_\Delta^T \tag{2.35}$$

e.g. as in [26, Eqn. 1.10] or [27, Eqn. 6]. Similarly, at hybrid transitions, the saltation matrix applies in an analogous way (see derivation in [6, App. C]):

$$\boxed{\Sigma(t^+) = \Xi_{(\mathrm{I},\mathrm{J})} \Sigma(t^-) \Xi_{(\mathrm{I},\mathrm{J})}^T} \tag{2.36}$$

[28, Eqn. 17], [7, Eqn. 7], which holds to first order. As with linear forms, the sandwich method (2.34) can be applied to retrieve the covariance propagation for an entire discrete timestep:

$$\boxed{\Sigma(t_{k+1}) = A_{\mathrm{J},\Delta_2} \Xi_{(\mathrm{I},\mathrm{J})} A_{\mathrm{I},\Delta_1} \Sigma(t_k) A_{\mathrm{I},\Delta_1}^T \Xi_{(\mathrm{I},\mathrm{J})}^T A_{\mathrm{J},\Delta_2}^T} \tag{2.37}$$

[7, Eqn. 19]. An example is shown in Fig. 2.4, where the covariance is once again updated only by the saltation matrix because the flows are constant in both modes (A_Δ terms are identity). If the Jacobian of the reset is used instead, the incorrect covariance is predicted. Algorithms, such as a Kalman filter [7], that propagate covariances with the dynamics can utilize this update law.

In the case of propagating a quadratic form of a co-vector, the matrix transpose terms flip sides similar to how a co-vector quadratic form propagates in the smooth domain:

$$P(t_k) = A_\Delta^T P(t_{k+1}) A_\Delta \tag{2.38}$$

as in [29, Eqn. 3.40]. This structure compared with (2.35) highlights the dual nature of the co-

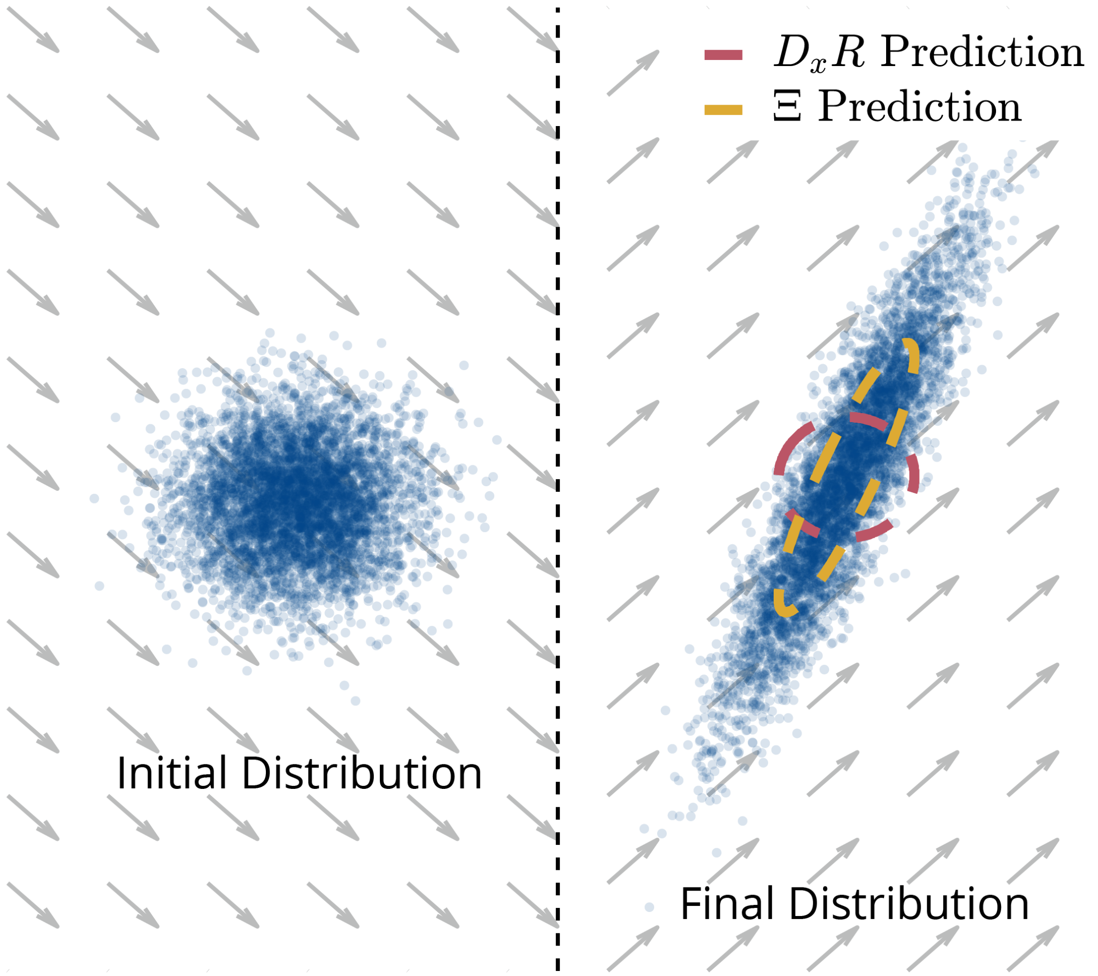


Figure 2.4: Constant flow hybrid system with identity reset map. The Jacobian of the reset map $\mathrm{D}_x R$ predicts no covariance change whereas using the saltation matrix Ξ predicts the correct covariance.

vectors and vectors. The co-vector propagation law for the hybrid transition uses the saltation matrix in an analogous way (see derivation in [6, App. D]):

$$\boxed{P(t^-) = \Xi^T_{(\mathrm{I},\mathrm{J})} P(t^+) \Xi_{(\mathrm{I},\mathrm{J})}} \tag{2.39}$$

[25, Eqn. 23], [30, Eqn. 31]. The main application of the co-vector case is in the update to the Riccati equation or Bellman value function update, e.g. in LQR [30, 25].

Chapter 3

Kalman Filtering for Hybrid Systems with Uncertain Structure

3.1 Abstract

In this chapter, we present a method for updating robotic state belief through contact with uncertain surfaces and apply this update to a Kalman filter for more accurate state estimation. Examining how guard surface uncertainty affects the time spent in each mode, we derive a novel guard saltation matrix – which maps perturbations prior to hybrid events to perturbations after – accounting for additional variation in the resulting state. Additionally, we propose the use of parameterized reset functions – capturing how unknown parameters change how states are mapped from one mode to the next – the Jacobian of which accounts for additional uncertainty in the resulting state. The accuracy of these mappings is shown by simulating sampled distributions through uncertain transition events and comparing the resulting covariances. Finally, we integrate these additional terms into the "uncertainty aware Salted Kalman Filter", uaSKF, and show a peak reduction in average estimation error by 24-60% on a variety of test conditions and systems.

3.2 Introduction

Making and breaking contact is critical for robots as they often need to physically interact with their environment to accomplish their tasks. For a legged robot to navigate to a desired location – for search and rescue, mapping, remote surveying, etc. – its feet will need to repeatedly impact the ground as it walks or runs. Manipulation robots must grasp, push, pull, etc, the objects they need to manipulate. In order to safely and reliably operate during these changing contact conditions, robots need to have an accurate estimation of their state in order to generate reasonable plans and complete their tasks. However, when dealing with these intermittent contacts, the robot's dynamics become non-smooth and even discontinuous, which presents a challenge for classic methods that assume smoothness [31, 32, 33, 34].

Another difficulty with intermittent contact systems is that outside of constrained environments like labs and factories, there will not be perfect models of the environment. The contact surface and physical properties may not be perfectly known ahead of time. In the language of hybrid systems [11, 12, 13], this environmental uncertainty requires us to consider the guards (where contact conditions change) and reset maps (how contact conditions change) to be stochastic. The combination of uncertainty in the guard with the discontinuity in dynamics results in additional state uncertainty due to variation in the time spent in each mode. For example, Fig. 3.1 shows a simple contact system – a ball bouncing off of a slanted surface. If there is uncertainty in the guard location or in the angle at which the ball will rebound, the resulting state uncertainty will grow. Filtering methods present a way to utilize the information of how state and guard uncertainty interact at hybrid events by directly updating the state belief based on hybrid model uncertainty.

To address these issues, this chapter presents an "uncertainty aware Salted Kalman Filter", or uaSKF, for hybrid dynamical systems. We propose to model uncertainty as distributions of guard locations and reset map parameters, and we derive how these distributions couple into the system state uncertainty. We introduce the "guard saltation matrix," which captures the uncertainty due to variations in the guard location and time to impact (Fig. 3.1b). To handle reset uncertainty, we use

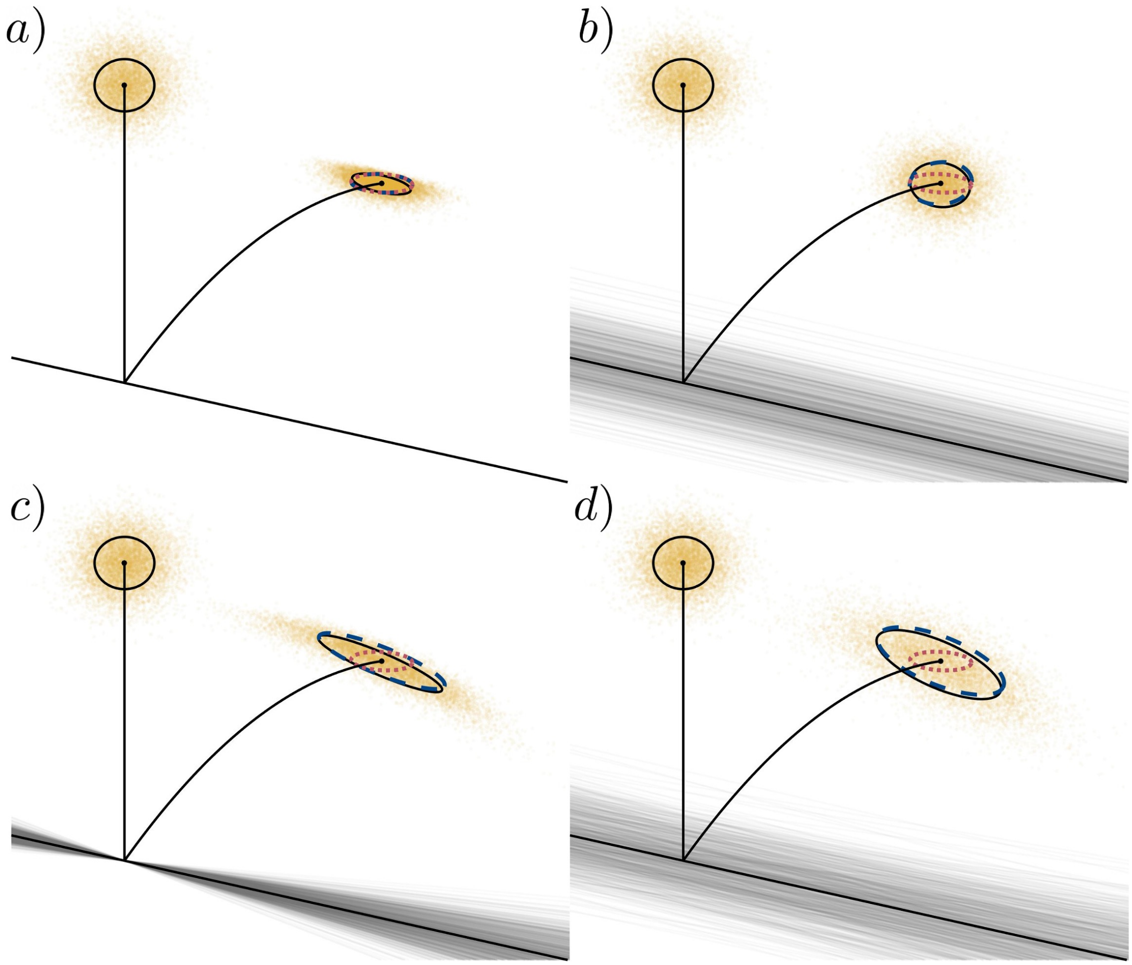


Figure 3.1: Simulating a 2D bouncing ball impacting an angled ground. a) No uncertainty in the guard offset or normal. b) Only offset (guard location) uncertainty. c) Only surface normal (reset function) uncertainty. d) Both offset and normal uncertainty. Yellow particles: Initial and final distribution. Black curve: Nominal trajectory. Black ellipses: Initial and final covariances. Red dotted ellipse: Predicted covariance using only saltation matrix (identical in all four plots). Blue dashed ellipse: Predicted covariance using the proposed method. The proposed method captures the effect of guard and reset uncertainty on the propagated covariance.

the Jacobian of the reset function with respect to the uncertain parameters (Fig. 3.1c). These terms combine to produce an accurate state uncertainty update through the uncertain ground interaction (Fig. 3.1d). Then we integrate these tools into a Kalman filter and provide results showing reduced estimation error on several example systems.

3.3 Related Work

While the fields of hybrid systems and state estimation both have long histories, here we focus on related work that specifically tackles the intersection of the two.

3.3.1 State Estimation for Hybrid Systems

Initial work on Kalman filter (KF) based methods for hybrid dynamical systems such as [31, 34] utilized the reset map to update the mean estimate and the Jacobian of the reset map to update covariance beliefs at hybrid events. However, recent work showed that the Jacobian does not capture all of the effects of the hybrid event on reshaping the covariance. Instead, [7] proposes the "Salted Kalman Filter" (SKF), which uses the saltation matrix in place of the Jacobian of the reset map for Kalman filtering. The saltation matrix is a rank 1 update to the Jacobian that accounts for state variations caused by time to impact variations. This work was extended to include filtering on manifolds in [35] with the hybrid invariant extended Kalman filter (HInEKF). In this work, the saltation matrix method is further extended to account for uncertainties in the structure of the underlying hybrid system, such as variation in guard location and the reset map. To do this, we use techniques similar to consider covariance methods [36, 37] to incorporate uncertainty in system parameters around hybrid events.

Other work on hybrid system state estimation has largely involved multiple estimators. Some have kept the number of estimators relatively low, such as in Interacting Multiple Model estimation (IMM) [38], which maintains KFs for each of the hybrid modes. Multiple model methods have been extended to a variety of problems including nonlinear dynamics [39] and non-identity rests [40]. Multiple model methods are not easily applicable on event-driven hybrid dynamical systems as one of the core assumptions of these methods is that the transitions between discrete modes follow a Markov model, which is not necessarily true when the probabilities of discrete state transitions are dependent on the continuous state beliefs.

Alternatively, many methods such as [41, 42] have adopted particle filtering approaches and have used large numbers of individual estimates to represent a distribution, as opposed to summary statistics like mean and covariance in the case of KFs. While these methods have many benefits, including capturing nonlinear dynamics and non-Gaussian beliefs, the computational complexity of running a particle filter is far greater than a simpler Kalman style filter. As such, this work aims to utilize KFs to maintain the benefits of fast computation times.

3.3.2 Nonlinear Event Mapping and Saltation Matrices

In this chapter, the standard KF is augmented with additional knowledge about the structure of reset maps from the saltation matrix.

The saltation matrix [14, 22, 16, 19] is used to map perturbations through nonsmooth dynamics at the boundary between modes. Previously, [28] demonstrated the saltation matrix can be used to map probability distributions through hybrid transitions and [7] extended this to use in Kalman filtering, as in (Fig. 3.1a).

The primary difference in this work is that perfect knowledge of guard locations and reset maps is not assumed. This is similar to [43] in which they examined the "noisy saltation matrix" for systems with guards that are time varying with random low amplitude, zero mean, mean reverting noise. This chapter uses similar time to impact analysis to derive our guard saltation matrix, which instead views the uncertainty in guard locations as stationary with an estimated distribution they are drawn from. This results in a different mapping, as the noisy saltation matrix views the guard as time varying (so the velocity of the noise affects the time to impact for the system), which is not present in our guard saltation matrix. The other difference is that the noisy saltation matrix does not assume any sort of distribution, only that information about time to impact can be extracted. This work assumes that Gaussian information is known about the guard and directly determines the time to impact from that distribution.

3.4 Background

3.4.1 Hybrid Dynamical Systems

A hybrid dynamical system is a system with both continuous states, such as positions and velocities, and discrete states or modes, such as whether a specific limb is in contact with the ground, in which the sequence of discrete states is determined by the evolution of the continuous states. This chapter will follow the definition of a hybrid system from Chapter 2 Def. 1.

3.4.2 Perturbation Analysis and the Saltation Matrix

Within a single mode, the Jacobian of the continuous dynamics can be used to update the covariance of a distribution. However, at hybrid events, the same method cannot be used. In order to properly update covariance through a mode transition, the time to impact of perturbations must be considered. The saltation matrix includes the time to impact for covariance updates [14, 22, 16, 19]:

$$\boxed{\Xi_x := D_x R_{I,J} + \frac{(f_J - D_x R_{I,J} f_I - D_t R_{I,J}) D_x g_{I,J}}{D_x g_{I,J} f_I + D_t g_{I,J}}} \tag{3.1}$$

where D_x and D_t represent Jacobians with respect to state and time, f_I is the linearization of the vector field F_I at the point of impact. The saltation matrix captures both the effects of the Jacobian of the reset map and variations in the time a system is acted upon by the dynamics of each mode. It maps pre-transition variations δx^- to post-transition variations δx^+ as, $\delta x^+ = \Xi_x \delta x^-$, and, by extension [28, 7], maps pre-transition covariance in state Σ_x^- to post-transition covariance Σ_x^+ as:

$$\boxed{\Sigma_x^+ = \Xi_x \Sigma_x^- \Xi_x^T} \tag{3.2}$$

In order to ensure the saltation matrix is well defined, we utilize the conventional assumptions from [44, Assumption 1], which most notably includes that transitions are transverse. This ensures that trajectories in a neighborhood of a guard must transition exactly once at small timescales. This assumption excludes Zeno behavior from this analysis. These assumptions are discussed in more detail in Sec. 2.1. Additionally, we make the assumption that the vector fields in each mode are extensible beyond the nominal guard as is done in [43]. This assumption allows us to analyze the effect of each mode's dynamics on a trajectory through a guard that is not at the nominal position.

More detailed discussion on the saltation matrix and its derivation are provided in Chapter 2.

3.5 Modeling Uncertainty in Guards and Reset Maps

A notable limitation of the saltation matrix formulation is that it assumes perfect knowledge of the structure of the hybrid system. However, in real applications, the guard boundaries and reset properties will be uncertain. This work seeks to capture these uncertainties for covariance propagation in hybrid systems with uncertainty in the guard (Sec. 3.5.1) and reset (Sec. 3.5.2).

3.5.1 Uncertainty in Guard Location

When the location of the guard is unknown along its normal direction, the post-transition state uncertainty is higher than if the guard location is perfectly known. The effects of uncertainty along the guard normal direction can be captured as an additional rank 1 update to the standard saltation matrix. Note that the effects of uncertainty in the normal direction of the guard surface are discussed in (Sec. 3.5.2), as the timing elements of uncertainty in normal are higher order terms that do not show up for a first order approximation, similar to the curvature of the guard surface. This section goes through the mathematical derivation of the guard-uncertainty saltation matrix, using a geometric derivation similar to the saltation matrix derivation in [16].

Here we consider how pre-impact displacements $\delta x(t^-)$ in mode I map to post-impact displacements $\delta x(\tilde{t}^+)$ in mode J. To do so, we start by considering how the nominal state $x(t)$ and a perturbed state $\tilde{x}(t)$ evolve from the time the nominal state reaches the guard before transitioning (t^-) to the time the perturbed state enters the new mode $(\tilde{t}^+)$. For simplicity of notation, in this derivation we assume that the nominal trajectory reaches the guard first, however, the same result is

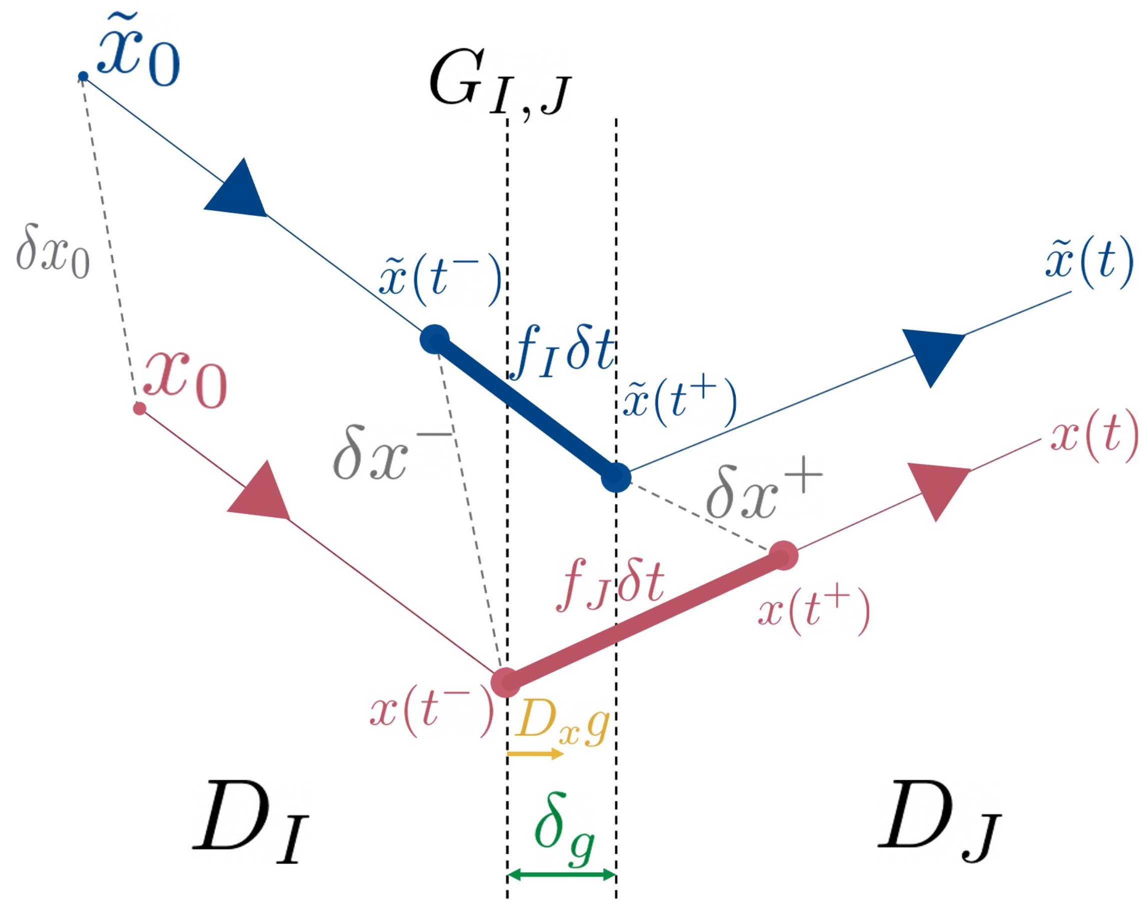


Figure 3.2: Uncertainty in guard location changing the relative position of variations before and after transition. For ease of depiction, the reset map is shown as identity, but accounting for them is an additive term in the saltation matrix. This is the basis of the guard saltation matrix.

reached if the opposite assumption is chosen. The displacements can be expanded as:

$$\delta x(t^-) = \tilde{x}(t^-) - x(t^-) \tag{3.3}$$

$$\delta x(\tilde{t}^+) = \tilde{x}(\tilde{t}^+) - x(\tilde{t}^+) \tag{3.4}$$

For readability, hereafter we use the + and - superscripts to represent the state at times $\tilde{t}^+$ and t^-, respectively, e.g. $x^+ = x(\tilde{t}^+)$. The evolution of these trajectories can be seen in Fig. 3.2.

We would like to solve for δx^+ as a function of δx^- and known system parameters by using the continuous and reset dynamics. For this first-order analysis we linearize the pre-impact dynamics F_I about the point $x(t^-)$ as f_I, and similarly we linearize the post-impact dynamics F_J about the point $x(t^+)$ as f_J. Following the linearized hybrid dynamics forward from time t^-, and assuming without loss of generality that the nominal trajectory impacts the guard first at time t^-, we have:

$$x^+ = R_{I,J}(x^-, t^-) + f_J \delta t \tag{3.5}$$

that is, the final state is equal to the initial state passed through the reset map and then following the dynamics of the new mode for time $\delta t = \tilde{t} - t$, the time between impact events, until time $\tilde{t}^+$. Similarly, we can follow the hybrid dynamics forward from $\tilde{x}^-$ to get:

$$\tilde{x}^+ = R_{I,J}(\tilde{x}^- + f_I \delta t, \tilde{t}^-) \tag{3.6}$$

where in this case the perturbed state first flows in the prior mode until time $\tilde{t}^-$ and then passes through the reset. Using the definition of δx^- and δt, (3.6) can be written entirely in terms of the pre-impact variation and times:

$$\tilde{x}^+ = R_{I,J}(x^- + \delta x^- + f_I \delta t, t^- + \delta t) \tag{3.7}$$

Substituting (3.5) and (3.7) back into (3.4) yields:

$$\delta x^+ = R_{I,J}(x^- + \delta x^- + f_I \delta t, t^- + \delta t) - R_{I,J}(x^-, t^-) - f_J \delta t \tag{3.8}$$

Using the first-order Taylor series expansion of the reset map about (x^-, t^-), we can replace this expression with:

$$\begin{aligned}\delta x^+ =& R_{I,J}(x^-, t^-) + D_x R_{I,J} \delta x^- + D_x R_{I,J} f_I \delta t \\ &+ D_t R_{I,J} \delta t - R_{I,J}(x^-, t^-) - f_J \delta t \qquad (3.9) \\ =& D_x R_{I,J} \delta x^- + \left(D_x R_{I,J} f_I + D_t R_{I,J} - f_J\right) \delta t \qquad (3.10)\end{aligned}$$

The next step is to determine what δt is in terms of δx^- and system parameters by examining the dynamics of the first mode along the guard normal direction, $D_x g$:

$$(D_x g f_I + D_t g)\delta t = -D_x g \delta x^- + \delta_g \tag{3.11}$$

$$\delta t = \frac{-D_x g \delta x^- + \delta_g}{D_x g f_I + D_t g} \tag{3.12}$$

where δ_g is the perturbation in the guard location along its normal direction, which holds whether the guard occurs early or late. Effectively, this means that time is equal to distance divided by velocity on infinitesimal perturbations. Note that this additional perturbation δ_g is the key difference compared to the derivation of the traditional saltation matrix. Additionally, it should be noted that this requires the extra assumption that the dynamics are extensible beyond the nominal guard surfaces.

Plugging (3.12) into (3.10) results in the following expression:

$$\begin{aligned}\delta x^+ = &\left(D_x R_{I,J} + \frac{(f_J - D_x R_{I,J} f_I - D_t R_{I,J}) D_x g}{D_x g f_I + D_t g}\right)\delta x^- \\ &+ \left(\frac{D_x R_{I,J} f_I + D_t R_{I,J} - f_J}{D_x g f_I + D_t g}\right)\delta_g \qquad (3.13)\\ :=& \Xi_x \delta x^- + \Xi_g \delta_g \qquad (3.14)\end{aligned}$$

where Ξ_x is the traditional saltation matrix, (3.1), and Ξ_g is a "guard saltation matrix", defined as:

$$\boxed{\Xi_g := \frac{D_x R_{I,J} f_I + D_t R_{I,J} - f_J}{D_x g f_I + D_t g}} \tag{3.15}$$

Note that $\Xi_x = D_x R_{I,J} - \Xi_g D_x g$, and so Ξ_g can be computed as part of computing Ξ_x. The guard saltation matrix is a single column vector if δ_g is only the normal direction component of guard uncertainty[1]. Note also that while (3.15) was derived assuming the nominal transitions first, the

[1] For a full dimensional guard uncertainty vector δg^-, we have $\delta_g = D_x g \delta g^-$, and can use the matrix $(\Xi_g D_x g)$ as $\delta x^+ = \Xi_x \delta x^- + (\Xi_g D_x g)\delta g^-$, however note that only the uncertainty along the normal direction affects the outcome.

same expression is obtained if the perturbed trajectory is assumed to transition first.

This can be used as an extended saltation matrix, which we call $\hat{\Xi}$,

$$\begin{bmatrix} \delta x^+ \\ \delta_g \end{bmatrix} = \begin{bmatrix} \Xi_x & \Xi_g \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x^- \\ \delta_g \end{bmatrix} = \hat{\Xi} \begin{bmatrix} \delta x^- \\ \delta_g \end{bmatrix} \tag{3.16}$$

Extending now to covariance, as in (3.2), when using this saltation matrix with no prior known covariance between the guard and state, the covariance updates for both the state and the guard are:

$$\begin{bmatrix} \Sigma_x^+ & \Sigma_{xg}^+ \\ \Sigma_{gx}^+ & \Sigma_g^+ \end{bmatrix} = \hat{\Xi} \begin{bmatrix} \Sigma_x^- & 0 \\ 0 & \Sigma_g^- \end{bmatrix} \hat{\Xi}^T \tag{3.17}$$

where Σ_x is the state covariance and Σ_g is the guard covariance. Pulling out the state covariance through an uncertain guard, we get:

$$\boxed{\Sigma_x^+ = \Xi_x \Sigma_x^- \Xi_x^T + \Xi_g \Sigma_g^- \Xi_g^T} \tag{3.18}$$

The improved covariance estimation of the guard saltation matrix is shown in Fig. 3.1b. Note that the covariance of the uncertain guard distribution is equivalent to the sum of the covariance of propagating the initial distribution through the certain guard and the resulting covariance of propagating a known starting condition through an uncertain guard. For this trial, the K-L divergence [45] (which is a measure of the difference between two probability distributions) between the actual covariance and the estimated is reduced from 402 to 0.03 by including the guard uncertainty propagation terms. The final error between the true and the estimated covariance from the guard saltation matrix is caused by the linearization of the dynamics.

There are trade-offs between using the extended matrix (3.17) and directly updating the state covariance (3.18). Using the extended matrix requires extending the state and increases the dimensionality of all terms, but it allows for past measurements to affect knowledge of the guard distribution. Directly updating the state distribution allows for a simpler state vector, but assumes that the guard distribution is static. If repeated behavior near the same region of a guard is expected, then it would be beneficial to try to estimate the guard parameters. However, in situations like locomotion on uneven ground where the system is not expected to re-traverse the same areas frequently, it makes sense to accept the mean and covariance as fixed (or calculated separately, e.g. based on exteroceptive sensor noise) parameters for the guard surface.

3.5.2 Uncertainty in Reset Parameters

Even in cases where the location of the guard is perfectly known, the exact properties of the reset map may not be known. Some physical examples of this include the coefficient of restitution in elastic systems and the precise surface normal in any contact system. These types of uncertainties can be handled by parameterizing the reset maps to be not only functions of state and time, but to also include other parameters. By including these additional parameters, the resulting uncertainty in state caused by variations in reset parameters can be examined through the Jacobian with respect to these additional parameters.

For this derivation, we re-define the reset map $R(x)$ to include its “fixed” parameters as arguments $R(x, \theta)$. This θ term can include values like the coefficient of restitution in the bouncing ball problem. Using this formulation, we can examine how uncertainty in model parameters can affect the resulting state estimation covariance after an impact, updating (3.5) and (3.6):

$$x^+ = R_{I,J}(x^-, \theta) + f_J \delta t \tag{3.19}$$

$$\tilde{x}^+ = R_{I,J}(\tilde{x}^- + f_I \delta t, \tilde{\theta}) \tag{3.20}$$

Taking the difference between these two to find δx^+:

$$\delta x^+ = R_{I,J}(\tilde{x}^- + f_I \delta t, \tilde{\theta}) - (R_{I,J}(x^-, \theta) + f_J \delta t) \tag{3.21}$$

Now using first order approximations of the reset map with the Taylor expansion, as in (3.10):

$$\begin{aligned}\delta x^+ =& R_{I,J}(x^-,\theta) + D_x R_{I,J}\delta x^- + D_x R_{I,J} f_I \delta t + D_\theta R_{I,J}\delta\theta + D_t R_{I,J}(x^-,\theta)\delta t \\ &- R_{I,J}(x^-,\theta) - f_J\delta t \qquad (3.22)\\ =& D_x R_{I,J}\delta x^- + \left(D_x R_{I,J} f_I + D_t R_{I,J} - f_J\right)\delta t + D_\theta R_{I,J}\delta\theta \qquad (3.23)\end{aligned}$$

where equality holds to first order.

By using δt from (3.12), and rearranging into a block matrix, this becomes:

$$\begin{bmatrix}\delta x^+ \\ \delta\theta^+\end{bmatrix} = \begin{bmatrix}\Xi_x & D_\theta R_{I,J} \\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta x^- \\ \delta\theta^-\end{bmatrix} \qquad (3.24)$$

Assuming there is no initial covariance between the state and the reset map information, this can be used to update the state covariance with:

$$\boxed{\Sigma_x^+ = \Xi_x\Sigma_x^-\Xi_x^T + (D_\theta R_{I,J})\Sigma_\theta^-(D_\theta R_{I,J})^T} \qquad (3.25)$$

where Σ_θ^- is the covariance of the θ parameters. Note that, as was the case with uncertain guard, the covariance of the uncertain reset distribution is equivalent to the sum of the covariance of propagating the initial distribution through the certain reset and the resulting covariance of propagating a known starting condition through an uncertain reset.

This expression can be used in combination with the uncertainty in guard location for a total covariance update:

$$\boxed{\Sigma_x^+ = \Xi_x\Sigma_x^-\Xi_x^T + \Xi_g\Sigma_g^-\Xi_g^T + (D_\theta R)\Sigma_\theta(D_\theta R)^T} \qquad (3.26)$$

Results demonstrating the improvement over assuming perfect knowledge of the reset map in uncertainty propagation can be found in Fig. 3.1c. For this trial, the K-L divergence between the actual covariance and the estimated is reduced from 357.6 to 19.8 by including the reset uncertainty propagation term. Furthermore, combining both guard and reset uncertainty, Fig. 3.1d, the K-L divergence between the actual covariance and the estimated is reduced from 739 to 0.03. The final error between the true and the estimated covariance is caused by the linearization of the dynamics.

3.6 Kalman Filtering with Uncertain Environment

This section applies the covariance update rules found in the prior section to the problem of state estimation using Kalman filtering (summarized briefly in Sec. 3.6.1). With these more accurate distribution updates through hybrid events, we can achieve better estimation accuracy. The implementation details for this "uncertainty aware SKF" (uaSKF) follow the algorithm for SKF, presented in [7]. The key differences from the SKF occur during hybrid transition events. We discuss how to handle these events in both the process and measurement updates in Sec. 3.6.2 and Sec. 3.6.3, respectively.

3.6.1 Kalman Filtering in the Smooth Domains

While the system is not interacting with any guard surfaces, the uaSKF behaves as a standard KF or EKF would. In these cases, the system follows the standard update rules. The standard KF updates the mean $\hat{x}$ and covariance $\hat{\Sigma}$ estimate in two steps [26, Eqns. 1.9–1.13]: first, the *a priori* update:

$$\hat{x}(k+1|k) = A_{I,\Delta}\hat{x}(k) \tag{3.27}$$

$$\hat{\Sigma}(k+1|k) = A_{I,\Delta}\hat{\Sigma}(k)A_{I,\Delta}^T + W_{I,\Delta} \tag{3.28}$$

at timestep $k+1$, where $A_{I,\Delta}$ is the discrete dynamic matrix for F_I, Δ is the discretization timestep, and $W_{I,\Delta}$ is the covariance of the additive Gaussian process noise. Second, the *a posteriori* update:

$$K_{k+1} = \hat{\Sigma}(k+1|k)C_I^T\left[C_I\hat{\Sigma}(k+1|k)C_I^T + V_I\right]^{-1} \tag{3.29}$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K_{k+1}\left[y(k+1) - C_I\hat{x}(k+1|k)\right] \tag{3.30}$$

$$\hat{\Sigma}(k+1|k+1) = \hat{\Sigma}(k+1|k) - K_{k+1}C_I\hat{\Sigma}(k+1|k) \tag{3.31}$$

where K_{k+1} is the Kalman gain, C_I is the measurement function, $y(k+1)$ is the measurement, and V_I is the covariance of the additive Gaussian measurement noise.

While the standard Kalman update works well during the continuous domains, additional consideration has to be taken for hybrid events. We will now present how to handle hybrid transitions in both the *a priori* and *a posteriori* updates.

3.6.2 Uncertainty Aware Hybrid A Priori Updates

In a discretized KF, the hybrid event will most likely not take place perfectly at the time steps. Accordingly, the *a priori* update must include the first mode dynamics, the discrete update, and the second mode dynamics to bridge the time from one sample time to the next. Conceptually, what happens here is three separate updates (or more if multiple hybrid transitions occur in a single timestep) combined into one, with the first and final portions following (3.27)–(3.28) with timesteps Δ_1 and Δ_2, respectively. This does not require knowledge of the number of impact events in the system as these updates are driven by the mean estimate reaching guards, not by ground truth impacts.

The instantaneous mean and covariance updates during the discrete transition from mode I to J use the following update rules, using (3.26) as in [7, Eqns. 16–17][2]:

$$x_J(t) = R_{I,J}(x_I(t)) \tag{3.32}$$

$$\Sigma_J(t) = \Xi_x \Sigma_I(t) \Xi_x^T + \Xi_g \Sigma_g \Xi_g^T + (D_\theta R_{I,J}) \Sigma_\theta (D_\theta R_{I,J})^T \tag{3.33}$$

By combining this instantaneous update with the standard KF updates (3.27)–(3.28) in the first and second modes, as in [7, Eqns. 18–19], we arrive at the following update rule:

$$\hat{x}(k+1|k) = A_{J,\Delta_2} R_{I,J}(A_{I,\Delta_1} \hat{x}(k)) \tag{3.34}$$

$$\hat{\Sigma}(k+1|k) = A_{J,\Delta_2} \Big(\Xi_x (A_{I,\Delta_1} \Sigma(k) A_{I,\Delta_1}^T + W_{I,\Delta_1}) \Xi_x^T$$

[2] In the notation of [7, Eqn. 17], we are setting $W_{R_{(I,J)}} = \Xi_g \Sigma_g \Xi_g^T + D_\theta R \Sigma_\theta D_\theta R^T$.

$$+ \Xi_g \Sigma_g \Xi_g^T + D_\theta R \Sigma_\theta D_\theta R^T \Big) A_{J,\Delta_2}^T + W_{J,\Delta_2} \tag{3.35}$$

3.6.3 Uncertainty Aware Hybrid A Posteriori Updates

When a measurement update causes the mean estimate to meet a guard condition, the uncertainty aware SKF applies the discrete update to the a posteriori mean and covariance estimates following equations (3.32)–(3.33).

3.7 Kalman Filtering Performance on Example Systems

To demonstrate the increased accuracy of these methods on systems with uncertainty in the guard conditions and reset dynamics, we provide filtering results from a variety of systems. For each system, a comparison is made against the standard SKF formulation without incorporating knowledge of guard uncertainty using the sign test for the median difference between trials [46], as used in [7].

3.7.1 Elastic Bouncing Ball

Consider a simple point mass in two dimensions (x_1, x_2), elastically impacting a plane with uncertain height δ_g, angle θ, and coefficient of restitution α. An example of propagating a distribution through this system can be seen in Fig. 3.1. This system has 1 mode with a self reset. The continuous dynamics are:

$$\dot{x} = \Big[\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4\Big]^T = \Big[x_3, x_4, 0, -a_g\Big]^T \tag{3.36}$$

where a_g is the acceleration due to gravity.

The guard for this system is:

$$g(x) = x_2 \cos\theta - x_1 \sin\theta - \delta_g \tag{3.37}$$

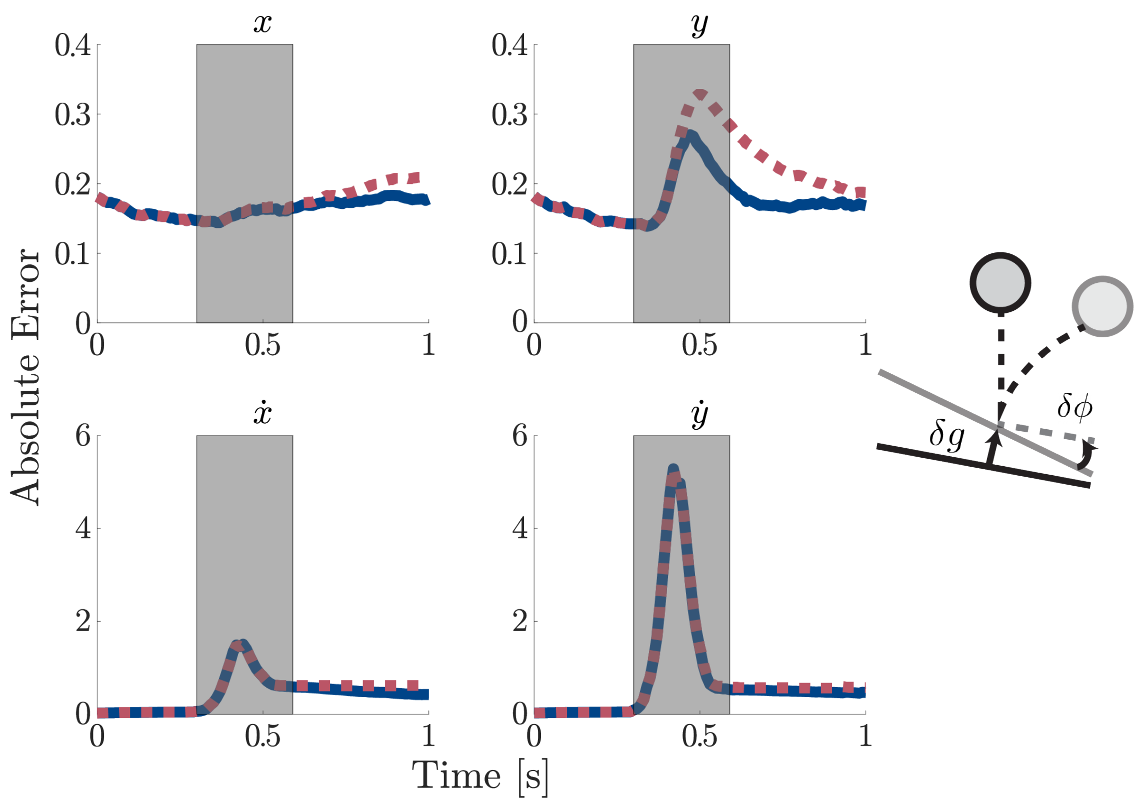


Figure 3.3: Average error for 1000 trials of Kalman filtering on a bouncing ball model with uncertain parameters. The standard SKF results are in red dashed lines and the uncertainty aware SKF results are in blue solid lines. The shading represents the time from the earliest to latest ground truth impacts. A sketch of the system and a typical trajectory is shown to the right.

The reset dynamics for the system are:

$$x^+ = \begin{bmatrix} x_1^- \\ x_2^- \\ x_3^- + \sin\theta(1+\alpha)(x_4^- \cos\theta - x_3^- \sin\theta) \\ x_4^- - \cos\theta(1+\alpha)(x_4^- \cos\theta - x_3^- \sin\theta) \end{bmatrix} \tag{3.38}$$

We simulated 1000 trials of this system running for 1 second from a height of 3 m with initial velocity of -5 m/s in the vertical direction and initial covariances of $0.05 I_2$ for position and $0.001 I_2$ for velocity, where I_2 is the 2×2 identity matrix. For this test, the mean ground height was 0 with a standard deviation of 0.25 m, the mean angle of the ground was -0.25 radians with a standard deviation of 0.05 radians, and the coefficient of restitution was 0.8. The process noise for this system was $W_p = 10I_2$ for position and $W_v = I_2$ for velocities. The system measured only positions with covariance $V = I_2$.

The results of the elastic ball test are shown in Fig. 3.3. The uncertainty aware method had a lower error than the standard saltation method with $p < 0.005$. The median improvement in MSE over complete 1 s trajectories is 0.6%. Additionally, the maximum percent improvement in average error magnitude for a timestep is 24% at 0.98s. The estimation is improved in all dimensions, but most notably in the vertical direction because the uncertain guard component accounts for the extra uncertainty in the vertical direction, while the uncertain reset term arising from the uncertain guard normal adds uncertainty in both velocity terms, which propagates to position as the filter progresses.

3.7.2 Point Mass with Plastic Impact and Separation Conditions

To demonstrate the method on a system with plastic impact and a constrained mode, we examine the case of dropping a plastic ball onto a circle of known center but unknown radius. The system consists of two modes, an unconstrained aerial mode with the same dynamics as the elastic ball example, and a constrained mode when the ball is sliding on the surface. The ballistic dynamics of this system are the same as the dynamics for the prior elastic ball example. The impact reset map for this system is also the same as (3.38), however, the angle of the guard is determined by the tangent plane to the circle at the point of impact and the coefficient of restitution α is zero. The constrained dynamics enforce zero acceleration along the radius extending from the center of the circle to the current position. A liftoff event is triggered with an identity reset map when the force required to maintain the acceleration constraint becomes attractive.

We tested this system by running 1000 trials for 3 seconds each with initial position of (0.5,5)m, zero initial velocity, and position and velocity covariance of $0.1I_2$ each. The nominal surface is a circle with radius 2m centered about the origin. The process noise for this system is $W_p = 0.1I_2$ for position and $W_v = 0.01I_2$ for velocity. The system measures positions with covariance $V = 0.1I_2$.

Results for this test can be seen in Fig. 3.4. On average, utilizing awareness of model uncertainties notably improves performance on position estimates, but since the dynamics are largely the

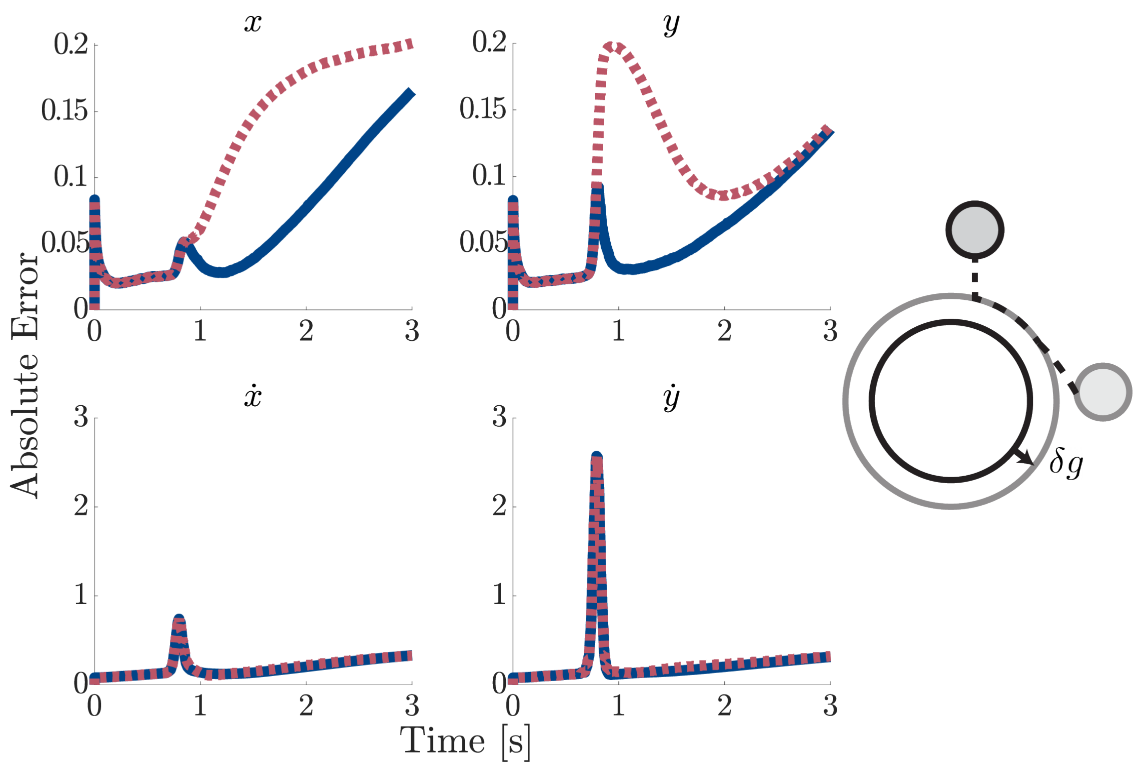


Figure 3.4: Average error of 1000 trials on a point mass plastically impacting a circle of unknown radius. The mean radius is 2m with a standard deviation of 0.25m. SKF results are red dashed lines and the uncertainty aware SKF results are blue solid lines. A sketch of the system and a typical trajectory is shown to the right.

same regardless of minor position variations, the velocity estimates are very similar. The uncertainty aware method outperforms the standard saltation method with $p < 0.005$. The median improvement in MSE over complete 3 s trajectories is 3.6%. Additionally, the maximum percent improvement in average error magnitude for a timestep is 35% at 0.93s.

3.7.3 ASLIP Hopper System

Finally, we demonstrate the efficacy of this method on a more complicated robot model, the ASLIP (asymmetric spring loaded inverted pendulum) hopper [47], which is shown on the right in Figure 3.5. The state for the ASLIP hopper is defined as:

$$q = [x_b, y_b, \theta_b, x_t, y_t, \dot{x}_b, \dot{y}_b, \dot{\theta}_b]^T \tag{3.39}$$

In the aerial phase, the system uses ballistic dynamics with a fixed leg length and angle. We

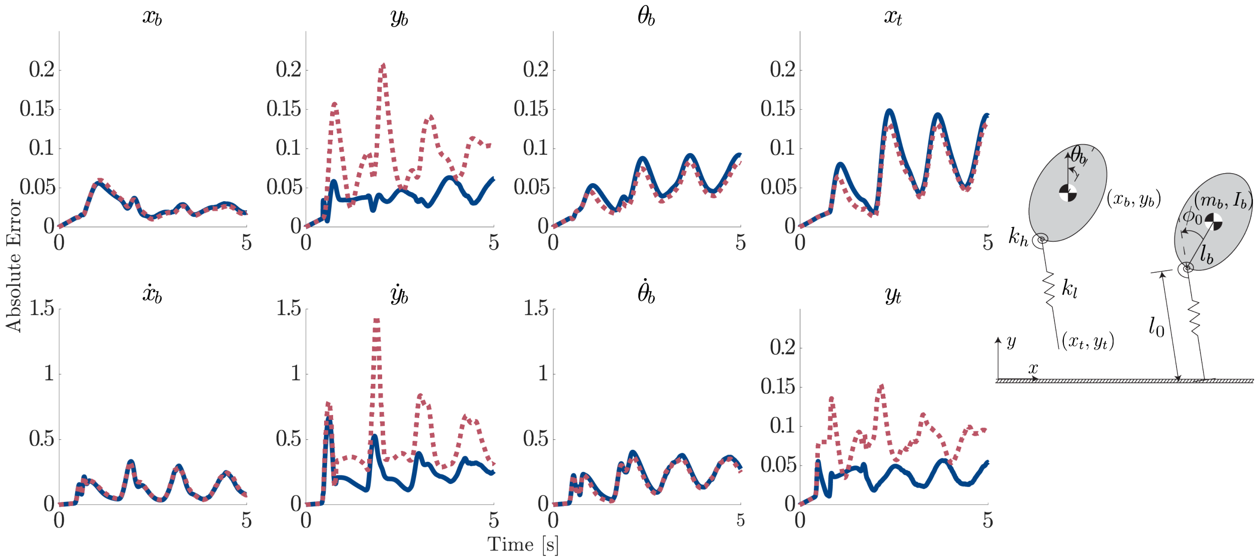


Figure 3.5: Left: Average error for 1000 trials on an ASLIP hopper with unknown ground height. The mean ground height was 0 with a standard deviation of 0.01m. The standard SKF results are in red dashed lines and the uncertainty aware SKF results are in blue solid lines. Note the improvement primarily in the vertical (y) directions that are aligned with the direction of ground uncertainty. Right: The model used for these experiments.

used Lagrangian dynamics to generate the dynamic equations for the contact mode. The guard for touchdown is based on the height of the toe reaching the ground, which is uncertain. The guard for liftoff is based on the leg returning to rest length. The complete derivation of these dynamics is given in [7].

For this example, 1000 trials were run for 5 seconds, which is approximately 4 hops depending on the sampled conditions, with system parameters:

$$m_b = 1\text{kg}, a_g = 9.8\text{m/s}^2, l_b = 0.5\text{m}, I_b = 1\text{kgm}^2, \tag{3.40}$$
$$k_h = 100\text{nm/rad}, k_l = 100\text{n/m}, l_0 = 1\text{m}, \phi_0 = 0\text{rad}$$

The initial state of the system is $y_b = 2.5\text{m}, x_b = 0\text{m}, \theta_b = 0\text{rad}, x_t = 0\text{m}, y_t = 1\text{m}$ with zero initial velocity. The covariance of the initial positions was $10^{-6}I_5$ and the velocity covariance was $10^{-6}I_3$. The process noise is $W = 0.001I_8$ and the system measures all positions with covariance $V = 0.01I_5$. The height of the ground is uncertain with zero mean and a standard deviation of 0.01m.

Results for this system can be seen in Fig. 3.5. The uncertainty aware method primarily improved results in the y direction for body position and velocity and toe position, as the guard saltation matrix primarily operates in the vertical direction due to the ground normal being vertical, but it propagates variations to other states through the dynamics outside of the touchdown event. The uncertainty aware method outperforms the standard saltation method with $p < 0.005$. The median improvement in MSE over complete five second trajectories is 54.7%. Additionally, the maximum percent improvement in average error magnitude for a timestep is 60.1% at 1.9s.

3.8 Conclusion and Future Work

In this work we derive a first order propagation law for guard and reset uncertainty by using the guard saltation matrix and a Jacobian of the reset map. This uncertainty aware method outperforms the standard saltation matrix method at estimating resulting probability distributions simulated through uncertain hybrid guards and reset maps. We then use this propagation law in the uncertainty aware SKF and achieved lower estimation error than the standard SKF. The uncertainty aware SKF reduces the average estimation error overall, with up to 24-60% improvement after impact events.

While these results aid in updating the covariance once a transition is believed to have occurred, there is still the problem of determining whether mode transitions have occurred. Future work will extend these results to include explicit reasoning about which mode the system is in. One potential approach for reasoning about whether mode transitions have occurred is to modify multiple model estimation methods [38, 39, 40] to account for variable transition probabilities based on the estimated position relative to guard locations.

Additionally, this work handles single mode transitions and multiple mode transitions with known sequences. This can be extended to reasoning about simultaneous (or near simultaneous) contact with the Bouligand derivative [20, 19], to encode multiple potential contact sequences.

Chapter 4

Hybrid Iterative Linear Quadratic Estimation: Optimal Estimation for Hybrid Systems

4.1 Abstract

In this chapter we present Hybrid iterative Linear Quadratic Estimation (HiLQE), an optimization based offline state estimation algorithm for hybrid dynamical systems. We utilize the saltation matrix, a first order approximation of the variational update through an event driven hybrid transition, to calculate gradient information through hybrid events in the backward pass of an iterative linear quadratic optimization over state estimates. This enables accurate computation of the value function approximation at each timestep. Additionally, the forward pass in the iterative algorithm is augmented with hybrid dynamics in the rollout. A reference extension method is used to account

for varying impact times when comparing states for the feedback gain in noise calculation. The proposed method is demonstrated on a simple two dimensional bouncing ball system with position measurements as well as an ASLIP hopper system with position measurements. In comparison to the Salted Kalman Filter (SKF), the algorithm presented here achieves a maximum of 62.25% reduction in estimation error magnitude over all state dimensions near impact events.

4.2 Introduction

Contact is essential for many robots as they must physically interact with their environment to accomplish their goals. For example, legged robots must repeatedly make and break contact with the ground to move around and perform their tasks, whether that is mapping, surveying, search and rescue, or any other task. Similarly, manipulation robots must make contact with the objects they seek to manipulate and their environment in order to push, pull, grasp, etc., those objects. To plan reliable control sequences through contact, robots need an accurate state estimate. However, intermittent contacts cause robots' dynamics to become non-smooth and potentially discontinuous, breaking the assumptions of classical methods of state estimation which assume smoothness [31, 32, 33, 34, 48].

Accurate state estimation is especially critical near contact surfaces as minor configuration variations can cause major changes in control authority. A key example is when a foot is in contact with the ground, that leg can exert large ground reaction forces on the robot, but if that same foot is even slightly off the ground, then it can exert no force. Accordingly, for controllers to generate feasible plans, a strong estimate of current state is necessary. This is typically approached as a filtering problem using either Kalman filters [31, 34, 7, 35, 8] or particle filters [41, 42].

There are many situations where full state ground truth cannot be obtained to evaluate the

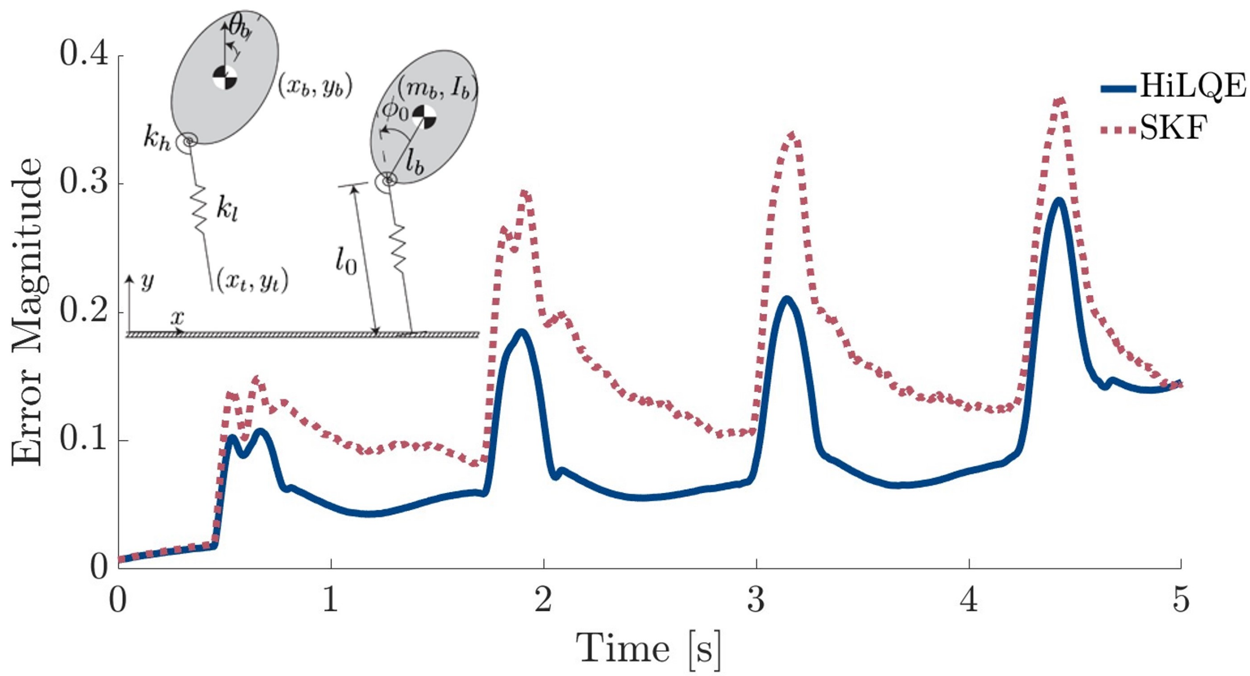


Figure 4.1: Average error magnitude over 100 trials when simulating a an ASLIP system which impacts the ground four times during its execution. The error from the proposed HiLQE method is shown as a solid blue line while the error from the SKF is shown as a dashed red line.

performance of the robot or tuning of these algorithms. In these cases, offline methods for state estimation [49, 48] can provide a log of the robot's behavior as well as a comparison point for estimation performance when tuning gains and designing online estimation algorithms. These offline methods can provide better performance as they consider all of the information from a given trial to achieve higher state estimation accuracy at each timestep. These frameworks can also be used to better estimate dynamic parameters of a system given the additional information.

To this end, here we propose Hybrid iterative Linear Quadratic Estimation (HiLQE), an offline iterative linear quadratic state estimation scheme for hybrid systems which optimizes over all sensor readings to achieve a more accurate state estimate. Similar to the Hybrid iterative Linear Quadratic Regulator (HiLQR) for control [30, 50], we use the saltation matrix [6] in the backward pass of the iterative update in order to properly capture the gradients through impact events. Additionally, we explicitly account for the differing hybrid system discrete modes [11, 12, 13, 5], utilizing the proper system dynamics for each mode during both the backward and forward passes in the smooth domains. In the forward pass, we utilize a reference extension method [30, 50] to create reasonable comparisons between states in differing domains. We demonstrate the efficacy of these methods on a simple two dimensional bouncing ball system as well as an asymmetric spring loaded inverted pendulum (ASLIP) system, which can be seen in Fig. 4.1. With these updated techniques for hybrid transitions, HiLQE notably outperforms the online, incremental Salted Kalman Filter (SKF) method.

4.3 Related Work

As the fields of hybrid systems [11, 12, 13, 5] and state estimation [51, 52] are relatively mature and have plenty of relevant work, here we focus specifically on work covering the intersection of the two.

4.3.1 State Estimation for Hybrid Systems

Initial works on Kalman filter (KF) based methods for hybrid dynamical systems such as [31, 34] utilize the reset map to update the mean estimate and the Jacobian of the reset map to update covariance beliefs at hybrid events. However, recent work demonstrated that the Jacobian does not capture all of the effects of the hybrid event on reshaping the covariance. Instead, [7] proposed the Salted Kalman Filter, which uses the saltation matrix [6] in place of the Jacobian of the reset map at hybrid events for Kalman filtering. The saltation matrix includes a rank 1 update to the Jacobian that accounts for state variations caused by time-to-impact variations, and is discussed in more detail below. This SKF will be the baseline against which we compare our accuracy in this work. This work was extended to include filtering on manifolds in [35] with the Hybrid Invariant Extended Kalman Filter (HInEKF) and to include uncertainty in guard conditions and reset maps in [8] with the uncertainty aware Salted Kalman Filter (ua-SKF). In this chapter, the saltation matrix is used to determine gradient information in an optimization problem over a series of measurements.

Other work on hybrid system state estimation has largely involved multiple estimators in order to track belief in multiple hybrid modes. For example, the Interacting Multiple Model estimation algorithm (IMM) [38] maintains KFs for each of the hybrid modes. Multiple model methods have been extended to a variety of problems including nonlinear dynamics [39] and non-identity rests [40]. Multiple model methods are not easily applicable on event-driven hybrid dynamical systems as one of the core assumptions of these methods is that the transitions between discrete modes follow a Markov model, which is not necessarily true when the probabilities of discrete state transitions are dependent on the continuous state beliefs.

Alternatively, many methods such as [41, 42] have adopted particle filtering approaches and use large numbers of individual estimates to represent a distribution as opposed to summary statistics like mean and covariance in the case of KFs. While these methods have many benefits, including capturing nonlinear dynamics and non-Gaussian beliefs, the computational complexity of running a particle filter is far greater than that of parametric filters which we seek to tune using our offline method.

In [49], an offline optimization based state estimation method was proposed. This work relaxed the discrete state estimates to a continuous state in order to estimate both the continuous and discrete portions of the state simultaneously. To handle the hybrid events in this continuous formulation, the model of the process noise was switched from a Gaussian distribution to a long tailed student's t distribution. In contrast, the work we present in this chapter explicitly considers the hybrid nature of the systems and accounts for non-identity reset maps directly. By explicitly considering the dynamics in an iterative formulation, the estimates generated by our method will always be physically realizable, which could be useful in a potential extension to an online moving window approach.

4.3.2 Nonlinear Event Mapping and Saltation Matrices

The saltation matrix [14, 22, 16, 19, 6] is used to map perturbations through non-smooth dynamics at transition events between discrete modes. In contrast to the Jacobian of the reset map at a hybrid event, the saltation matrix captures to first order the effects of variations in dwell time in each hybrid mode caused by displacements in state. The difference between the two methods of propagation through hybrid events can be seen in Fig. 2.3 in Sec. 2.3. In the context of the work we present here, the saltation matrix will be used to capture gradient information through hybrid events.

Previously, [28] demonstrated the saltation matrix can be used to map probability distributions through hybrid transitions in a quadratic formulation. In this work we will use this to propagate quadratic cost functions backwards in an iterative linear quadratic estimation context.

4.3.3 Iterative Linear Quadratic Methods for Hybrid Systems

In recent work, Bailly presented the MAPE-DDP algorithm [48], which uses a DDP/iLQ framework to estimate the centroidal state of a legged robot. This is similar to prior work on smoothing filters such as the iterative extended Kalman smoother [53] or the Backward Smoothing Extended Kalman Filter [54]. A key difference between this approach for estimation and DDP or iLQR for path planning is that the Jacobian of the measurement function must be used in calculating cost gradients. While a legged robot is a hybrid system, the floating base model presented does not treat the robot as a hybrid system. In contrast, the work we present in this chapter explicitly considers the differing dynamics in each mode and the effects of mode transitions on gradient information. By accounting for hybrid dynamics, our formulation would allow estimation to use information from lower level control architecture which deals with legged control directly.

Another recent work solved the iLQR trajectory planning problem on hybrid systems [30, 50]. In that work the saltation matrix is used to calculate gradient information through mode transition events in the backward pass of an iLQR problem for trajectory planning. The work we present in this chapter can be viewed as the solution to the dual problem to the planning problem presented there. One of the key differences is that the estimation is looking backward in time to estimate a trajectory whereas the planning problem looks forward to generate a trajectory. Additionally, rather than a reference trajectory, the estimation problem we present in this work uses a series of past measurements, which may be in a lower dimensional space than the trajectory itself.

4.4 Background

4.4.1 Hybrid Dynamical Systems

A hybrid dynamical system is a system with continuous states, such as positions and velocities, and discrete states or modes, such as whether a specific limb is in contact with the ground. In a hybrid dynamical system, the sequence of discrete states is determined by the evolution of the continuous states. This chapter will follow the definition of a hybrid system from Chapter 2 Def. 1.

4.4.2 Perturbation Analysis and the Saltation Matrix

Within a single mode, the Jacobian of the continuous dynamics can be used to obtain gradient information. Analogously, the saltation matrix is used to obtain gradient information through hybrid events. The saltation matrix captures both the Jacobian of the reset map and the variations in time to impact as:

$$\Xi_{(\mathrm{I},\mathrm{J})} := \mathrm{D}_x R^- + \frac{\left(F_\mathrm{J}^+ - \mathrm{D}_x R^- F_\mathrm{I}^- - \mathrm{D}_t R^-\right) \mathrm{D}_x g^-}{\mathrm{D}_t g^- + \mathrm{D}_x g^- F_\mathrm{I}^-} \tag{4.1}$$

where F_I^- is the value of the first mode's vector field at the point of impact, F_J^+ is the value of the second mode's vector field at the nominal reset point, R^- is the reset map at the point of impact, and g^- is the guard function representing the boundary between the two modes. The saltation matrix captures both the effects of the Jacobian of the reset map and variations in the time a system is acted upon by the dynamics of each mode. It maps pre-transition variations δx^- to post-transition variations δx^+ as, $\delta x^+ = \Xi \delta x^-$, as can be seen in Fig. 2.3 in Sec. 2.3.

In order to ensure the saltation matrix is well defined, we utilize the conventional assumptions from [44, Assumption 2], which most notably include that the dynamics are transverse to guards and that Zeno behavior will not occur. This ensures that trajectories in a neighborhood of a guard must transition exactly once at small timescales, which is a useful simplifying assumption when crafting algorithms to operate on these systems.

4.4.3 Iterative Linear Quadratic Estimation

The iterative Linear Quadratic Regulator (iLQR) algorithm [55] is traditionally used to solve trajectory optimization problems. The same methods can be used to solve state estimation problems when properly formulated. The state estimation optimization problem is written as [48]:

$$\begin{aligned}
\min_{X_{0:N}, W_{0:N-1}} \; & \frac{1}{2}||x_0 - \bar{x}_0||^2_{P_x} \\
& + \sum_{i=1}^{N} \left(\frac{1}{2}||h(x_i) - y_i||^2_{P_v} + \frac{1}{2}||w_{i-1}||^2_{P_w} \right) \\
s.t. \quad & x_{i+1} = f(x_i, u_i) + w_i \forall i
\end{aligned} \tag{4.2}$$

Where X and W are the vectors of states and process noises, x_i and w_i, for the full trajectory, $\bar{x}_0$ is the nominal initial state, $h(x)$ is a measurement function from the full state to the measurement space, y_i is a measurement taken at timestep i, and P_v,P_w,P_x are the inverses of the measurement noise, process noise, and arrival cost covariances respectively. In this formulation, $||x||^2_P$ represents the squared weighted norm $x^T Px$. This form of cost function is known as the Mahalanobis distance, which penalizes deviations in directions with more certainty more harshly than deviations in directions with low certainty.

Intuitively, this optimization simultaneously seeks to minimize process noise and deviations from measurements and the initial state while ensuring the dynamics are satisfied.

The iLQR algorithm seeks to solve this optimization problem by examining it in a series of backward-forward passes through a trajectory using the principle of optimality to separate the problem into a series of smaller optimal estimation problems. This backward and forward iterative process is repeated until either some convergence property is met, such as the decrease in cost of a step falling below some threshold, or until a limit of iterations is reached.

Backward Pass

Algorithm 1 Backward Pass

Initialize X, W, Y
$\mathcal{V}_{xx,N}, \mathcal{V}_{x,N} \leftarrow$ TERM_GRAD(x_N, y_N)
for $i \leftarrow N-1 : -1 : 1$ **do**
    if i = 1 **then**
        $l_x, l_{xx}, l_w, l_{ww} \leftarrow$ INIT_GRAD(x_i, w_i, y_i)
    else
        $l_x, l_{xx}, l_w, l_{ww} \leftarrow$ STAGE_GRAD(x_i, w_i, y_i)
    end if
    $A, B_w \leftarrow$ LINEARIZE_DYNAMICS(x_i, w_i)
    $Q_x \leftarrow l_x + A^T \mathcal{V}_{x,i+1}$ ▷ (4.7)
    $Q_w \leftarrow l_w + B_w^T \mathcal{V}_{x,i+1}$
    $Q_{xx} \leftarrow l_{xx} + A^T \mathcal{V}_{xx,i+1} A$
    $Q_{ww} \leftarrow l_{ww} + B_w^T \mathcal{V}_{xx,i+1} B_w$
    $Q_{xw} \leftarrow A^T \mathcal{V}_{xx,i+1} B_w$
    $k_i \leftarrow Q_{ww}^{-1} Q_w$ ▷ (4.8)
    $K_i \leftarrow Q_{ww}^{-1} Q_{xw}^T$
    $\mathcal{V}_{xx,i} \leftarrow Q_{xx} - Q_{xw} Q_{ww}^{-1} Q_{xw}^T$ ▷ (4.9)
    $\mathcal{V}_{x,i} \leftarrow Q_x - Q_{xw} Q_{ww}^{-1} Q_w$
    $\Delta J \leftarrow \Delta J + Q_w^T k_i$ ▷ Expected decrease
end for
return $K, k, \Delta J$

Using the principle of optimality, the optimal cost-to-go $\mathcal{V}_k$ can be recursively written as:

$$\begin{aligned} \mathcal{V}_i(x_i, u_i, w_i) = &\min_{w_i} l_i(x_i, u_i, w_i) \\ &+ \mathcal{V}_{i+1}(f(x_i, u_i, w_i)) \end{aligned} \tag{4.3}$$

Where l_i is the cost associated with a single timestep in the optimization, which is represented by a single term in the summation in (4.2). However, for ease of notation, we will suppress the u_i in these expressions as the control input is assumed to be known and not optimized over. The cost-to-go, Q, which will be optimized over to obtain $\mathcal{V}$ can be written as:

$$Q_i(x_i, w_i) = l_i(x_i, w_i) + \mathcal{V}_{i+1}(f(x_i, w_i)) \tag{4.4}$$

This can be approximated with a second order Taylor expansion to determine cost change:

$$\Delta Q_i = Q_i(x_i + \delta x_i, w_i + \delta w_i) - Q_i(x_i, w_i) \tag{4.5}$$

$$\approx \begin{bmatrix} 1 \\ \delta x_i \\ \delta w_i \end{bmatrix}^T \begin{bmatrix} 0 & Q_{x,i}^T & Q_{w,i}^T \\ Q_{x,i} & Q_{xx,i}^T & Q_{xw,i}^T \\ Q_{w,i} & Q_{xw,i} & Q_{ww,i}^T \end{bmatrix} \begin{bmatrix} 1 \\ \delta x_i \\ \delta w_i \end{bmatrix} \tag{4.6}$$

Where the block matrix entries can be written as:

$$Q_{xx} = l_{xx} + A^T \mathcal{V}_{xx,i+1} A \tag{4.7a}$$

$$Q_{ww} = l_{ww} + B_w^T \mathcal{V}_{xx,i+1} B_w \tag{4.7b}$$

$$Q_{xw} = l_{xw} + A^T \mathcal{V}_{xx,i+1} B_w \tag{4.7c}$$

$$Q_x = l_x + A^T \mathcal{V}_{x,i+1} \tag{4.7d}$$

$$Q_w = l_w + B_w^T \mathcal{V}_{x,i+1} \tag{4.7e}$$

Where B_w is the Jacobian of the dynamics function with respect to the process noise. Using this second order Taylor expansion, the optimal update for w_i can be calculated as:

$$\begin{aligned} \delta w_i^* &= -Q_{ww,i}^{-1} Q_{xw,i}^T \delta x_i - \alpha_i Q_{ww,i}^{-1} Q_{w,i} \\ &:= -K_i \delta x_i - \alpha_i k_i \end{aligned} \tag{4.8}$$

Where α_i is a line search parameter and K and k are defined to be the feedback and feedforward gains.

With the Taylor series approximation, the backward pass update to approximate the value function gradient and hessian at each timestep can be derived by substituting the optimal update into (4.6):

$$\mathcal{V}_{x,i} = Q_{x,i} - Q_{xw,i} Q_{ww,i}^{-1} Q_{w,i} \tag{4.9a}$$

$$\mathcal{V}_{xx,i} = Q_{xx,i} - Q_{xw,i} Q_{ww,i}^{-1} Q_{xw,i}^{T} \tag{4.9b}$$

These updates are then applied backward from the last measurement to the prior on the initial condition. An outline of this process is found in Algorithm 1.

Initial Point Update

Since the initial point is not known with certainty as in an iLQR control problem, an additional step must be added to update the initial point in the estimation problem [48]:

$$x_0' = x_0 - \mathcal{V}_{xx,0}^{-1} \mathcal{V}_{x,0} \tag{4.10}$$

This can be derived by looking at the approximation of the initial cost and taking the derivative of the expression with respect to a perturbation in x. The optimal point should have a derivative of zero:

$$\mathcal{V}(x + \delta x) = \mathcal{V}(x) + \mathcal{V}_x(x)\delta x + \delta x^T \mathcal{V}_{xx}(x)\delta x \tag{4.11a}$$

$$\frac{\partial}{\partial x}[\mathcal{V}(x + \delta x)] = \mathcal{V}_x(x) + \mathcal{V}_{xx}(x)\delta x \tag{4.11b}$$

$$0 = \mathcal{V}_x(x) + \mathcal{V}_{xx}(x)\delta x \tag{4.11c}$$

$$\delta x^* = -\mathcal{V}_{xx}^{-1}\mathcal{V}_x \tag{4.11d}$$

Practically, this is taken as a direction of improvement and is incremented by a search parameter α during a line search in the forward pass of the algorithm.

Forward Pass

After the backward pass determines the feedforward and feedback gains on the noise input, a forward pass is then performed to update the state and process noise estimates at each timestep. The forward pass of the algorithm performs a line search with α over the feedforward gain k calculated in the backward pass for each timestep with (4.8). The outline of this process is found in Algorithm 2.

Algorithm 2 Forward Pass

Initialize X, W, J, K, k
it $\leftarrow 1$
$\alpha \leftarrow 1$
while $J > J_{goal}$ & it < max-itr **do**
    $X, W, J \leftarrow$ ROLLOUT(X,W,K,k,α) ▷ (4.8)
    $\alpha \leftarrow \alpha/2$ ▷ Line Search
    it++
end while
return X, W, J

4.5 Methods

In this section we present the details of our proposed algorithm, Hybrid iterative Linear Quadratic Estimation (HiLQE). As in the iLQE methods discussed in Sec. 4.4.3, HiLQE will iteratively perform backward and forward passes until either a convergence parameter is met, such as a threshold in the cost decrease for a timestep, or until a set number of iterations is performed.

4.5.1 Backward Pass

The backward pass generates cost gradient information at each timestep so that the forward pass can update the process noise estimate to reduce the estimation cost. In this work, we limit the class of hybrid systems to those with continuous measurement information through hybrid events. This constraint ensures that the cost gradients are well defined near hybrid events. In systems where the measurements are not continuous through hybrid events, there is mode information implicitly encoded in the measurements and a different approach should be taken to utilize that information. In this pass, we deal with the smooth timesteps and the hybrid timesteps in different manners.

Smooth Case

In timesteps that did not experience hybrid events in the prior rollout, the cost gradients are calculated in the same way as in the standard estimation algorithm presented in Sec. 4.4.3. Based on the optimality principle, individual timesteps can be handled independently. Because of this, we do not need to account for hybrid events that occur in other timesteps when calculating gradients for a smooth timestep.

Hybrid Case

When the backward pass calculates the gradient information for a timestep that contains a hybrid event, the reset map and time-to-impact variations must be accounted for in the gradient information. To account for this, we augment the state and process noise Jacobians with the saltation matrix from

(4.1). For simplicity, we assume that hybrid events occur at the end of timesteps. Using the saltation matrix we update the backward pass from (4.7a)-(4.7e):

$$Q_{xx} = l_{xx} + A^T \Xi^T \mathcal{V}_{xx,k+1} \Xi A \tag{4.12a}$$

$$Q_{ww} = l_{ww} + B_w^T \Xi^T \mathcal{V}_{xx,k+1} \Xi B_w \tag{4.12b}$$

$$Q_{wx} = l_{wx} + B_w^T \Xi^T \mathcal{V}_{xx,k+1} \Xi A \tag{4.12c}$$

$$Q_{x} = l_{x} + A^T \Xi^T \mathcal{V}_{x,k+1} \tag{4.12d}$$

$$Q_{w} = l_{w} + B_w^T \Xi^T \mathcal{V}_{x,k+1} \tag{4.12e}$$

The saltation matrix Ξ is calculated at the stored pre-impact state for each of the timesteps with hybrid events.

4.5.2 Forward Pass

The forward pass is used to update the state and noise estimates based on the information from the cost gradient of the previous backward pass. This step involves integrating the dynamics forward at each step based on updated noise inputs. As in the backward pass, the smooth and hybrid timesteps are handled differently.

Smooth Case

The dynamics that are used to integrate the state forward at each timestep are selected based on the currently estimated active mode as opposed to the standard algorithm where there is one set of dynamics applied for all timesteps. In each timestep, the forward integration is augmented with an event function to check for situations where guard conditions for the currently active mode are met.

In cases where no guard condition is met, the feedforward and feedback gains are applied as they would be in a standard forward rollout step.

Hybrid Case

In contrast, when a timestep contains a hybrid transition, the forward pass needs to be modified. In these cases, a guard condition will be met during a forward rollout. At the instant when that guard condition is met, the reset map between the two hybrid modes will be applied and the dynamics will be updated from the first mode's dynamics to the second mode's dynamics. From there, the integration will continue forward to the end of the discrete timestep under the new mode's dynamics.

The feedback gain calculation needs to be updated as well since reset maps may make trajectories that are very close together arbitrarily far apart in state space. An example of this is the case of a bouncing ball with a coefficient of restitution of 0.8 in which a the reference state has a velocity of $-10.1\frac{m}{s}$ slightly before impact and an estimated velocity for the current iteration of $+8\frac{m}{s}$ slightly after impact. Nominally, the difference in velocity between these two states is $18.1\frac{m}{s}$. However, in this case, these two states should be treated as very close together since the hybrid transition makes the post impact velocities very similar. Without handling this, the estimate for the updated measurement noise can be updated far more than it should because of the large δx that would be used in its calculation, as in algorithm 2.

This must be handled by reference extension for feedback gain calculation. In the case of reference extension, the current estimated state should not be directly compared to the state estimate from the prior forward pass. Instead it will be compared to a value meant to reflect the prior estimate's distance from the current estimate projected into the current mode.

In cases where the new rollout has not made a transition and the reference state has, we back up to the impact point of the reference state and integrate it forward with the original dynamics to obtain the reference state. In cases where the new rollout has made a transition before the former trajectory, we combine the difference between the current state and the nominal post-impact state from the prior rollout with the difference between the current reference state and the nominal reference impact state propagated with the saltation matrix to map that displacement into the new mode:

$$\delta x = X_n(i) - r(X_{ref}(j)) + \Xi(X_{ref}(j) - X_{ref}(i)) \tag{4.13}$$

Where X_n is the current rollout, X_{ref} is the reference state from the prior forward pass, i is the current timestep of interest, and j is the timestep at which the reference state reaches the hybrid event.

4.6 HiLQE Performance on an Sample Systems

4.6.1 Bouncing Ball

In this section, we present the estimation performance on a two-dimensional bouncing ball system. The state of the system is represented as:

$$q = [x, y, \dot{x}, \dot{y}]^T \tag{4.14}$$

Additionally, the coefficient of restitution of the system was 0.8. For this test, the position of the center of mass is measured, and not the velocity to ensure that the measurements are continuous near the impact event. The data for this analysis was collected over 1000 one second long trials with measurements at 100Hz, which results in a single impact event during each trial. The initial

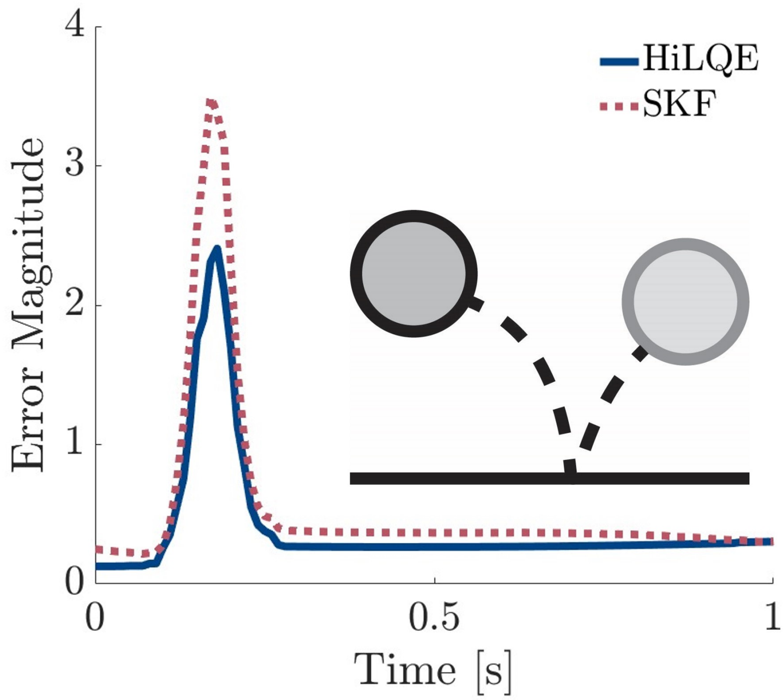


Figure 4.2: Average error magnitude over 1000 trials when simulating a bouncing ball system (inset) which impacts the ground once during its execution. The errors from the proposed method are shown in blue solid lines while the errors from the salted Kalman filter are shown in dashed red lines.

conditions for the trial were $x = 0, y = 1, \dot{x} = 0.5, \dot{y} = -5$. The system was corrupted with a process noise of covariance $0.1I_4$ and the measurement was corrupted with covariance I_2.

The comparison of the total magnitude of the error as well as an illustration of the system can be seen in Fig. 4.2. The HiLQE method outperformed the SKF at a peak of 48.60% for a timestep near the impact event, and the median MSE trajectory improvement across all of the trials was 30.48%.

This improvement arises out of HiLQE's ability to shift around the impact timing with more measurement information. Associated with this, HiLQE reduces the average mode mismatch at peak near impact events by 34% when compared to the SKF.

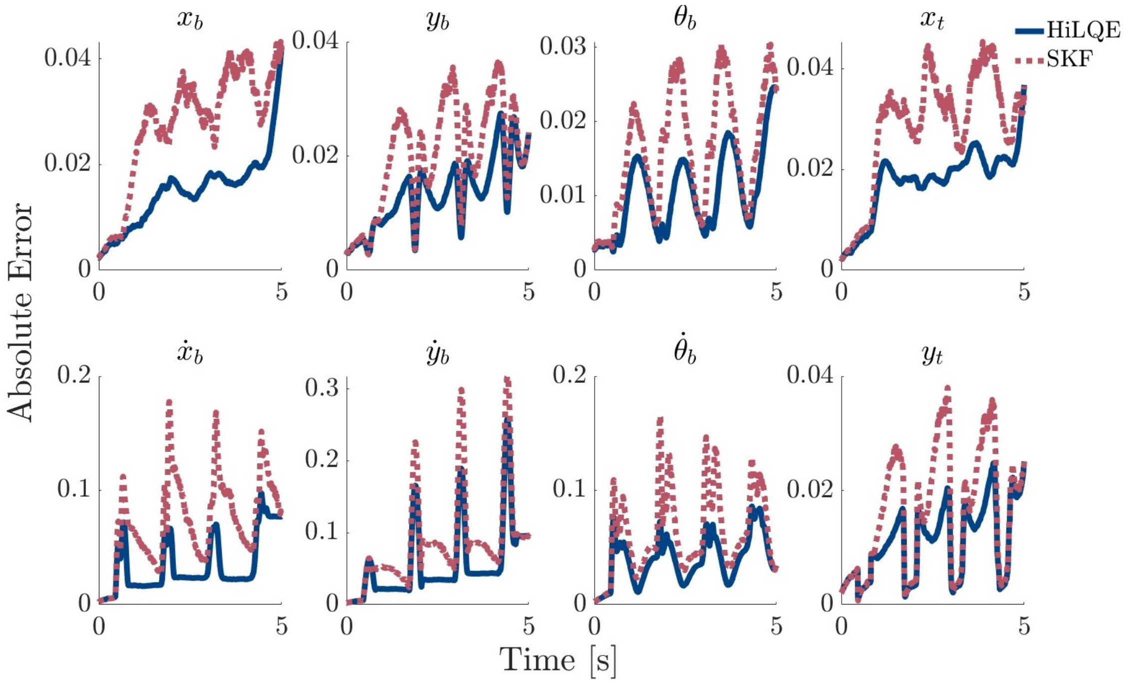


Figure 4.3: Average error for each state over 100 trials when simulating an ASLIP system which impacts the ground four times during its execution. The errors from the proposed method are shown in blue solid lines while the errors from the salted Kalman filter are shown in dashed red lines.

4.6.2 ASLIP Hopper System

We also demonstrate the efficacy of this method on the ASLIP (asymmetric spring loaded inverted pendulum) hopper [47], which can be seen in Fig. 4.1. The state for the ASLIP hopper is defined as:

$$q = [x_b, y_b, \theta_b, x_t, y_t, \dot{x}_b, \dot{y}_b, \dot{\theta}_b]^T \tag{4.15}$$

Where $x_b, y_b,$ and θ_b are the center of mass positions and orientation and $x_t,$ and y_t are the toe positions. A schematic for this system can be seen in Fig. 4.1. In the aerial phase, the system uses ballistic dynamics with a fixed leg length and angle. We used Lagrangian dynamics to generate the dynamic equations for the contact mode. The guard for touchdown when $y_t = 0$, or when the toe height reaches zero. The guard for liftoff is $l_l - l_0 = 0$, where l_l is calculated from the displacement between the toe and the hip attachment point to the body and l_0 is the rest length of the leg spring. The full derivation of the dynamics is presented in [7].

For the results presented in this work, the data is gathered from 100 five second long trials at a sample rate of 1 kHz, which typically results in four jumps, depending on the sampled noise. The parameters used for these trials were:

$$m_b = 1\mathrm{kg}, a_g = 9.8\mathrm{m/s}^2, l_b = 0.5\mathrm{m}, I_b = 1\mathrm{kgm}^2,$$
$$k_h = 100\mathrm{nm/rad}, k_l = 100\mathrm{n/m}, l_0 = 1\mathrm{m}, \phi_0 = 0\mathrm{rad}$$

For these trials, the state was nominally initialized at $y_b = 2.5\mathrm{m}$, $y_t = 1\mathrm{m}$, with all other states and velocities set to zero. This initial state is corrupted with a covariance of $10^{-5}I_8$. The process noise sampled for these trials was $W = 10^{-1}I_8$ and the measurements were taken for all positions (but not velocities) with measurement covariance $V = I_5$.

The comparison of the total magnitude of the error across all dimensions can be seen in Fig. 4.1, and a breakdown of the error reduction in each dimension of the state can be seen in Fig. 4.3. The peak average percent improvement for a single timestep was 62.25% throughout these trials, which indicates that in many of the spikes near impact, the HiLQE algorithm strongly outperforms the SKF. Similarly, the median MSE improvement throughout entire runs was 60.11% throughout the 100 trials as HiLQE consistently outperforms the SKF at all timesteps within these trials.

4.7 Conclusion and Future Work

In this chapter we presented HiLQE, an offline method for estimating the state of hybrid systems. We used the saltation matrix to obtain gradient information through hybrid events to approximate the value function gradient and hessian at each timestep in the backward pass. In the forward pass we modified the dynamics to include checking for guard conditions and applying reset maps. Additionally, when calculating feedback gains in the forward pass, we applied a reference extension method to deal with mode mismatch due to shifting estimates of impact times.

The performance of the hybrid estimator was demonstrated on a simple two dimensional bouncing ball system as well as an ASLIP hopper model. This algorithm produced notably better results than the comparable SKF. In the presented trials, HiLQE outperformed SKF by a peak of 62.25% at the worst timesteps near impact events on the ASLIP hopper.

There are still a wide variety of interesting research directions in hybrid state estimation. An immediate extension to this work would be to implement this iterative method online over a moving horizon of state estimates in an analogous manner to Model Predictive Control (MPC) for control problems. This work could also be combined with information seeking control. To deal with uncertainty in hybrid mode, we could look to designing control schemes that seek to gain as much confidence in the current mode as possible while still achieving other control goals. In legged systems, an optimization like this might result in a behavior such as slamming legs into the ground rapidly to ensure contact is made. Another potential direction for future work is platform and controller design for hybrid systems. One interesting aspect of this would be to investigate whether it is possible to optimize robot designs and gaits in a way that the dynamics can be relatively accurately expressed as smooth systems. This would save on the need to consider time-to-impact variations and simplify the gradient computations. Additionally, this work could be integrated with the Simultaneous Localization And Mapping (SLAM) problem by treating the solutions to the visual estimations as measurements on the system state in this optimization framework.

Chapter 5

Contact Mode Estimation for Bipedal Robotic Systems

5.1 Abstract

As bipedal robots become more and more popular in commercial and industrial settings, the ability to control them with a high degree of reliability is critical. To that end, this paper considers how to accurately estimate which feet are currently in contact with the ground so as to avoid improper control actions that could jeopardize the stability of the robot. Additionally, modern algorithms for estimating the position and orientation of a robot's base frame rely heavily on such contact mode estimates. Dedicated contact sensors on the feet can be used to estimate this contact mode, but these sensors are prone to noise, time delays, damage/yielding from repeated impacts with the ground, and are not available on every robot. To overcome these limitations, we propose a momentum observer based method for contact mode estimation that does not rely on such contact

sensors. Often, momentum observers assume that the robot's base frame can be treated as an inertial frame. However, since many humanoids' legs represent a significant portion of the overall mass, the proposed method instead utilizes multiple simultaneous dynamic models. Each of these models assumes a different contact condition. A given contact assumption is then used to constrain the full dynamics in order to avoid assuming that either the body is an inertial frame or that a fully accurate estimate of body velocity is known. The (dis)agreement between each model's estimates and measurements is used to determine which contact mode is most likely using a Markov-style fusion method. The proposed method produces contact detection accuracy of up to 98.44% with a low noise simulation and 77.12% when utilizing data collect on the Sarcos© Guardian® XO® robot (a hybrid humanoid/exoskeleton).

5.2 Introduction

Without explicit knowledge of which feet are in contact with the ground, a legged robot may take unreasonable or unsafe control actions. As an example, if a robot expects to step on the ground while its foot is still mid-air or if it expects to swing a leg while it is still in contact with the ground under some load, this mismatch between expectation and reality can result in the robot tripping and potentially falling. Therefore, it is imperative that the current contact mode of a robot is estimated so that appropriate control actions can be taken.

Traditional methods for detecting the contact mode on legged robots make certain assumptions that often do not hold for bipedal (humanoid) robots. These methods typically take one of two approaches. First, a contact sensor, such as a switch or a force-torque sensor, is installed at the robot's foot so that contact detection becomes a problem of filtering or thresholding a single sensor's data [31, 56, 57, 58, 59]. This method requires hardware changes and is not ideal for robots requiring

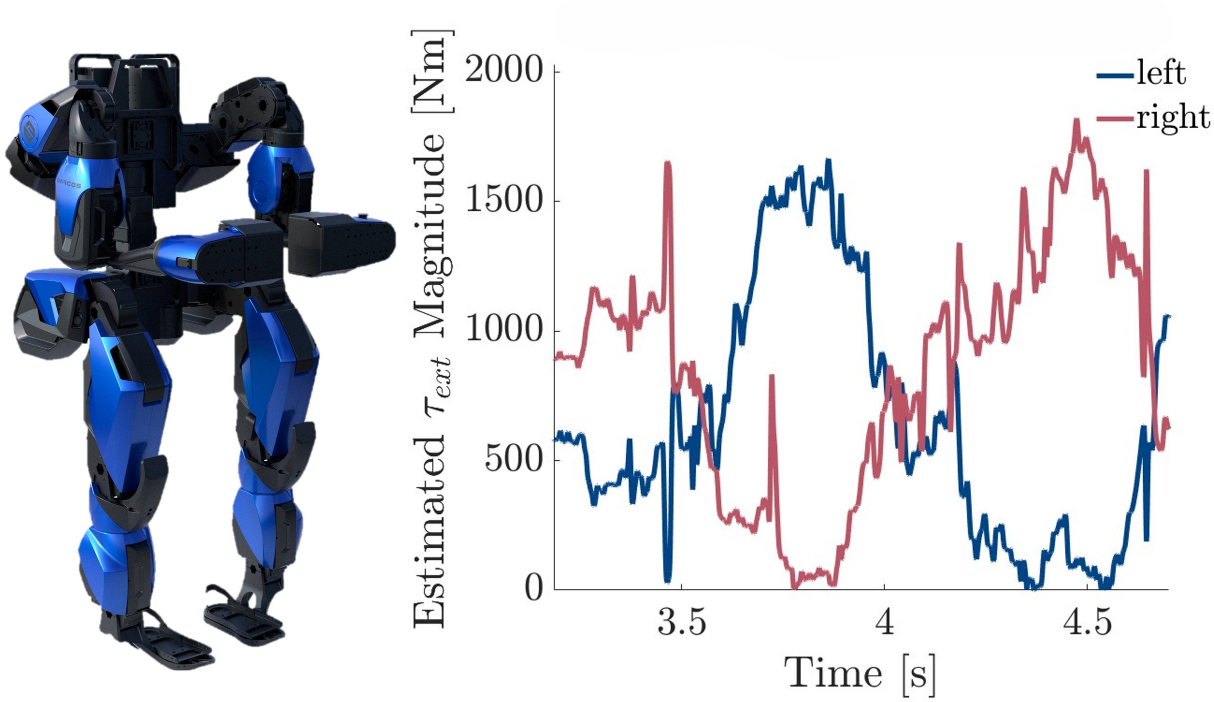


Figure 5.1: Left: The Sarcos© Guardian® XO®, a hybrid humanoid robot/exoskeleton on which we demonstrate these algorithms. Right: The estimated external torques on the system during a gait cycle using each separate momentum observer.

robust sensing over long lifespans, as contact sensors are easily damaged due to the impulsive nature of repeated impact with the ground during locomotion. The other method that is often used for contact detection is a thresholding method on estimated contact forces from joint torques [60, 61]. The issue with this method is that it assumes that the robot's body frame can be treated as an inertial frame and that the legs have significantly lower mass than the body. This assumption does not hold in the case of many bipedal robots as the legs often represent a substantial portion of the robot's mass, such as the Sarcos© Guardian® XO® shown in Fig. 5.1 which has close to 50% of total body mass in each leg, and therefore movement of the legs has an appreciable effect on the state of the body.

This work proposes a new contact estimation system that addresses these challenges through the use of multiple momentum observers that adopt different contact assumptions. We avoid the assumption that the robot's body frame can be taken as an inertial frame by instead using the world frame as the inertial frame with an assumed stationary foot contact for the stance foot to reduce the order of the dynamics. This assumption allows the 6 constraints at a given foot (in 3D) to remove the need to estimate the base link position, orientation, and velocities. Then, a momentum observer [62, 63, 64, 32, 65] is applied to the constrained dynamics of the system in each contact mode. By comparing the results of momentum observers on each of these assumptions, as well as monitoring the relative velocity between the feet, we obtain a probability of each contact condition (including dual support).

The proposed approach is thus a novel combination of reduced order dynamics, momentum observers, and Markov models which enables contact detection in large scale bipedal robots without: 1) direct contact sensors, 2) using the second derivative of any measurement, 3) precise knowledge of the body velocity, or 4) assuming that the legs are much lighter than the body or moving much faster than the body. The knowledge of contact states provided by this algorithm will enable the use of a variety of full body state estimators that rely on contact knowledge.

5.3 Related Work

There has been a wide array of research on state estimation for legged robotic systems, here we will primarily focus on floating base and contact estimation methods. Additionally, we will discuss the use of momentum observers for estimation of external contact forces.

5.3.1 Floating Base Estimation

Often, floating base estimators will assume that contact information is already provided. To enable these floating base estimators, we will need to provide this information, which is not directly available in most bipedal robots.

In [31], an inertial measurement unit (IMU) is used as a process model in an extended Kalman filter (EKF) and the legged kinematics of a quadrupedal robot are used in the measurement update. This work was then extended in [56] to apply to humanoid robots. In these algorithms, the contact information is used to update the process noise in the prediction step of the EKF. In the case where a foot is not believed to be on the ground, its process noise is set to infinity, or a very large number, so that it is free to reposition.

In [66], IMU and kinematic data is fused through an unscented Kalman filter (UKF). This method similarly rejects information from legs that are not detected to be in contact with the ground. However, it additionally uses outlier rejection to ignore data from feet that may be slipping. This outlier rejection could potentially be used to differentiate feet that are not in contact with the ground, however we propose a different method for contact detection.

Other works have used factor graph based methods for state estimation to fuse together different sensor modalities. In [57, 58], forward kinematic constraints are added into a factor graph formulation alongside visual and IMU factors to estimate trajectories. This work also requires a binary contact sensor or other means of determining active contact modes to determine which kinematic constraints are active between nodes on the factor graph. Similar work was presented in [67], which seeks to rely on contact classification less, but still uses estimates of external contact forces. This was also extended in [68] to include an additional foot velocity factor, but still utilized the kinematic constraint factors from the other works.

In [61], the factor graph formulation is extended to simultaneously estimate parameters for the system that may be inaccurate such as link lengths. While this improves over the other methods in many ways, it still requires a sensor to provide the current contact state to construct the kinematic constraints for the optimization problem.

5.3.2 Contact Estimation

Contact estimation is critical to our ability to estimate whole body state in legged robots. There have been a variety of approaches to estimating the current contact conditions of a system.

Many methods have placed sensors directly on the feet to determine which contact modes are active. In [66, 57, 58], binary contact sensors are employed to determine the active contact modes for the floating base estimators. Rather than a binary sensor, [59] uses a force-torque sensor mounted on the feet to determine the active contact mode. Similarly, [69] places additional IMUs on each of the feet as an additional means of sensing the active contact mode. With the relatively slow walking gait of the bipedal robot we consider in this paper, the differences in velocities are not as drastic between contact modes as in quadrupeds with faster leg trajectories during walking gaits. Furthermore, the standard walking gait has the feet slow down before impact to minimize the impulse on the heavy legs. Therefore, while the feet of our system do have an IMU already on them, during slow walking they do not provide much information on the active contact mode.

Alternatively, many algorithms have sought to estimate contact modes through encoder and torque measurements in robot's legs [70, 71]. These internal sensors have also been used in methods such as the one proposed in [32], which utilizes a momentum observer. Our approach also uses a set of momentum observers as a part of the overall system, and we further discuss these systems in the following section. However, these prior methods assume that the robot's legs are significantly lighter than the body in order to ignore inertial effects on the body. This assumption does not hold in the case of large bipedal robots as the legs represent a significant portion of the robot's mass. Additionally, the method in [32] relies on knowledge of the ground plane height as an additional means of determining contact probabilities. Our method avoids making this assumption to remove the method's dependence on knowledge of the environment.

A comparison between each of these methods is presented in Table 5.1. Our proposed method is able to perform contact estimation without utilizing assumptions that do not hold for our large-scale bipedal robot and without modifying the hardware on the robot.

	Rough Ground	Non-inertial Base Frame	No Hardware Modification	Low Speed
Proposed Method	✓	✓	✓	✓
Leg-Momentum Observer [32]	✗	✗	✓	✗
Contact Sensor [66, 57, 58]	✓	✓	✗	✓
Foot IMU [69]	✓	✓	✓/✗	✗

Table 5.1: Comparison between the proposed method and a variety of existing contact sensing methods.

5.3.3 Momentum Observers

One method for determining the current contact condition is to use a momentum observer. Momentum observers, as introduced in [62, 63] and summarized in [64], utilize the generalized momentum of a system to estimate external contact forces. By examining momenta rather than accelerations, these methods have the benefits of avoiding costly inversions of generalized inertia matrices and the use of noisy estimates of joint accelerations from encoder data. The difference between the expected and actual momenta at the end of each timestep are then taken to be the result of some externally applied torques. Further explanation of the details of momentum observers can be found in Sec. 5.4.1.

Recently, momentum observers have begun to see use in contact state estimation for legged robots. In [32], the estimates from a momentum observer are fused with gait cycle information and the estimated foot height to determine the current contact mode. This method assumes a reasonable and steady gait cycle so as to determine the likelihood of contact based on clock timing, which does not always hold for more variable/dynamic walking behaviors. Additionally, this method uses an estimate of the height of each foot as well as a measure of ground roughness as additional factors in determining contact probability. As we seek to decouple floating base estimation and the contact state estimation, we cannot rely on estimates of the current foot height for contact mode estimation as they are typically derived from estimates of the centroidal state of the robot. Alternatively, in [65], a momentum observer is used to detect entanglements of a robot's legs with vines or other environmental disturbances during planned swing phases of a quadruped's gait cycle. However, both of these methods implicitly assume that the robot's body can be taken as an inertial frame due to the relative masses of the bodies and legs of the robots, which does not hold for larger bipedal robots.

5.4 Methods

5.4.1 Momentum observers

Many force estimation methods rely on acceleration data which is often noisy as it requires position data to be differentiated twice. Alternatively, momentum observers avoid the use of these noisy acceleration estimates by instead utilizing the difference between the evolution of the expected momentum and the observed momentum of a system to estimate the external forces and torques acting on it.

Consider a floating base (humanoid) robot with the following dynamic equations:

$$M\ddot{q} + C\dot{q} + G + A^T\lambda = B^T\left(\tau_{\text{mot}} + \tau_{\text{ext}}\right) \tag{5.1}$$

where $\{q, \dot{q}, \ddot{q}\}$ represent the generalized coordinates of the system (i.e., joint angles + base position/orientation) and their derivatives, M is the generalized momentum matrix, C is the Coriolis matrix, and G is the gravity vector. Modeled constraint forces (λ) act on the system through the constraint matrix A, while applied motor torques (τ_{mot}) and unmodeled/external torques (τ_{ext}) act through the selector matrix ($B^T\tau \equiv [0_{6\times1}^T\ \tau^T]^T$) that simply ensures the floating base degrees-of-freedom (DOFs) are not actuated. Trivially, it is possible to calculate the external torques with measurements/estimates of q (and its derivatives), λ, and τ_{mot} utilizing the inverse dynamics approach. But rather than relying on noisy acceleration data, we instead consider the derivative of the system's generalized momentum $p = M\dot{q}$:

$$\begin{aligned}\dot{p} &= \dot{M}\dot{q} + M\ddot{q} \\ &= \dot{M}\dot{q} - (C\dot{q} + G + A^T\lambda) + B^T\left(\tau_{\text{mot}} + \tau_{\text{ext}}\right)\end{aligned} \tag{5.2}$$

Taking advantage of the identity $\dot{M} = C + C^T$, the dependence on the time derivative of the mass matrix can be removed to recover an equation for the rate of momentum that does not depend explicitly on joint accelerations:

$$\dot{p} = C^T\dot{q} - G - A^T\lambda + B^T\left(\tau_{\text{mot}} + \tau_{\text{ext}}\right) \tag{5.3}$$

$$= \beta + B^T\tau_{\text{ext}} \tag{5.4}$$

where $\beta = C^T\dot{q} - G - A^T\lambda + B^T\tau_{\text{mot}}$. Using this formulation, now we maintain an estimate of the system's momentum, $\hat{p}$, and compare it to the measured momentum at each timestep to obtain estimates of the external torques $\hat{\tau}$, as in [65]:

$$p = M\dot{q} \tag{5.5a}$$

$$\hat{\tau} = B^{-T}K_O(p - \hat{p}) \tag{5.5b}$$

$$\dot{\hat{p}} = \beta + B^T\hat{\tau} \tag{5.5c}$$

Where K_O is an observer gain for the estimation of the external torques. Practically, this is implemented by discretizing the system's dynamics and considering the evolution of momentum estimates when $\dot{\hat{p}}$ is assumed constant over the discretized timestep.

5.4.2 Constrained Dynamics

When a system is subject to holonomic constraints, such as flat-footed contact with the ground, the free dimensions are reduced by each independent constraint. In these cases, we can utilize the linearization of the constraints at a given operating point to reduce the degrees of freedom of the system. Consider the generalized dynamics from (5.1). We can simultaneously select a set of reduced coordinates $y = Yq$ and enforce the constraints $A\dot{q} = 0$:

$$\begin{bmatrix} A \\ Y \end{bmatrix} \dot{q} = \begin{bmatrix} 0 \\ I \end{bmatrix} \dot{y} \tag{5.6a}$$

$$\dot{q} = \begin{bmatrix} A \\ Y \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \dot{y} \tag{5.6b}$$

$$\dot{q} = H\dot{y}, \qquad H := \begin{bmatrix} A \\ Y \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{5.6c}$$

where the choice of coordinates y must be independent of the constrained directions such that the matrix in (5.6b) is invertible. By using $\dot{q} = H\dot{y}$ and $\ddot{q} = H\ddot{y} + \dot{H}\dot{y}$, and noting that $H^T A^T = 0$, we can now rewrite (5.1) as a reduced order (constrained) system that no longer explicitly depends on the constraint forces λ:

$$\begin{aligned} H^T M H \ddot{y} + \left(H^T C H + H^T M \dot{H}\right)\dot{y} \\ + H^T G + H^T A^T \lambda = H^T B^T (\tau_{\text{mot}} + \tau_{\text{ext}}) \end{aligned} \tag{5.7a}$$

$$\tilde{M}\ddot{y} + \tilde{C}\dot{y} + \tilde{G} = \tilde{\tau}_{\text{mot}} + \tilde{\tau}_{\text{ext}} \tag{5.7b}$$

where,

$$\tilde{M} = H^T M H \tag{5.8a}$$

$$\tilde{C} = H^T C H + H^T M \dot{H} \tag{5.8b}$$

$$\tilde{G} = H^T G \tag{5.8c}$$

$$\tilde{\tau}_{\text{mot}} = H^T B^T \tau_{\text{mot}} \tag{5.8d}$$

$$\tilde{\tau}_{\text{ext}} = H^T B^T \tau_{\text{ext}} \tag{5.8e}$$

5.4.3 Multiple Model Observers

For quadruped robots (with proportionately lighter limbs), the momentum observers described in (5.5) typically assume that the base frame of the robot can be treated as an inertial frame and thus only consider the dynamics of an individual leg. For larger bipedal robots, this assumption does not hold, so instead we utilize the ground as an inertial frame. In order to do so, we consider the various *contact-constrained* dynamics that arise from assuming only one foot in (flat) contact with the ground[1]:

$$H_i = \begin{bmatrix} A_i \\ S_{\text{DOFs}} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{5.9a}$$

$$\dot{q} = H_i \dot{y}_i \tag{5.9b}$$

$$\ddot{q} = H_i \ddot{y}_i + \dot{H}_i \dot{y}_i \tag{5.9c}$$

where $i \in \{l,\ r\}$ represents the single support contact modes. For each contact assumption, the reduced order dynamics are obtained (as in Sec. 5.4.2) by assuming the plant foot position/orientation is constant (i.e., $A_i \dot{q} = 0$) and selecting the joint positions as our reduced coordinates $Yq = S_{\text{DOFs}} q = y$.

$$\tilde{M}_i \ddot{y}_i + \tilde{C}_i \dot{y}_i + \tilde{G}_i = \tilde{\tau}_{i,\text{mot}} + \tilde{\tau}_{i,\text{ext}} \tag{5.10}$$

Using these constrained system dynamics, we can separately maintain a momentum observer like those in (5.5) for each of the contact modes:

$$p_i = \tilde{M}_i \dot{y}_i \tag{5.11a}$$

$$\beta_i = \tilde{C}_i^T \dot{y}_i - \tilde{G}_i + \tilde{\tau}_{i,\text{mot}} \tag{5.11b}$$

[1] It is important to consider flat contacts in 3D to fully constrain the system and to allow the constraint forces, λ, to be fully determined by the configuration of the robot (and its derivatives).

$$\hat{\tau}_{i,\text{ext}} = K_O(p_i - \hat{p}_i) \tag{5.11c}$$

$$\dot{\hat{p}}_i = \beta_i + \hat{\tau}_{i,\text{ext}} \tag{5.11d}$$

Intuitively, in situations where a leg is in single support and the other leg is in a swing phase and not impeded by its environment, then the estimated external torques $\hat{\tau}_{[l,r]}$ corresponding with that stance phase should be near zero. We can utilize this information to inform our belief in which contact mode is currently active. Additionally, when neither observer is estimating near zero external torques, it is likely that the robot is in a dual support phase (i.e., both limbs are experiencing "external" torques produced by contact with the ground). Lastly, spikes in the estimated external torque for the active mode also represent a low latency means of detecting touchdown events.

5.4.4 Velocity Constraints

While the external torque estimates can quickly enable detection of touchdown events, they are not as sensitive to feet lifting off of the ground. To fill in this gap, we additionally use the violation of a *relative* velocity constraint as a means of detecting liftoff. Specifically, during a dual support phase, the relative velocity between the two feet (calculated explicitly from joint measurements) should be zero, so when nonzero velocities are estimated it is likely that a liftoff event has occurred. We can therefore use the relative motion between the two feet as a metric for how likely it is the system is in dual support:

$$v_{\text{rel}} = A_{lr}\dot{q} = (A_l - A_r)\dot{q} \tag{5.12}$$

5.4.5 Model Fusion

To combine the estimates from each of the momentum observers as well as the velocity constraint violation, we maintain an estimate of the likelihood of each contact condition. To update this estimate, we utilize a Markov model based on how likely it is that each transition event has occurred.

To determine how likely a touchdown event is to have occurred from a swing-leg phase, a threshold value τ_t and a band value τ_b are defined to apply to a sigmoid activation function:

$$\pi_{i,\text{Dual}} = \text{sgm}\left(\frac{||\hat{\tau}_{i,\text{ext}}||_2 - \tau_t}{\tau_b}\right) \tag{5.13}$$

where $\pi_{i,Dual}$ is the probability of transition from the given single support mode into dual support.

Similarly, when calculating the transition probability from dual support to single support, we apply a sigmoid function centered about v_t with width v_b to the estimated velocity constraint violation. Then we determine which stance leg is more likely by comparing the relative magnitudes of external torques estimated by the two momentum observers – whichever has a lower estimate is more likely to be the active mode.

$$\pi_{\text{Dual},i} = \text{sgm}\left(\frac{v_{\text{rel}} - v_t}{v_b}\right)\frac{||\hat{\tau}_{j,\text{ext}}||_2}{||\hat{\tau}_{i,\text{ext}}||_2 + ||\hat{\tau}_{j,\text{ext}}||_2} \tag{5.14}$$

where, for each $i \in \{l, r\}$, $j \in \{l, r\}\backslash\{i\}$ and $\pi_{Dual,i}$ is the transition probability from dual support to the given single support mode.

The means and widths for each of these transition probability sigmoids can be tuned for a given system to achieve some desired performance characteristics. By combining these transition probabilities, we are able to calculate a Markov transition model for each timestep to update the likelihood of each given contact condition. We maintain the likelihood of each mode in vector form as:

$$\mathrm{P} = \begin{bmatrix} \mathrm{p}_l & \mathrm{p}_r & \mathrm{p}_{\text{Dual}} \end{bmatrix}^T \tag{5.15}$$

To update these probabilities, we construct the transition matrix Π at each timestep based on the individual transition likelihoods:

$$\Pi = \begin{bmatrix} 1 - \pi_{l,\text{Dual}} & 0 & \pi_{\text{Dual},l} \\ 0 & 1 - \pi_{r,\text{Dual}} & \pi_{\text{Dual},r} \\ \pi_{l,\text{Dual}} & \pi_{r,\text{Dual}} & 1 - \pi_{\text{Dual},l} - \pi_{\text{Dual},r} \end{bmatrix} \tag{5.16}$$

such that we can update the probabilities at each timestep by:

$$\mathrm{P}_{k+1} = \Pi \, \mathrm{P}_k \tag{5.17}$$

To select the active contact mode estimate, we take the most likely mode at a given time. This may sometimes result in selecting an active mode that is estimated to have less than a 50% likelihood. In practice, there are many ways to handle this information. The likelihoods themselves could be fed into other estimation methods as a means of representing how much certainty should be associated with dynamic constraints from a contact mode. Alternatively, an additional mode that represents an uncertain contact condition could be added in order to prevent switching when uncertainty is more likely. For the work proposed here, the contact mode estimate is only allowed to transition to a new contact mode once a threshold of belief in a given mode is crossed (e.g., $> 50\%$ likelihood for a given mode is required to switch to that mode).

5.5 Results

5.5.1 Simulated Walker

In this section we present the results of estimation performed on a planar five link walker simulated in Drake [72]. This walker was modeled to have significant mass distribution in its legs – each leg weighs twice as much as the base link – to ensure that the assumption that the floating base is an inertial frame does not hold. The model used in this system is shown in Fig. 5.2. Each link has a mass of 1 kilogram with a uniform mass distribution. The system has seven degrees of freedom, the position and orientation of the base link are three unactuated degrees of freedom, and the angle of each of the joints in the legs – left and right hips and knees – are four actuated degrees of freedom. The state representation of the system is:

$$q = \begin{bmatrix} x_b & y_b & \theta_b & \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T \tag{5.18}$$

and the dynamics matrices M, C, and G and the constraint matrix A are calculated using Lagrangian dynamics.

The bipedal walker is run through a left-right gait for these trials. Each momentum observer for this simulated system assumes that a foot is in point contact with the ground, which provides two constraints to the system. Using the constrained dynamics, as discussed in Sec. 5.4.3, we are able to remove the dependence of the estimator on the two body position variables, so the measurements used are the orientation of the base and each of the joint angles.

For a single 5 second trial running at 1kHz using a signal to noise ratio of 70dB, the associated confusion matrix, which compares the estimated modes to the ground truth modes from the

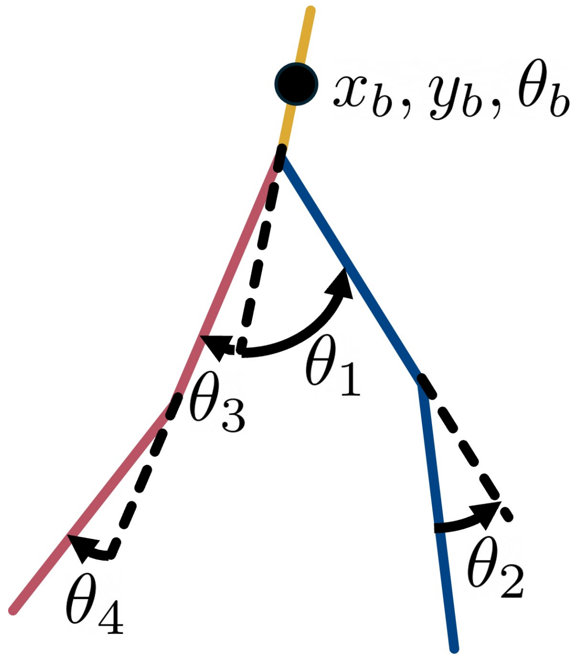


Figure 5.2: The five link walker model we use in simulation to evaluate the algorithm. The floating base location is defined as a position and angle relative to the vertical. The link angles are defined relative to straight hip and straight knee positions.

simulation, is shown in Fig. 5.3. Similarly, a timeseries of the estimates for a single trial can be seen in Fig. 5.4. The associated mode accuracy for this system is 98.44% with much of that error occurring in the initial timesteps while the momentum observers initialize.

To better characterize the performance of the estimator across a variety of levels of noise in the system, we present a plot comparing the mode accuracy as a function of the signal to noise ratio (calculated as $10\log(\frac{\text{signal}}{\text{noise}})$) in Fig. 5.5 over 100 trials at each noise level with randomly sampled noise. In this, we can see that when the noise in the system is minimal, a properly tuned estimator can reach near 100% accuracy. At higher noise levels, we see much lower accuracy. As the momentum observer is highly dependent on accurate velocity estimation, the error increasing due to noise in the velocity makes sense. This leads to neither observer estimating external torques

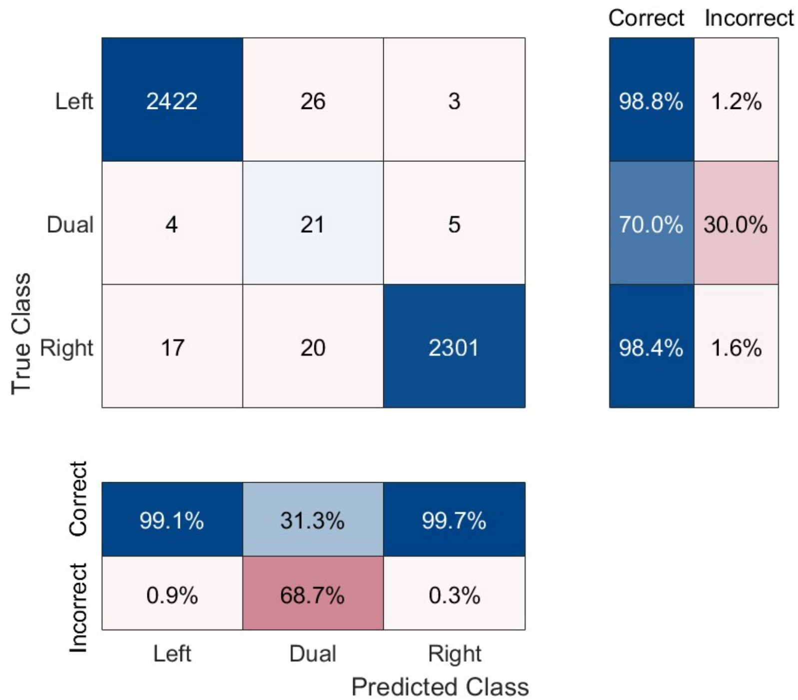


Figure 5.3: A confusion matrix for the mode estimation from the simulated five link walker over a five second trajectory with a signal to noise ratio of 70dB. The values on the diagonal in blue represent correct estimations and the values off the diagonal in red represent incorrect estimations. The bottom and side contain the percentages of the time that a specific estimate is correct and that a specific mode is correctly estimated, respectively.

near zero magnitude, so it is more likely to select dual support, leading to low accuracy.

5.5.2 Exoskeleton Walker

For this portion of the work, we utilize data collected from a Sarcos© Guardian® XO® robot (a hybrid humanoid/exoskeleton), which is shown in Fig. 5.1. This robot has a mass of 150kg and much of that mass is distributed in the limbs, so the assumption that the body link is an inertial frame cannot be adopted. This data was collected during a trial in which the XO walked as a humanoid on its own without a person inside controlling it. The measurements used for this experiment are the

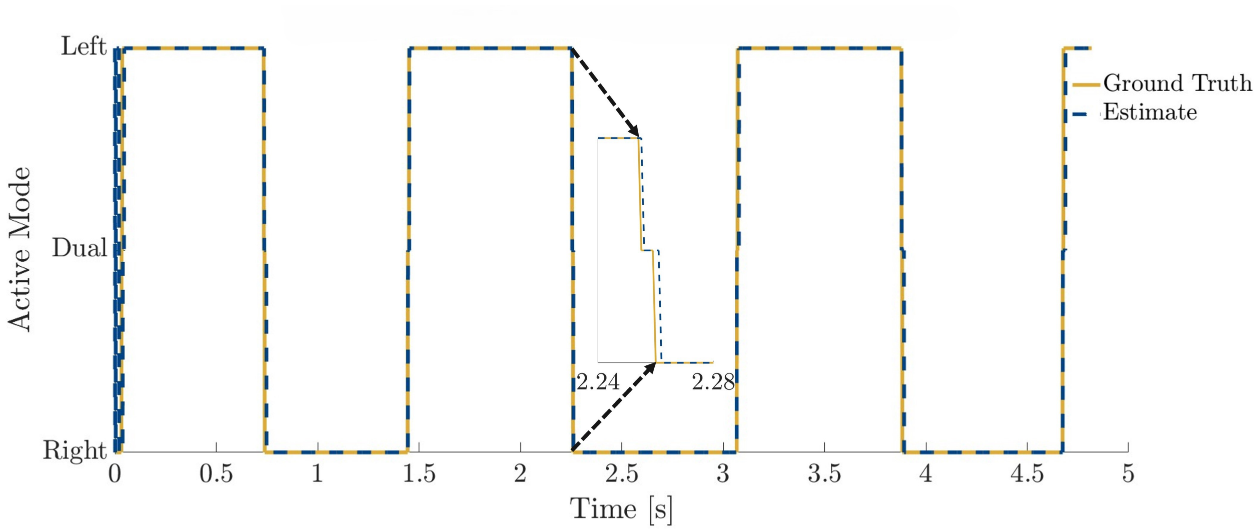


Figure 5.4: A comparison between the current contact mode estimate and the ground truth of the simulated five link biped system on a trial with a signal to noise ratio of 70dB. The estimate is shown as a blue dashed line and the ground truth is shown as a solid yellow line. An inset highlights the short lag in detecting touchdown and liftoff at a stance transition event.

measured joint torques (as opposed to the commanded), the orientation of the body via an IMU, and the joint positions via encoders.

All of the mode estimates on the XO are compared against hand-labeled contact modes derived from a video of the corresponding trial. A confusion matrix comparing the estimated mode to the mode extracted from the video is shown in Fig. 5.6. The associated estimates from the trial are shown as a time series in Fig. 5.7. The corresponding mode accuracy for this system is 77.12%, compared to 70.15% from the planned contact sequence. Directly utilizing the most likely contact mode for a given time without the probabilistic transition model results in a similar accuracy of 76.10%, however it predicts twice as many transitions than truly occur compared to 27.5% fewer than true transitions predicted by the transition model. While this is not a perfect model of the transitions, it is preferable to have a stable estimate of the current contact state for control purposes. Some of the misalignment can be explained by the difference in the estimation update rate and

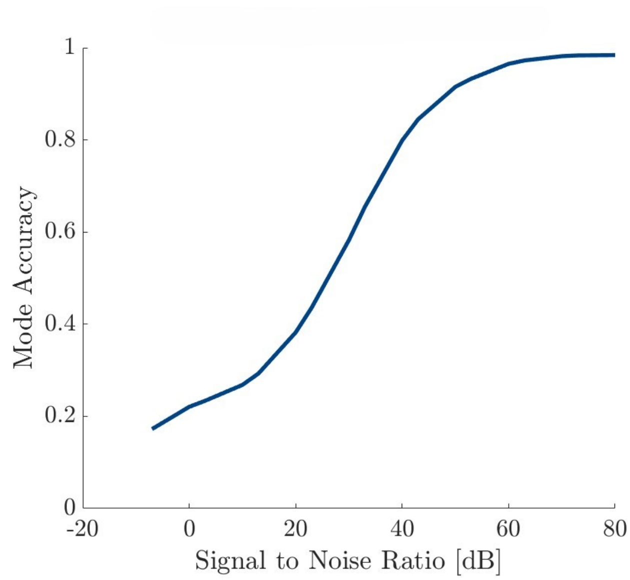


Figure 5.5: Estimation accuracy as compared to the signal to noise ratio of the system. With lower noise levels, we see very accurate mode estimation. However, as the momentum observer is sensitive to velocity noise, the performance degrades notably with higher noise levels.

the camera frame rate. Additionally, due to the hand-labeled nature of the ground truth, there are orientations in which it is difficult to discern the true contact conditions. However, the general timing and gait patterns align well between the estimate and the values derived from the associated video. To improve the practical performance, the cutoff values for the sigmoids can be tuned to either over- or under-estimate dual support depending on the context in which the estimation is being used. Additionally, this should be fused with more information, such as estimated ground heights from a visual or lidar sensor. However, for this work we sought to make this estimation agnostic to the profile of the ground.

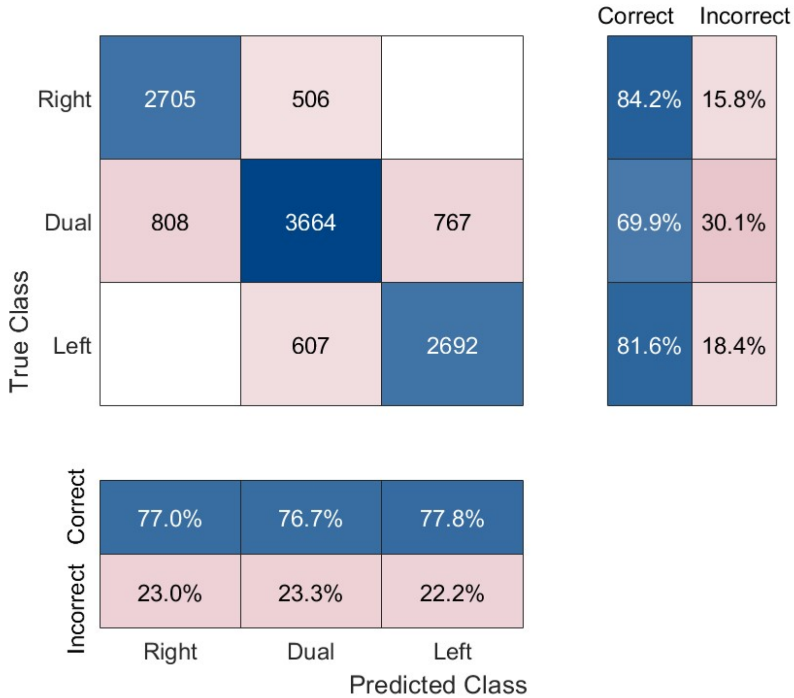


Figure 5.6: A confusion matrix for the mode estimation from the bipedal walker in hardware over an approximately twelve second trajectory. The values on the diagonal in blue represent times where the estimator and the video frame align and the values off the diagonal in red represent estimations where the two do not align. The bottom and side contain the percentages of the time that a specific mode estimate aligns with the video and that a specific mode from the video is correctly estimated, respectively.

5.6 Conclusion and Future Work

In this work we show that through the use of multiple momentum observers we are able to obtain estimates of the active contact mode of a system through only proprioceptive means. Using this method we are also able to avoid making the assumption that the base of the robot is an inertial frame.

We then demonstrate the effectiveness of this method by implementing it for both a simple simulated bipedal system and a large scale bipedal exoskeleton in hardware. In simulation we show

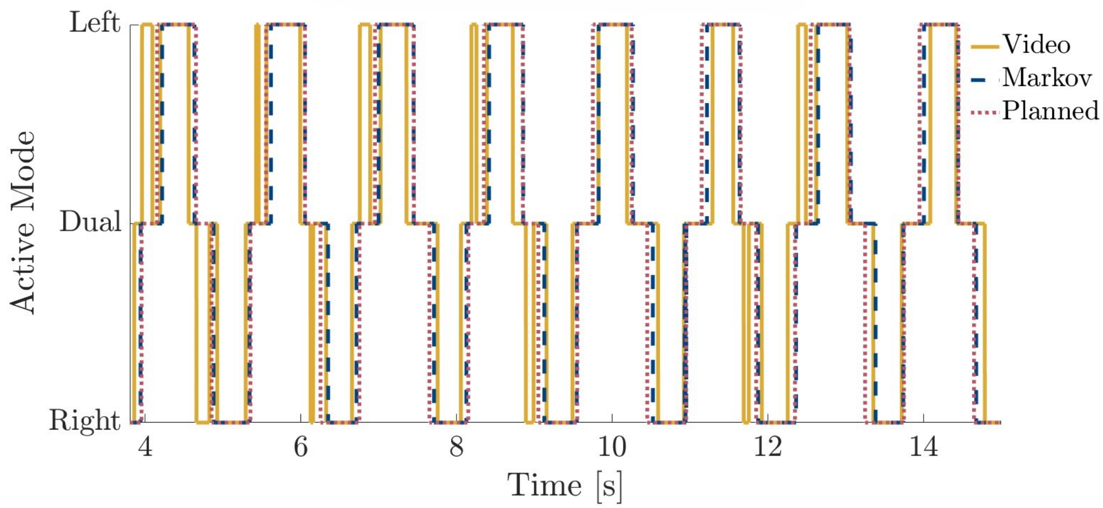


Figure 5.7: A comparison between the current contact mode estimate with the proposed method, the planned contact sequence, and the contact mode derived from the corresponding video. The estimate is shown as a blue dashed line, the planned state is shown as a red dotted line, and the mode extracted from the video is shown as a solid yellow line.

that the mode estimation accuracy is able to reach 98.44% with low noise. Additionally, in hardware we show that the estimator is able to track the active contact mode that we observe through our collected video during the trial more accurately than relying on the planned contact mode.

In combination with other sensor information, this method will allow robots without contact sensors in their feet to better estimate their current stance phases. This will enable them to take more well informed control actions, leading to more robust gaits.

However, there is still room for extensions to this work. So far, we have considered flat foot stance for each of the feet, limiting us to 3 contact modes. This could be extended to include heel strike and toe-off modes to better understand the control authority of the system at a given time.

Additionally, this work assumes that there is no flight phase as the systems we considered are not capable of flight phases. This method could be modified to differentiate between situations where neither individual momentum observer agrees with the measurements when in dual stance or when in flight.

Similarly, we assume that any contact occurring corresponds to a foot touchdown event. It is possible the robot will also be making contact or experiencing external forces and torques not due to foot touchdown. An interesting extension would be to differentiate between ground contact and other external forces, such as tripping events as discussed in [65].

While this method was designed specifically for bipedal systems, it could also be interesting to consider how these assumptions can be used to better understand the contact modes of quadrupedal and other systems. While the mass distribution of those systems does make the assumption of taking the floating base as an inertial frame more reasonable, it still doesn't hold perfectly. A difficulty with this extension would be considering the combinatorial nature of the contact modes.

Chapter 6

Conclusion

In this thesis we address the problem of state estimation for contact systems such as legged robots and robotic manipulators. We specifically seek to improve performance of estimation algorithms around contact events such as foot touchdown and initial grasping as they can result in nonsmooth and potentially discontinuous dynamics. Pre-existing methods are unable to explicitly handle these events, resulting in inaccurate estimations or failures. To handle contact events, this thesis uses the framework of hybrid dynamical systems, which treats different contact conditions as different discrete modes and uses continuous state information to trigger transitions between different discrete modes. By examining hybrid dynamical systems, we improve upon a variety of state estimation methods which each have unique advantages for different estimation goals.

For cases in which the environment and system dynamics are uncertain or potentially stochastic, we created the uncertainty aware Salted Kalman Filter (ua-SKF). This method explicitly considers distributions of potential parameter values. The ua-SKF has the benefit of being able to handle situations such as a robot walking over rough terrain where the exact height of the ground is unknown or grasping cases where the manipulated object has variation in dimensions.

When a system doesn't have access to a method of determining ground truth states, such as a motion capture environment, it is still important to be able to evaluate the performance of online estimators. To this end, we present Hybrid iterative Linear Quadratic Estimation (HiLQE). This work utilizes full horizon optimization to generate an estimate of states over the entire trajectory, which will be more accurate than any online method. This estimate can then be used as a baseline to evaluate online estimation methods when ground truth is not available. This method iteratively solves quadratic optimization problems rather than optimizing a full nonlinear problem at once. The saltation matrix is used for gradient information in the backward pass of the algorithm and the hybrid dynamics are rolled out in the forward pass.

Another major concern for hybrid systems is the determination of the current contact mode as the control authority of the system varies greatly depending on this contact mode. For example, in cases where a leg is in contact with the ground it can exert large forces, yet if it is even slightly off the ground it cannot exert any forces. To determine the current contact mode as well as the confidence in that contact mode we present a momentum observer based estimator specifically designed for bipedal robots. This method for bipedal robot contact estimation seeks specifically to leverage assumptions about the robot dynamics to detect touchdown and liftoff without access to contact sensors or ground reaction force sensors.

The work in this thesis expands the capabilities of legged robots and manipulators through access to better state estimation. Without accurate state estimation, reasonable control actions cannot be taken, leading to issues in these systems such as falling over, not moving in the desired direction, dropping objects, or crushing objects.

6.1 Potential Extensions

Beyond the scope of this thesis, there are many opportunities for future work in the area of state estimation for hybrid dynamical systems. This section will introduce a few of these potential extensions.

6.1.1 Simultaneous or Near Simultaneous Impacts

One major place where this work can be extended is in the case where multiple hybrid transitions occur simultaneously or in quick succession. In cases with near simultaneous transitions, the ordering of the transition can significantly change the expected mean value of the post-impact state as well as the shape of the distribution. However, there are also cases where the ordering can result in insignificant changes in distributions. This presents two possible directions for analysis: first, one could seek to understand how to update distributions while taking into account the bipartite nature of the dynamics. These cases can be handled by considering the Bouligand derivative [20, 19] in place of the saltation matrix. Another direction would be to classify conditions in which the ordering of impacts do not cause notable variations in the resulting distributions. In these cases, the order of the combination of the reset maps and saltation matrices could be arbitrarily selected.

6.1.2 Information Seeking Control

A central theme throughout this work is the difficulty with estimating the current contact conditions, or discrete mode, of a system. Part of this difficulty could be offloaded to the control of these systems through information seeking control [73, 74, 75, 76]. An additional control goal/cost could be added to a controller to maximize the certainty in the hybrid mode. In the case of a legged robot, this could take the form of intentionally moving feet toward the ground to ensure contact is made.

6.1.3 System Design

An interesting challenge would be to consider the effects of hybrid transitions during the design of a system. In a design optimization framework, an additional penalty could be added to the distance of the saltation matrix from identity during typical contact events so that the saltation matrix can be safely ignored in control and estimation. Similarly, one could seek to maximize the convergence of the system by minimizing the eigenvalues of the saltation matrix in these common contact events [77, 10].

6.1.4 Applications to Simultaneous Localization and Mapping

Current implementations of simultaneous localization and mapping (SLAM) rely on process models driven by the integration of inertial measurement units (IMUs) and rarely consider the underlying robotic system's dynamics. In some systems, the kinematics are also leveraged [31, 57, 58], but this is still leaving out a lot of useful information from control inputs and a system's natural dynamics. Understanding the dynamics in continuous modes as well as during hybrid transitions with the reset map and saltation matrix would enable an additional factor in a factor graph to relate the states of a robot at each timestep.

Bibliography

[1] K. Shin and N. McKay, “A dynamic programming approach to trajectory planning of robotic manipulators,” *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, 1986.

[2] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[3] C. E. García, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[4] S. A. Burden, S. Revzen, and S. S. Sastry, “Model reduction near periodic orbits of hybrid dynamical systems,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2626–2639, 2015.

[5] A. M. Johnson, S. A. Burden, and D. E. Koditschek, “A hybrid systems model for simple manipulation and self-manipulation systems,” *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1354–1392, 2016.

[6] N. J. Kong, J. J. Payne, J. Zhu, and A. M. Johnson, “Saltation matrices: The essential tool for linearizing hybrid dynamical systems,” *Proceedings of the IEEE*, pp. 1–24, 2024.

[7] N. J. Kong, J. J. Payne, G. Council, and A. M. Johnson, "The Salted Kalman Filter: Kalman filtering on hybrid dynamical systems," *Automatica*, vol. 131, p. 109752, 2021.

[8] J. J. Payne, N. J. Kong, and A. M. Johnson, "The uncertainty aware Salted Kalman Filter: State estimation for hybrid systems with uncertain guards," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2022.

[9] H. Kumar, J. J. Payne, M. Travers *et al.*, "Periodic slam: Using cyclic constraints to improve the performance of Visual-Inertial SLAM on legged robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9477–9483.

[10] J. Zhu, J. J. Payne, and A. M. Johnson, "Convergent iLQR for safe trajectory planning and control of legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8051–8057.

[11] A. Back, J. M. Guckenheimer, and M. Myers, "A dynamical simulation facility for hybrid systems," in *Hybrid Systems*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1993, vol. 736, pp. 255–267.

[12] J. Lygeros, K. H. Johansson, S. N. Simic *et al.*, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 2–17, 2003.

[13] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, 2009.

[14] M. Aizerman and F. Gantmakher, "On the stability of periodic motions," *Journal of Applied Mathematics and Mechanics*, vol. 22, no. 6, pp. 1065–1078, 1958.

[15] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*. Springer, 1988.

[16] R. I. Leine and H. Nijmeijer, *Dynamics and bifurcations of non-smooth mechanical systems*. Springer Science & Business Media, 2013, vol. 18.

[17] G. Council, S. Revzen, and S. A. Burden, "Representing and Computing the B-Derivative of the Piecewise-Differentiable Flow of a Class of Nonsmooth Vector Fields," *Journal of Computational and Nonlinear Dynamics*, vol. 17, no. 9, 06 2022, 091004.

[18] A. M. Pace and S. A. Burden, "Piecewise-differentiable trajectory outcomes in mechanical systems subject to unilateral constraints," in *International Conference on Hybrid Systems: Computation and Control*, 2017, pp. 243–252.

[19] S. A. Burden, S. S. Sastry, D. E. Koditschek, and S. Revzen, "Event–selected vector field discontinuities yield piecewise–differentiable flows," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 2, pp. 1227–1267, 2016.

[20] S. Scholtes, *Introduction to piecewise differentiable equations*. Springer Science & Business Media, 2012.

[21] M. Bernardo, C. Budd, A. R. Champneys, and P. Kowalczyk, *Piecewise-Smooth Dynamical Systems: Theory and Applications*. Springer Science & Business Media, 2008, vol. 163.

[22] I. A. Hiskens and M. Pai, "Trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 2, pp. 204–220, 2000.

[23] A. Saccon, N. van de Wouw, and H. Nijmeijer, "Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking," in *IEEE Conference on Decision and Control*, 2014, pp. 3065–3070.

[24] A. Mathai and S. B. Provost, *Quadratic Forms in Random Variables*. Marcel Dekker, Inc., 1992.

[25] M. Rijnen, A. Saccon, and H. Nijmeijer, "On optimal trajectory tracking for mechanical systems with unilateral constraints," in *IEEE Conference on Decision and Control*, 2015, pp. 2561–2566.

[26] G. Welch and G. Bishop, "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Tech. Rep. 95–041, 1995, Updated: July 24, 2006.

[27] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[28] M. Biggio, F. Bizzarri, A. Brambilla, and M. Storace, "Accurate and efficient PSD computation in mixed-signal circuits: A time-domain approach," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 905–909, 2014.

[29] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*. Athena scientific, 2012, vol. 1.

[30] N. J. Kong, G. Council, and A. M. Johnson, "iLQR for piecewise-smooth hybrid dynamical systems," in *IEEE Conference on Decision and Control*, December 2021.

[31] M. Bloesch, M. Hutter, M. A. Hoepflinger *et al.*, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," in *Robotics: Science and Systems*, 2012, pp. 17–24.

[32] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 4399–4406.

[33] P. Varin and S. Kuindersma, “A constrained Kalman filter for rigid body systems with frictional contact,” in *Workshop on the Algorithmic Foundations of Robotics*, 2018.

[34] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, “Contact-aided invariant extended Kalman filtering for robot state estimation,” *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.

[35] Y. Gao, C. Yuan, and Y. Gu, “Invariant extended Kalman filtering for hybrid models of bipedal robot walking,” *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 290–297, 2021, Modeling, Estimation and Control Conference.

[36] J. C. Hinks, “A new generalized consider covariance analysis and other tools for challenging estimation scenarios,” Ph.D. dissertation, Cornell University, 2012.

[37] J. C. Hinks and M. L. Psiaki, “Generalized square-root information consider covariance analysis for filters and smoothers,” *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1105–1118, 2013.

[38] H. A. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[39] N. Barhoumi, F. Msahli, M. Djemaï, and K. Busawon, “Observer design for some classes of uniformly observable nonlinear hybrid systems,” *Nonlinear Analysis: Hybrid Systems*, vol. 6, no. 4, pp. 917–929, 2012.

[40] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, and A. Sangiovanni-Vincentelli, "The design of dynamical observers for hybrid systems: Theory and application to an automotive control problem," *Automatica*, vol. 49, no. 4, pp. 915 – 925, 2013.

[41] X. Koutsoukos, J. Kurien, and F. Zhao, "Monitoring and diagnosis of hybrid systems using particle filtering methods," in *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[42] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.

[43] E. J. Staunton and P. T. Piiroinen, "Discontinuity mappings for stochastic nonsmooth systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132405, 2020.

[44] S. A. Burden and S. D. Coogan, "On infinitesimal contraction analysis for hybrid systems," *IEEE Conference on Decision and Control (CDC)*, Dec. 2022.

[45] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951.

[46] W. J. Dixon and A. M. Mood, "The statistical sign test," *Journal of the American Statistical Association*, vol. 41, no. 236, pp. 557–566, 1946.

[47] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.

[48] F. Bailly, J. Carpentier, and P. Souères, “Optimal estimation of the centroidal dynamics of legged robots,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4912–4918.

[49] J. Zhang, A. M. Pace, S. A. Burden, and A. Aravkin, “Offline state estimation for hybrid systems via nonsmooth variable projection,” *Automatica*, vol. 115, p. 108871, 2020.

[50] N. Kong, C. Li, G. Council, and A. M. Johnson, “Hybrid iLQR model predictive control for contact implicit stabilization on legged robots,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4712–4727, 2023.

[51] Q. Li, R. Li, K. Ji, and W. Dai, “Kalman filter and its application,” in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77.

[52] C. Cadena, L. Carlone, H. Carrillo *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[53] A. Ypma and T. Heskes, “Iterated extended kalman smoothing with expectation-propagation,” in *2003 IEEE XIII Workshop on Neural Networks for Signal Processing (IEEE Cat. No.03TH8718)*, 2003.

[54] M. L. Psiaki, “Backward-smoothing extended Kalman filter,” *Journal of guidance, control, and dynamics*, vol. 28, no. 5, pp. 885–894, 2005.

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

[56] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, “State estimation for a humanoid robot,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 952–958.

[57] R. Hartley, J. Mangelson, L. Gan *et al.*, “Legged robot state-estimation through combined forward kinematic and preintegrated contact factors,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4422–4429.

[58] R. Hartley, M. G. Jadidi, L. Gan *et al.*, “Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3783–3790.

[59] M. Qin, Z. Yu, X. Chen *et al.*, “A novel foot contact probability estimator for biped robot state estimation,” in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2020, pp. 1901–1906.

[60] M. Camurri, M. Fallon, S. Bazeille *et al.*, “Probabilistic contact estimation and impact detection for state estimation of quadruped robots,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, 2017.

[61] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, “Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4193–4199.

[62] A. de Luca and R. Mattone, “Sensorless robot collision detection and hybrid force/motion control,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 999–1004.

[63] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1623–1630.

[64] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[65] J. K. Yim, J. Ren, D. Ologan *et al.*, "Proprioception and reaction for walking among entanglements," 2023.

[66] M. Bloesch, C. Gehring, P. Fankhauser *et al.*, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 6058–6064.

[67] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.

[68] Y. Kim, B. Yu, E. M. Lee *et al.*, "Step: State estimator for legged robots using a preintegrated foot velocity factor," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4456–4463, 2022.

[69] S. Yang, Z. Zhang, B. Bokser, and Z. Manchester, "Multi-imu proprioceptive odometry for legged robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[70] A. M. Johnson, G. C. Haynes, and D. E. Koditschek, “Disturbance detection, identification, and recovery by gait transition in legged robots,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5347–5353.

[71] S. Wang, A. Bhatia, M. T. Mason, and A. M. Johnson, “Contact localization using velocity constraints,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7351–7358.

[72] R. Tedrake and the Drake Development Team, “Drake: Model-based design and verification for robotics,” 2019. [Online]. Available: https://drake.mit.edu

[73] L. Valente, Y.-H. R. Tsai, and S. Soatto, “Information-seeking control under visibility-based uncertainty,” *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 339–358, 2014.

[74] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, “Decentralized active information acquisition: Theory and application to multi-robot slam,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4775–4782.

[75] S. Emoto, I. Akkaya, and E. A. Lee, “Information seeking and model predictive control of a cooperative multi-robot system,” *Artificial Life and Robotics*, vol. 21, no. 4, pp. 393–398, 2016.

[76] Y. Lu, J. Wang, and V. Kumar, “Curiosity driven self-supervised tactile exploration of unknown objects,” 2022.

[77] J. Zhu, N. J. Kong, G. Council, and A. M. Johnson, “Hybrid event shaping to stabilize periodic hybrid orbits,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 01–07.