ROBUST FORWARD INVARIANT SETS FOR NONLINEAR SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Shayok Mukhopadhyay

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2014

ROBUST FORWARD INVARIANT SETS FOR NONLINEAR SYSTEMS

Approved by:

Professor Fumin Zhang, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor David Taylor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Patricio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Michael Malisoff
Department of Mathematics
Louisiana State University

Date Approved: 27 June 2014

To my mother Rita,

my father Amit,

and my aunt Anita.

ACKNOWLEDGEMENTS

My time in graduate school has been made possible through the help of many people. I express my sincere gratitude towards all of them. I thank my Ph.D. advisor Dr. Fumin Zhang for providing me the opportunity to carry out research in his team, and supporting me throughout my journey. His scientific views and critiques have helped me get accustomed to the rigor of scientific research. Working with him has also helped me with my time management skills. I thank him for considering me for a variety of projects. This has contributed significantly to my learning experience.

I thank Dr. David Taylor, Dr. Patricio Vela, Dr. Yorai Wardi, and Dr. Michael Malisoff for being members of my Ph.D. dissertation committee. I thank them for spending their valuable time reading my thesis and providing comments. I thank Dr. Michael Malisoff for his detailed comments regarding portions of my work. His comments helped me get a very clear contrast between existing work in the literature and the work presented in this document.

I thank the School of Electrical and Computer Engineering and Georgia Institute of Technology for providing the environment, and the tools required for research. I am thankful to my fellow lab members, and the entire Georgia Tech Systems Research (GTSR) team for making my time in graduate school enjoyable. I have had the good fortune of being able to interact with the team through various projects, e.g. the field trip to Grand Isle, Louisiana in the summer of 2011. I have learnt a lot by interacting with the GTSR members in such a setting outside of school, and it is their company which made such experiences productive and successful.

Finally, I would like to thank my friends, and my family. Without them, this work would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The process of quantifying the robustness of a given nonlinear system is not necessarily trivial. If the dynamics of the system in question are not sufficiently involved, then a tight estimate of a bound on system performance may be obtained. As the dynamics of the system concerned become more and more involved, it is often found that using the results existing in the literature provides a very conservative bound on system performance. Therefore, the motivation for this work is to develop a general method to obtain a less conservative estimate of a bound on system performance, compared to the results already available in the literature. The scope of this work is limited to two dimensions at present. Note that working in a two dimensional space does not necessarily make the objective easily achievable. This is because quantifying the robustness of a general nonlinear system perturbed by disturbances can very easily become intractable, even on a space with dimension as low as two.

The primary contribution of this work is a computational algorithm, the points generated by which are conjectured to lie on the boundary of the smallest robust forward invariant set for a given nonlinear system. A well known path-planning algorithm, available in existing literature, is leveraged to make the algorithm developed computationally efficient.

If the system dynamics are not accurately known, then the above computed approximation of an invariant set may cease to be invariant over the given finite time interval for which the computed set is expected to be invariant. Therefore, the secondary contribution of this work is an algorithm monitoring a computed approximation of an invariant set. It is shown that for a certain type of systems, this secondary

monitoring algorithm can be used to detect that a computed approximation of an invariant set is about to cease to be invariant, even if the primary algorithm computed the set based on an unsophisticated dynamical model of a system under consideration.

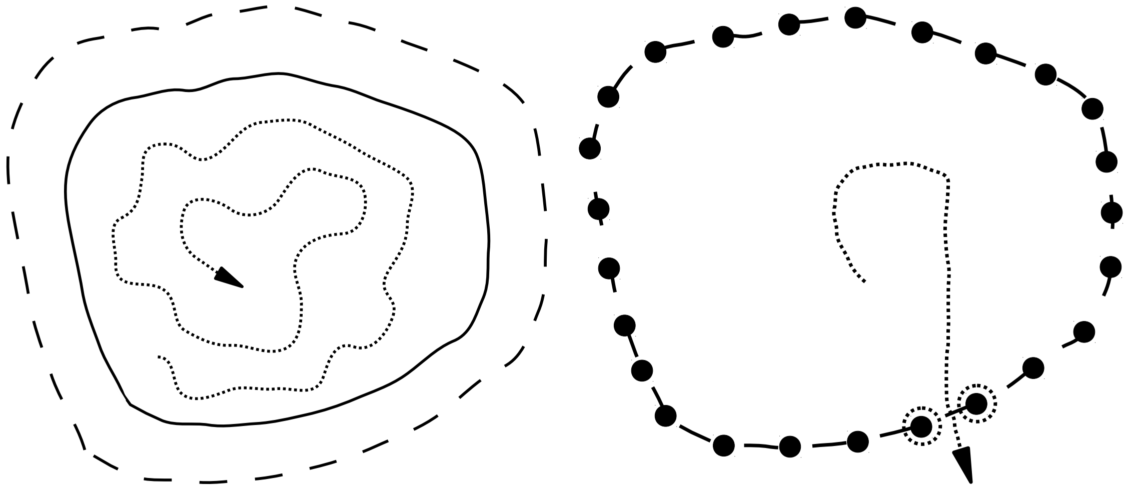
The work related to computing approximations of invariant sets is tested mainly with the curve tracking problem in two dimensions. The algorithm monitoring whether a computed approximation of an invariant set is about to cease to be invariant is inspired by work related to detecting Lithium-ion (Li-ion) battery terminal voltage collapse detection.

CHAPTER I

INTRODUCTION

A dynamical system can be used to model a particular physical phenomenon of interest. To model real-world phenomena, it is common to have systems affected by additive perturbations. Explicit performance bounds can help evaluate the performance of such systems affected by disturbances. The concept of input-to-state stability (ISS) can be used to obtain a performance bound for systems subjected to locally bounded time-varying disturbances. However, performance bounds established based on ISS can be very conservative. Compared to results based on ISS, it may be possible to establish more aggressive bounds on system performance by finding invariant sets. The shape and size of such an invariant set will depend on the properties of a system. However, there may be parametric uncertainties in a system model. Given an invariant set for a system without any parametric uncertainty, and the magnitude of the uncertainty, results from perturbation theory can be used to estimate the size of a set in which system trajectories remain in the presence of parametric uncertainties (see Figure 1(a)).

If such estimates of the size of an invariant set are only valid over a finite interval of time, then it is important to know when the estimates of the size of an invariant set for a given system do not hold. If the estimates of the size of an invariant set are valid for a given interval of time, then the invariant set may be said to be reliable over that interval. Similarly, if an invariant set ceases to be invariant within a given interval of time, then it can be said that the given invariant set is not reliable. Finding invariant sets that are valid for all time, which may require finding suitable Lyapunov functions, can be very involved for time varying nonlinear systems. Hence, estimates



(a) An invariant set for a given system shown by the solid curve. The dotted curve shows a system trajectory. The dashed curve shows an estimate of a set obtained for the given system when small parametric uncertainties appear in the system. For small bounded parametric uncertainty, the trajectory of the system with parametric uncertainty may remain within the set shown using dashes, if initial conditions lie within the set shown using a solid curve. But large disturbances/uncertainties may cause behavior as represented in the figure alongside.

(b) The solid black dots show some points that may need to be monitored to check if the trajectory gets close to the boundary. The dotted circles show an example of a region around boundary points which are monitored. If the number of points monitored is insufficient, or the region monitored is not large enough then the system trajectory (dotted curve) may leave a computed set and the monitoring algorithm may fail to detect it.

Figure 1: An illustration of invariant sets computed algorithmically and the effects of changes in system behavior on the reliability of a computed set.

of the size of an invariant set that are valid for all time may not always be found. In such a situation, it may be helpful to know if the estimated size of an invariant set, which is only valid over a finite interval of time, is about to cease to be reliable. Thus motivated, this work has the following goals. The first goal is the development of a computational algorithm to find robust forward invariant sets (RFISs) for a system with bounded additive disturbances. The second goal is to develop an algorithm for detecting that a computed robust forward invariant set (RFIS) is about to cease to be invariant.

The computational load involved in monitoring a computed RFIS can be reduced if certain knowledge regarding system behavior is taken into account. For example, points on the boundary of an RFIS can be monitored to check if system states are about to breach the boundary (see Figure 1(b)). This approach is computationally expensive even for monitoring a finite number of points on the boundary, because the demand on the computational resources grows with every new point on the boundary that requires monitoring. If there is a way to determine that system behavior has changed drastically so as to cause an impending system instability, this information can be useful to reduce the computational load. Monitoring system states is then enough to determine that a computed RFIS is at risk of ceasing to be invariant. However, a concern with this approach may be the accuracy of the available system model. To mitigate such problems, this work leverages research related to detecting Lithium-ion (Li-ion) battery terminal voltage collapse detection. This approach relies on tools from adaptive control theory, and relaxes the requirement of a sophisticated system model for monitoring the validity of a computed RFIS. This allows detecting that a computed RFIS is about to cease to be invariant, even if the model used to compute such an RFIS is not very accurate. Another concern may be that as the number of states to be monitored increases, the above mentioned approach utilizing tools from adaptive control theory may not be very effective in reducing computational load. In reality, a system with a finite number of states is usually sufficient to model a phenomenon of interest, and it is reasonable to expect that the number of system states is less than the number of distinct points on the boundary of an RFIS which need monitoring.

The focus of this work is on the development of computational algorithms because closed form analytical expressions for performance bounds may be unattainable for complex systems. Systems with fractional dynamics, and nonlinear model predictive control systems, are some examples of such complex systems. Generalization of the algorithms developed in this work to higher dimensional systems is left for future efforts.

This document is organized as follows. A review of literature related to this work is presented in Chapter 2, which helps clarify the motivation for this work and provides the contrast needed to evaluate the results presented. Chapter 3 is focused on developing a computational algorithm for finding the smallest RFIS for a given nonlinear system affected by bounded additive disturbances. Chapter 4 presents some ideas that are related to Li-ion battery terminal voltage collapse. These ideas serve as a foundation for the material presented in Chapter 5. Chapter 5 shows that perturbation theory may be used to generate estimates of the size of a set in which the trajectories of a perturbed system with bounded parametric uncertainty stay, given initial conditions belonging to an RFIS of a nominal perturbed system without parametric uncertainty. Further, Chapter 5 uses the strategies presented in Chapter 4, which were originally developed for detecting Li-ion battery terminal voltage collapse, to detect if a computed RFIS is about to cease to be invariant in the event of a loss of system stability. Chapter 6 summarizes the results produced, and provides a glimpse into future efforts.

CHAPTER II

BACKGROUND

Finding the least conservative performance bound for the problem of curve tracking control [85] is a motivation to develop a computational algorithm to find an RFIS. As mentioned in [43], the dynamics of the interaction between a unit speed mobile robot and its projection on a curve which the robot is trying to track can be represented using the following nonlinear system of equations:

$$\begin{aligned} \dot{\rho}(t) &= -\sin(\phi(t)) & (1) \\ \dot{\phi}(t) &= \frac{\kappa \cos(\phi(t))}{1+\kappa\rho(t)} - u(t). & (2) \end{aligned}$$

Here the scalar $\rho(t) = \| \mathbf{r}(t) - \bar{\mathbf{r}}(t) \|_2$. Vector $\mathbf{r}$ represents the position of a mobile robot on a plane. The two dimensional vector $\bar{\mathbf{r}}$ represents the position of the closest point on a given curve, with respect to the position of the robot. Scalars ϕ, κ, and u are the bearing, the positive curvature, and the control effort, respectively. As shown in [83], it is possible to design a controller to achieve asymptotic stabilization so the distance converges to a desired positive constant ρ_0, and the bearing converges to zero. Such controllers have been used in real-world applications involving obstacle avoidance using wheeled robots and marine sampling [50, 83]. The controllers appear quite robust to real world disturbances. This motivated the authors in [43, 44] to justify the observed robustness using tools like input-to-state stability (ISS). Suppose the control $u(t)$ in Equation (2) is perturbed by an additive disturbance $\delta(t)$ valued in an interval $[-\delta_0, \delta_0]$, where δ_0 is a real positive constant. An analytical method for constructing robust forward invariant hexagons for such systems is provided in [43,44]. For linear systems, extensive research exists related to problems which are similar to the problem finding an RFIS [37, 65, 66]. The problem of finding an RFIS is similar to the problem of finding invariant sets, or estimating the region of attraction.

The problem of estimating the domain of attraction of an autonomous nonlinear system having the form $\dot{x} = f(x)$, with $f(0) = 0$, has seen a variety of research over an extended period of time [74]. Quite naturally, there are many ways to find the domain of attraction for a given system. Zubov's method [87] consists of solving a partial differential equation, known as the Zubov equation, for computing the domain of attraction. Davison [20] provides a numerical method to obtain a quadratic Lyapunov function so that the volume of the asymptotic stability region is maximized. In [42], the authors use Carlemann linearizations to get a sequence of sets which converge to the domain of attraction. This method is applicable to systems where $f(\cdot)$ is analytic. The authors in [47] use computer generated Lyapunov functions to estimate the region of attraction. In [55], Luré type Lyapunov functions are constructed using a theorem of Popov [10]. The Lyapunov function thus obtained is used to estimate the region of asymptotic stability for a post-fault power system. Similar analysis using Lyapunov functions is found in [77, 79]. The authors in [75, 76] explore the problem of finite regions of attraction.

Concepts similar to those reviewed above have been applied to perturbed systems. However, analytical solutions for perturbed nonlinear systems are not always easily found [32,46]. Numerical analysis is favored in such situations [8,31]. Some references in [26] describe topics related to numerical analysis of dynamical systems. Numerical methods have been used to study objects like invariant manifolds [9], attracting sets [35], homoclinic orbits [7], and Morse-Smale systems [22]. Any type of numerical analysis involves discretization. This creates concerns about the closeness of the numerically achieved results with respect to the true analytical results. Such issues are studied in [25]. In [12], the authors extend Zubov's method to perturbed dynamical systems.

A large body of literature exists on determining Lyapunov functions whose sublevel sets can be used to characterize the region of attraction [23]. Research focused on sum-of-squares relaxations for polynomial optimization [57] has been utilized by the authors in [71] for estimating the region of attraction. The work of the authors in [15, 63] uses parameter dependent Lyapunov functions to obtain less conservative estimates of the region of attraction as compared to approaches using parameter independent Lyapunov functions. However, the parameter dependent approaches usually have higher computational complexity compared to parameter independent approaches. The work [73] follows the work [72] closely. This work uses the sum-of-squares optimization method together with parameter independent Lyapunov functions, and branch-and-bound [5,82] type refinement algorithms, to estimate the region of attraction for dynamical systems with bounded parametric uncertainty. However, this procedure requires a family of candidate Lyapunov functions. Even though there exist specialized methods for constructing Lyapunov functions [45], obtaining Lyapunov functions for any given dynamical system can be challenging. Obtaining an entire family of Lyapunov functions, as required in [73], is not necessarily an easy task. There has been some work specifically directed toward computing robust control invariant sets [2], but this work focuses exclusively on piecewise affine systems. The authors in [4] use Newton's method and the secant method to find zeros for set-valued maps. This requires appropriately redefining a given problem in a Banach space setting, and redefining Fréchet derivatives for set valued maps. Such approaches, as in [4], have been used to find invariant sets for systems in [3]. However, Newton's method does not perform well for ill-behaved problems. If the initial guess is not sufficiently close to the required solution, an algorithm based on Newton's method may not converge.

Regardless of the method used to compute an invariant set, it is important to decide if a computed set is valid when parametric uncertainties affect a system. Bounds on the error between solutions of a nominal system and that of a perturbed system can be established using results from perturbation theory [32]. Detecting that a computed invariant set is valid when subjected to parametric uncertainties is conceptually similar to monitoring a dynamical system for bifurcations. The qualitative behavior of a dynamical system may change if a particular parameter changes. Such a change is known as a bifurcation, and the points at which such changes occur are known as bifurcation points [30, 70]. When a bifurcation occurs, fixed points may vanish or be created. Sometimes, fixed points switch from being stable to unstable as a result of a bifurcation. When such switches occur, a set that is invariant for a nominal system may no longer be invariant when the system is affected by perturbations. Knowing when a system is about to undergo a bifurcation is important. For example, the authors in [6] incorporate information obtained from bifurcation analysis to arrive at a better estimate of the domain of attraction for a nonlinear system. In [81], the authors use a probe signal to detect an impending bifurcation for a power system. In [48], the authors use a frequency domain approach to detect period doubling bifurcations. The concept of detecting changes in system behavior near a bifurcation point can be used to detect if a computed invariant set is at risk of losing its property of being invariant. This idea is inspired by previous work related to detecting Lithium ion (Li-ion) battery terminal voltage collapse [51, 52].

High charge densities [40] make Li-ion batteries popular. Thermal failure, loss of charge, and voltage collapse are some problems associated with batteries. The terminal voltage of a battery drops sharply from its operating value when it is in a low state of charge (SoC) [40]. The terminal voltage, available capacity, state of charge (SoC), and state of health (SoH) [62] can be used to determine that the terminal voltage of a battery is about to collapse. A constant threshold voltage is often used to determine that a particular battery is discharged [33, 62]. The terminal voltage of a battery depends on the discharge current it supplies. Hence, using a constant voltage threshold can lead to false alarms in the presence of noise, large spikes, or rapid changes in the discharge current. A threshold on the battery state of charge (SoC) can also detect that the terminal voltage of a battery is about to collapse. Determining the SoC involves integrating the measured discharge current. Measurement errors can introduce errors in the SoC computation [62]. Incorporating battery models improves the accuracy of detecting an impending terminal voltage collapse. Various types of battery models [14, 67] and associated identification techniques exist [1, 68]. Run-time models [24], which primarily consist of look up tables providing the battery terminal voltage, are the simplest battery models. Mathematical models for digital systems [58] have been considered for maximizing battery life, or reducing switching latency in digital circuits. Mathematical models do not necessarily translate to practical applications. Stochastic battery models [16], and detailed electrochemical models [34], have been investigated to understand the recovery effect in detail. Using state estimators for such detailed models results in excessively long computation times. To enable fast computation, some recent efforts have focused on simplifying models describing battery thermal runaway [69]. However, solving partial differential equations is still not tractable for small onboard robotic systems.

Therefore, simple dynamic models, which capture essential physical phenomenon, coupled with filtering algorithms [59–61] for state estimation, are popular methods used to detect battery terminal voltage collapse. Fault detection strategies like residual generation [21] can also be used to detect that a battery is about to die. All the above methods require detailed battery models. Substantial time and effort [18] is needed to obtain such models. In reality, battery characteristics may differ from the model used. This may result in voltage collapse before a particular algorithm detects it. Battery characteristics may also change within a span of a few days or months. This may necessitate frequent investment of massive amounts of time and effort into modeling.

The aim is reducing model dependence, while being able to detect that the terminal voltage of a battery is about to collapse. Reducing the effects of load changes, temperature changes, discharge current spikes, and measurement noise on terminal voltage collapse detection, is also desired. Concepts from universal adaptive stabilization (UAS) [28, 49] can be used to extend the work [52] related to Li-ion battery terminal voltage collapse detection, to detect that an RFIS computed for a nominal system is invalid for a perturbed system.

When contrasted with existing literature on computing invariant sets, the work in this document for computing an approximation of an RFIS does not use Newton-type methods for estimating a region of attraction. Methods used to estimate the entire domain of attraction are not extended. Inspired by the notion of Hamiltonians in [17], a new approach is developed to compute an approximation of an RFIS for a given system. This work does not follow Lyapunov based techniques because the invariant sets computed may turn out to be the sub-level sets, hence offering conservative results. An approximation of the smallest RFIS contained in a given domain of interest may be computed by considering the ensemble of all trajectories of a given system. System trajectories can be obtained by solving differential equations modeling a given system, with initial conditions at every point in the domain of interest. The smallest region in the domain of interest in which system trajectories remain for all time, may be one way to find an approximation of the smallest RFIS. With such an approach, the computation time increases as the number of different initial conditions to be tried increase. This work uses an approach which does not require solving for system trajectories. The novelty of the approach developed in this work is that the problem of finding the boundary of an approximation of the smallest RFIS is formulated as a path planning problem [53], and it is hypothesized that the optimal path will be a good approximation of the desired boundary. Path planning algorithms such as the A^* [27] algorithm, are well known to reduce the amount of computation required to find optimal paths for robot navigation. Conditions are derived under which the proposed algorithm terminates. Previous results related to Li-ion battery terminal voltage collapse detection [52] are used to develop an algorithm for monitoring the validity of the computed result, when the original system is subjected to parametric uncertainty.

CHAPTER III

COMPUTING APPROXIMATIONS OF THE SMALLEST ROBUST FORWARD INVARIANT SETS

Robustness of nonlinear systems can be analyzed by computing robust forward invariant sets (RFISs). The smallest RFIS provides the least conservative estimate of system performance under perturbations. In this chapter, a novel algorithm is developed to find an approximation of the smallest RFIS for two-dimensional systems subjected to a bounded additive disturbance. The algorithm developed leverages a path planning algorithm. The required notations and definitions are presented first, followed by a formulation which transforms the problem of computing an RFIS into a path planning problem. Further, the algorithm is mathematically justified, and simulation results are provided showing that the algorithm developed can be used to find an approximation of the smallest RFIS. Since efficient path planning algorithms are used, the amount of computation is effectively reduced. Hence it may be feasible to generalize the algorithm to higher dimensional systems in future.

3.1 Mathematical preliminaries

This section provides the definitions required to describe the problem setup.

Consider the system $\dot{x}(t) = \vartheta(x(t), \delta(t))$. Here $\vartheta : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$, and $\delta : \mathbb{R} \to \mathbb{R}^2$ is a time varying bounded disturbance such that standard conditions for existence and uniqueness of solutions are met. Let $\bar{x} \in \mathbb{R}^2$, and $\vartheta(\bar{x}, 0) = 0$. For this work, it is assumed that the dynamics can be written in the form $\vartheta(x(t), \delta(t)) = f(x(t)) + \delta(t)$, where $f : \mathbb{R}^2 \to \mathbb{R}^2$, and the disturbance function $\delta : [0, +\infty) \to \mathcal{U}$. The set $\mathcal{U}$ is defined as $[-\delta_0, \delta_0] \times [-\delta_0, \delta_0]$, for some known value $\delta_0 \in [0, +\infty)$. The set $\mathcal{U}$ is therefore a bounded box in two-dimensions. Let $\mathcal{M}_{\mathcal{U}}$ denote the set of all measurable, and locally essentially bounded functions $\delta : [0, +\infty) \to \mathcal{U}$. If the dynamics under a different disturbance function belonging to $\mathcal{M}_{\mathcal{U}}$ are desired to be studied, then the function $\delta(t)$ can be replaced with such a desired function from the set $\mathcal{M}_{\mathcal{U}}$. Such notation is used because it may help extend this work to find an approximation of an RFIS for a differential inclusion [17].

As per [44], an RFIS can be defined for an undelayed system in $\mathbb{R}^2$ as follows. Take any open subset $\mathcal{X}$ of $\mathbb{R}^2$, and any point $\mathcal{E} \in \mathcal{X}$. Consider a forward complete system $\dot{x}(t) = \mathcal{F}(x(t), \delta(t))$ with state space $\mathcal{X}$ and perturbations $\delta \in \mathcal{M}_{\mathcal{U}}$ such that standard existence and uniqueness properties of solutions are satisfied for all initial conditions in $\mathcal{X}$ and all perturbations $\delta \in \mathcal{M}_{\mathcal{U}}$. Let $\mathcal{F}(\mathcal{E}, 0) = 0$. Let $\mathcal{S} \subseteq \mathcal{X}$ be any neighborhood of $\mathcal{E}$. The set $\mathcal{S}$ is an RFIS for the system $\dot{x}(t) = \mathcal{F}(x(t), \delta(t))$ with perturbations valued in $\mathcal{U}$ if all trajectories of the above system, for all initial conditions valued in $\mathcal{S}$ and all perturbations $\delta \in \mathcal{M}_{\mathcal{U}}$ remain in $\mathcal{S}$ for all positive times. Furthermore, if all trajectories of the above system $\dot{x}(t) = \mathcal{F}(x(t), \delta(t))$, for all initial conditions $x(t_0)$ valued in $\mathcal{S}$ and all perturbations $\delta \in \mathcal{M}_{\mathcal{U}}$ remain in $\mathcal{S}$ for all time $t \in [t_0, t_1]$ where $t_1 > t_0 \geq 0$, then it is said that $\mathcal{S}$ is an RFIS for the interval $[t_0, t_1]$.

Let $\mathcal{D} \subseteq \mathbb{R}^2$ be a region of interest, and let all simple closed curves in $\mathcal{D}$ be positively oriented. Let P and Q be two sequences of equal length consisting of points p_i, and q_j respectively from Euclidean space. Define the distance between sequences P and Q as $d_2(P, Q) = \sum_{l=1}^{n} \parallel p_l - q_l \parallel_2$, where $p_l \in P$, and $q_l \in Q$.

Given two points a and b belonging to Euclidean space, let $\operatorname{seg}(a; b)$ denote the segment joining points a, and b. The following definitions common to convex analysis are presented from [19], [56]. A *cone* $\mathscr{C} \in \mathbb{R}^n$ is defined as a set of points $x \in \mathbb{R}^n$ such that, if $x \in \mathscr{C}$, then for all non-negative $\lambda \in \mathbb{R}$, $\lambda x \in \mathscr{C}$. A cone $\mathscr{C} \in \mathbb{R}^n$ is a *convex cone* if and only if, (a) for $x \in \mathscr{C}$, $\lambda x \in \mathscr{C}$ for all non-negative $\lambda \in \mathbb{R}$; (b) if $x_1, x_2 \in \mathscr{C}$,

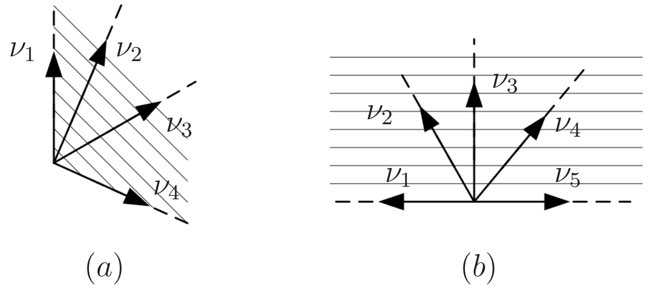


Figure 2: Types of convex cones: (a) A pointed convex cone. For any given vector, a pointed convex cone does not contain the entire line passing through the origin and the given vector. (b) A convex cone which is not pointed since it contains the entire line formed by vector ν_1 or ν_5.

then $x_1 + x_2 \in \mathscr{C}$. A convex cone $\mathscr{C}$ is said to be *pointed* if, given any arbitrary vector $\bar{\mathbf{a}} \in \mathbb{R}^n$ it contains no line $\mathscr{L} = \{x : x = \lambda\bar{\mathbf{a}} \text{ for all } \lambda \in \mathbb{R}, x \in \mathbb{R}^n\}$. A non-zero vector $\mathbf{a}$ in a pointed-convex cone $\mathscr{C}$ is an *extreme ray* if and only if $\alpha\nu_1 + \beta\nu_2 \neq \mathbf{a}$ for all $\alpha, \beta \geq 0$, for all $\nu_1, \nu_2 \in \mathscr{C} \setminus \{\lambda\mathbf{a} | \lambda \geq 0\}$. In other words, an extreme ray of a pointed convex cone is a vector that cannot be expressed as a non-negative linear combination of vectors in the cone which are distinct from it. The extreme ray can be thought of as a boundary vector of a pointed convex cone.

The following definitions are related to path planning on directed graphs [27]. A *graph* is defined to be a set $\{p_{ij}\}$ of elements called *nodes* and a set $\{(p_{ij}, p_{kl})\}$ of pairs called *arcs*. Here i, j, k, and l belong to some bounded subset of $\{0\} \cup \mathbb{N}$ (the set of natural numbers along with the number zero), and each arc is directed from node p_{ij} to node p_{kl}. Each node p_{ij} has two indices i, j representing the physical location of a node $p_{ij} \in \mathbb{R}^2$. Given an arc (p_{ij}, p_{kl}), node p_{kl} is a *successor node* of node p_{ij}, and the nodes p_{ij}, and p_{kl} are *connected*. Node p_{kl} is said to be *accessible* from p_{ij}. The *cost* of an arc (p_{ij}, p_{kl}) is represented by the scalar $c(p_{ij}, p_{kl})$. The exact definition of the quantity $c(\cdot, \cdot)$ is provided later. A *path* is an ordered set of nodes with each node $p_{(i+1)k}$ a successor node of node p_{ij}. Indices j, and k do not depend on each other or on index i. Every path has a cost obtained by adding the individual costs for each arc in the path. An *optimal path* from p_{ij} to p_{kl} is a path having the smallest cost over the set of all paths from p_{ij} to p_{kl}. Some specified non-empty set $S \subset \{p_{ij}\}$ is known as the *source set*. A single specified node p_{ij} belonging to a given source set S is known as the *source node*. Given a graph formed by a set of nodes $\{p_{ij}\}$ and a set of arcs $\{(p_{ij}, p_{kl})\}$, a *goal set* T is a non-empty set of nodes $T \subset \{p_{ij}\}$ accessible from some specified source node in $\{p_{ij}\}$. Let the node p_{kl} belong to a particular goal set. The node p_{kl} is a *preferred goal node* of a particular source node p_{ij} if the cost of an optimal path from p_{ij} to p_{kl} does not exceed the cost of any other path from p_{ij} to any other member of the chosen goal set.

3.2 Formulating the problem of computing an invariant set

In this section the problem of finding the smallest RFIS is formulated as an optimization problem that may be solved by applying path planning algorithms.

3.2.1 Problem statement

Using the notations and definitions in Section 3.1 the problem of interest to this work is stated as follows. Consider a system of the form $\dot{x}(t) = \vartheta(x(t), \delta(t))$ with a unique asymptotically stable equilibrium $\bar{x}$ i.e. $\vartheta(\bar{x}, 0) = 0$, and a given constant $\delta_0 \in [0, +\infty)$. The disturbance $\delta(\cdot)$ is allowed to vary over the set $\mathcal{M}_{\mathcal{U}}$, where $\mathcal{M}_{\mathcal{U}}$ is as defined in Section 3.1. Time $t \in [0, T_{\max}]$, where $T_{\max} \in \mathbb{R}$ and $T_{\max} > 0$. Starting from an invariant set $B \subseteq \mathcal{D}$ which contains the equilibrium $\bar{x}$, find a set contained in B which is an RFIS for all time $t \in [0, T_{\max}]$, which contains the equilibrium $\bar{x}$, and whose boundary is a positively oriented simple closed curve. Additionally, the length of a ray beginning at $\bar{x}$ and terminating at a point on the boundary (simple closed curve) is minimized for all angles $\theta \in [0, 2\pi]$, where θ represents the angle between a ray beginning at $\bar{x}$ and the x-axis. Such a set is referred to as a ***smallest RFIS*** in this work. Here $T_{\max}$ is introduced because the computational approach developed in this work samples the vector field at discrete points in time and space. Given a general time varying vector field of the form $\dot{x}(t) = \vartheta(x(t), \delta(t))$, since the algorithm developed in this work does not have any additional knowledge about the behavior of a given system, the algorithm can only be used to comment on invariance within the finite time interval in which the vector field is sampled. Also, the reason to introduce the set B is that the algorithm developed will have to start from a initial guess of the smallest RFIS, which is represented by the set B.

Expected solutions to the above problem are shown in Figure 3(a), where the dashed square box represents an initial guess, the small dotted ellipse shows the true smallest RFIS, the simple closed curve (solid boundary) enclosing the ellipse represents an optimal solution, and the simple closed curve (dashed) enclosing the optimal solution, represents a possible sub-optimal solution (a more conservative estimate of an invariant set). If the exact shape of the smallest RFIS is to be found using computational techniques, then an infinitely high spatial and temporal resolution may be necessary. Hence, the goal is to find the closest approximation of the boundary of the smallest RFIS.

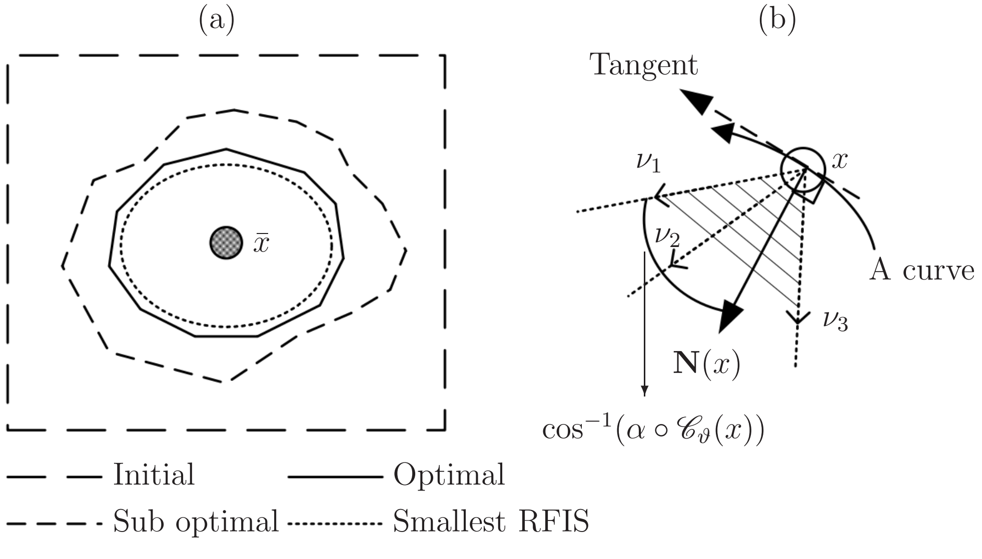


Figure 3: Illustrations: (a) Types of solutions to the problem of finding an RFIS. (b) The quantity $\alpha \circ \mathscr{C}_\vartheta(x)$ for a pointed convex cone at a point x.

3.2.2 Conversion into an optimization problem

Suppose the vector field $\vartheta : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ and a pointed convex cone $\mathscr{C}(x)$ with vertex at particular point $x \in \mathbb{R}^2$ are given. Let $\mathscr{C}_\vartheta(x)$ denote a pointed convex cone with vertex at a given point $x \in \mathbb{R}^2$ such that, for all $\nu \in \mathscr{C}_\vartheta(x)$, $\nu \in \text{Image}(\vartheta(x, \cdot))$. Let a vector $\mathbf{N}(x)$ be given at the point x. The quantity α for a given pointed convex cone $\mathscr{C}_\vartheta(x)$, at the point $x \in \mathbb{R}^2$ is written as $\alpha \circ \mathscr{C}_\vartheta(x)$, and defined as

$$\begin{aligned} \alpha \circ \mathscr{C}_\vartheta(x) &= \frac{\langle \mathbf{N}(x), \nu^\star \rangle}{\| \mathbf{N}(x) \| \, \| \nu^\star \|}, \text{ where} \\ \nu^\star &= \operatorname*{arg\,min}_{\nu \in \mathscr{C}_\vartheta(x)} \frac{\langle \mathbf{N}(x), \nu \rangle}{\| \mathbf{N}(x) \| \, \| \nu \|}. \end{aligned} \tag{3}$$

Here $\cos^{-1}(\alpha) \in [-\pi, \pi]$. If $\alpha \circ \mathscr{C}_\vartheta(x) \geq 0$, then it is straightforward to see that $\nu^\star$ is one of the extreme rays of a cone $\mathscr{C}_\vartheta(x)$. Before proceeding further, it is worthwhile to consider the usage of the set $\mathcal{U} = [-\delta_0, \delta_0] \times [-\delta_0, \delta_0]$. The set $\mathcal{U}$ is used as follows for computing an RFIS. The set $\mathcal{U}$ is discretized, and at every point $x \in \mathcal{D}$ a convex cone formed by vectors belonging to a given vector field $\vartheta(\cdot, \cdot)$ is considered for every discrete value of disturbance in $\mathcal{U}$. Then the angles between the extreme rays of such a cone and a given candidate normal vector $\mathbf{N}(x)$ are computed. An illustration of the above discussion is provided in Figure 3(b), where the cosine of the angle between vectors ν_1, and $\mathbf{N}(x)$ is the least when compared with the angle between $\mathbf{N}(x)$ and the other extreme ray (vector) ν_3. The idea is that the vector $\mathbf{N}(x)$ is a normal vector to some curve at the given point x. One such curve may be the boundary of the smallest RFIS.

Consider Figure 3(b) and the definition of the quantity $\alpha \circ \mathscr{C}_\vartheta(x)$. Further consider a positively oriented simple closed curve $\mathscr{R}$ such that all points in $\mathscr{R}$ are contained in the domain of interest $\mathcal{D} \subseteq \mathbb{R}^2$. Let $\mathbf{N}(x)$ represent the normal vector at a point x belonging to $\mathscr{R}$ such that at every point $x \in \mathscr{R}$, vector $\mathbf{N}(x)$ points towards the interior of the region enclosed by $\mathscr{R}$. Suppose at every point $x \in \mathscr{R}$ and for all time $t \in [0, T_{\max}]$, the quantity $\alpha \circ \mathscr{C}_\vartheta(x) \geq 0$ for all vectors $\nu \in \mathscr{C}_\vartheta(x)$ such that

$\nu \in \mathrm{Image}(\vartheta(x,\cdot))$. This means that for all $t \in [0, T_{\max}]$, at all points x belonging to $\mathscr{R}$, all vectors $\nu \in \mathscr{C}_\vartheta(x)$ such that $\nu \in \mathrm{Image}(\vartheta(x,\cdot))$ either point inside the closed region enclosed by $\mathscr{R}$, or point along a tangent to $\mathscr{R}$. This further means that solutions to $\dot{x}(t) = \vartheta(x(t), \delta(t))$ will stay in the region enclosed by $\mathscr{R}$ for all $t \in [0, T_{\max}]$. This is because for a solution to leave the region enclosed by $\mathscr{R}$, there must exist at least one point $x \in \mathscr{R}$ with at least one vector $\nu \in \mathscr{C}_\vartheta(x)$ such that $\nu \in \mathrm{Image}(\vartheta(x,\cdot))$, and an instant of time $t \in [0, T_{\max}]$ such that ν does not point into the interior region enclosed by $\mathscr{R}$, and ν does not point along a tangent to $\mathscr{R}$. However no such vectors ν exist, if for all $t \in [0, T_{\max}]$ the quantity $\alpha \circ \mathscr{C}_\vartheta(x) \geq 0$ for all vectors $\nu \in \mathscr{C}_\vartheta(x)$ such that $\nu \in \mathrm{Image}(\vartheta(x,\cdot))$. All points $x \in \mathscr{R}$ form the boundary of an RFIS, and therefore the problem of finding an RFIS (which is not necessarily the smallest RFIS) can now be formulated as the following problem. Find all $x^\star \in \mathcal{D}$ satisfying

$$x^\star = \operatorname*{arg\,min}_{x \in \mathcal{D}} \ \alpha \circ \mathscr{C}_\vartheta(x), \ \text{ s.t. } \ \alpha \circ \mathscr{C}_\vartheta(x) > 0. \tag{4}$$

The above is a constrained optimization problem because solutions x are desired to minimize the quantity $\alpha \circ \mathscr{C}_\vartheta(x)$ subject to certain constraints. The strict inequality constraint $\alpha \circ \mathscr{C}_\vartheta(x) > 0$ is used due to the following reason. If the constraint $\alpha \circ \mathscr{C}_\vartheta(x) \geq 0$ is used, then it is possible that a solution $x^\star$ to Equation (4) is obtained on the boundary of an RFIS, and $\alpha \circ \mathscr{C}_\vartheta(x) = 0$. This implies that solutions to $\dot{x}(t) = \vartheta(x(t), \delta(t))$ are tangential to the boundary of an RFIS. Hence, if the initial conditions of $\dot{x}(t) = \vartheta(x(t), \delta(t))$ equal $x^\star$, then solutions may leave the RFIS. Solutions to Equation (4) may be obtained by trying all points $x \in \mathcal{D}$ and then picking x which minimize $\alpha \circ \mathscr{C}_\vartheta(x)$ subject to the constraints in Equation (4). Such a brute force approach, which involves trying every point, is inefficient.

Hence the following formulation is used to transform the constrained optimization problem in Equation (4) into an equivalent unconstrained one. Pick a positive constant $\widetilde{g}$ that is sufficiently large (e.g. satisfying the conditions in Lemma 3.4.1 that will be derived later), and consider the function $\eta : [-1, 1] \to (0, 1] \cup \widetilde{g}$ defined as below.

$$\eta(x) = \begin{cases} x, & x \in (0,1] \\ \widetilde{g}, & x \in [-1,0] \end{cases} \tag{5}$$

Now, the problem in Equation (4) can be reformulated as,

$$x^\star = \underset{x \in \mathcal{D}}{\arg\min}\ (\eta \circ \alpha \circ \mathscr{C}_\vartheta)(x). \tag{6}$$

The resulting problem in Equation (6) is not necessarily convex, but assigning infeasible solutions a high cost $\widetilde{g}$ using the function $\eta(\cdot)$ helps recast the problem in Equation (6) into a path planning problem. Since the smallest RFIS around an equilibrium point $\bar{x}$ is desired, the problem in Equation (6) is modified to include a cost related to the size of the solution set, which accounts for the distance of a point $x \in \mathcal{D}$ to the equilibrium point $\bar{x}$, as follows

$$x^\star = \underset{x \in \mathcal{D}}{\arg\min}\ ((\eta \circ \alpha \circ \mathscr{C}_\vartheta)(x) + \lambda_1 \parallel x - \bar{x} \parallel_2). \tag{7}$$

Here $\lambda_1 > 0$ is a real weight which is fixed to be a desired number.

3.2.3 A path planning problem

We wish to find a solution to the problem in Equation (7) that is a closed path going around the equilibrium, and forming a positively oriented simple closed curve which approximates the boundary of an RFIS. To enable the algorithm to search for the required shape of the boundary of the smallest RFIS, we start from a convex set B with a simple closed curve as its boundary such that the equilibrium $\bar{x}$ belongs to the interior of B. We use B as an initial guess for the desired RFIS, so the set B must itself be invariant. It is also assumed that such a set B exists. Sub-level sets of strict Lyapunov functions serve as good choices for B, so techniques from [45] may be helpful to initialize the algorithm presented in this work. The following sections describe the setup required to apply the A^* path planning algorithms to compute a solution to the problem in Equation (7).

3.2.3.1 Discretization of the search space

The first step is to discretize the state space in the interior of set B into radial grid points. Select a collection of $N+1$ distinct points from the boundary of B denoted by $\{b_0, b_1, \cdots, b_N\}$. Then connect the equilibrium point $\bar{x}$ to each of these points $b_0, b_1, \cdots, b_N$. Let the index $i \in \{0, 1, \cdots, N\}$. For simplicity, choose b_i so that the angle between each segment $\operatorname{seg}(b_i; \bar{x})$ and the x-axis is $\frac{2\pi i}{N+1}$. This way, the angle between each segment $\operatorname{seg}(b_i; \bar{x})$ and the x-axis belongs to the interval $[0, 2\pi]$ by construction. The points b_0, b_1, b_2, and b_N are indicated on the boundary of a square set in Figure 4(a). The set B is convex by choice, and $\bar{x}$ is in the interior of B. Therefore B contains all line segments joining $\bar{x}$ to each point b_i by construction. Next, discretize each segment $\operatorname{seg}(b_i; \bar{x})$ into $n+1$ grid points p_{ij} where $i \in \{0, 1, \cdots, N\}$ and $j \in \{0, 1, \cdots, n\}$ where $p_{ij} = b_i + \frac{j(\bar{x} - b_i)}{n}$. Notice that a point $p_{i0} = b_i$ (on the boundary of B), and $p_{in} = \bar{x}$ by construction. Let a set $\mathcal{A}_i$ be a set of grid points $\{p_{ij} \in \operatorname{seg}(b_i; \bar{x})\}$ where $i \in \{0, 1, \cdots, N\}$ and $j \in \{0, 1, \cdots, n\}$. The sets $\mathcal{A}_0$ to $\mathcal{A}_N$ are illustrated by dashed line segments in Figure 4. The grid points p_{ij} where $i \in \{0, 1, \cdots, N\}$ and $j \in \{0, 1, \cdots, n\}$ belonging to each set $\mathcal{A}_i$ are marked by circles. Discretizing the search space as above allows converting the problem in

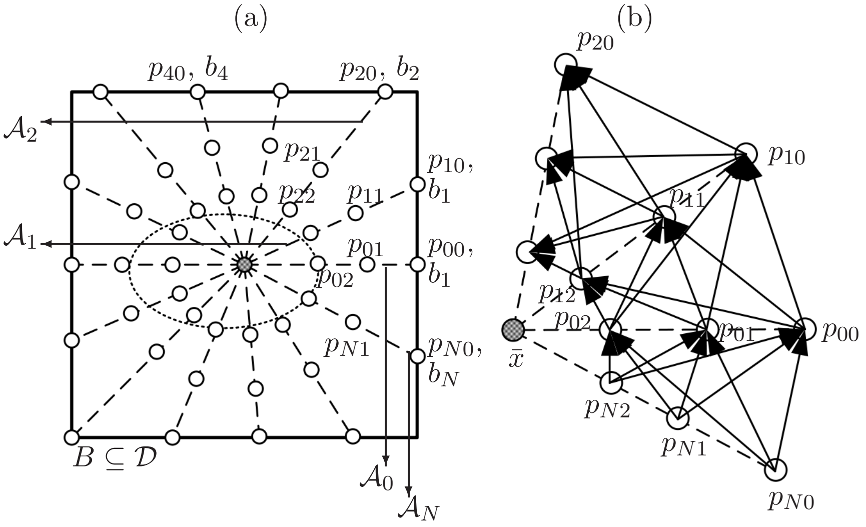


Figure 4: (a) Discretizing the search space to find the smallest RFIS (dotted ellipse) contained within a given invariant set B. The shaded circle is the equilibrium $\bar{x}$. (b) Embedding a directed graph structure.

Equation (7) into the following $N+1$ separate unconstrained optimization problems: for all $i \in \{0, 1, \cdots, N\}$, find an $x^{\star}$ satisfying

$$x^{\star} = \underset{x \in \mathcal{A}_i}{\arg\min}((\eta \circ \alpha \circ \mathscr{C}_{\vartheta})(x) + \lambda_1 \parallel x - \bar{x} \parallel_2), \lambda_1 > 0. \tag{8}$$

Definition 3.2.1. *A path is said to be the closest approximation of the boundary of the smallest RFIS if all nodes on this path are solutions to Equation* (8)*. Here a path, as defined in Section 3.1, is an ordered set of nodes with each node* $p_{(i+1)k}$ *a successor node of node* p_{ij} *where* $i \in \{0, 1, \cdots, N\}$ *and* $j, k \in \{0, 1, \cdots, n\}$*. Also* j, k *do not depend on each other or on* i*.*

3.2.3.2 A tie-breaker rule

For each $i \in \{0, 1, \cdots, N\}$, solutions to Equation (8) are grid points $p_{ij} \in \mathcal{A}_i$ where $i \in \{0, 1, \cdots, N\}$ and $j \in \{0, 1, \cdots, n\}$. For each $i \in \{0, 1, \cdots, N\}$ there may exist multiple different $j \in \{0, 1, \cdots, n\}$ such that each point p_{ij} is a solution to Equation (8). This situation is known as a tie, and if a tie occurs for any $i \in \{0, 1, \cdots, N\}$ then the point $p_{ij} \in \mathcal{A}_i$ with the smallest index $j \in \{0, 1, \cdots, n\}$ is selected as a solution to Equation (8). Such a tie-breaker rule guarantees a unique solution to Equation (8) for all $i \in \{0, 1, \cdots, N\}$. This claim of uniqueness is easily verified as follows. By construction, given an $i \in \{0, 1, \cdots, N\}$ there exists exactly one unique point p_{ij} corresponding to each $j \in \{0, 1, \cdots, n\}$. To see why, pick an $i \in \{0, 1, \cdots, N\}$ and suppose that there exist points $p_{ij} \in \mathcal{A}_i$, and $p_{ij'} \in \mathcal{A}_i$ such that $p_{ij} = p_{ij'}$ but $j \neq j'$, where $j, j' \in \{0, 1, \cdots, n\}$. By construction, $p_{ij} = p_{ij'}$ implies $b_i + \frac{j(\bar{x} - b_i)}{n} = b_i + \frac{j'(\bar{x} - b_i)}{n}$, which yields $j = j'$ on simplification. Therefore, in case of a tie, picking the smallest j from the finite set $\{0, 1, \cdots, n\}$ produces a unique solution $p_{ij} \in \mathcal{A}_i$ to Equation (8), for each $i \in \{0, 1, \cdots, N\}$. The set of problems given by Equation (8) can then be solved using path planning algorithms.

3.2.3.3 Embedding a directed graph structure

To use path planning algorithms for producing positively oriented simple closed curves around the equilibrium point as solutions to Equation (8), the following graph structure (shown in Figure 4(b)) is proposed. Excluding the equilibrium point $\bar{x}$, connect every grid point in set $\mathcal{A}_i$ to every grid point in set $\mathcal{A}_{i+1}$. We do not connect any grid point in a set $\mathcal{A}_i$ to any other grid point in the same set, or to any grid point in set $\mathcal{A}_{i-1}$, i.e. paths cannot go backward (or clockwise around the equilibrium $\bar{x}$). To produce closed paths, the index i is made to wrap around i.e. $i+1=0$ if $i=N$, and $i-1=N$ if $i=0$. This connects grid points (excluding the equilibrium) in set $\mathcal{A}_N$ to grid points in set $\mathcal{A}_0$. An illustration of this graph structure, which is named G' for convenience, is seen in Figure 4(b). Paths exist from each of the grid points $p_{00}, p_{01}, p_{02} \in \mathcal{A}_0$ directed to each of the grid points $p_{10}, p_{11}, p_{12} \in \mathcal{A}_1$. Similarly, given a grid point $p_{N2} \in \mathcal{A}_N$, directed paths exist to grid points $p_{00}, p_{01}, p_{02} \in \mathcal{A}_0$.

The following reasons motivate embedding the directed graph structure described above. By construction, if a set of paths is chosen from among the paths belonging to the graph G' shown in Figure 4(b) to form a directed simple closed curve, then the set formed by the region enclosed by such a curve contains the equilibrium point $\bar{x}$. This is due to the fact that every grid point p_{ij} belongs to a set $\mathcal{A}_i$.

To form the required directed simple closed curve, a path planning algorithm can aid in picking the right subset of paths from paths belonging to the graph G'. To plan an optimal path which provides an approximation of the boundary of the smallest RFIS, the graph G' is used with the A^* path planning algorithm from [27].

3.3 Leveraging the A^* path planning algorithm

3.3.1 Setting up sources, goals, and cost functions

Path planning algorithms generally require a source node, a goal set and an evaluation function used to estimate the cost of a path planned. Finding the smallest RFIS requires the set of source nodes and goal nodes to be the same, i.e. set $\mathcal{A}_0$. The A^* algorithm [27] cannot start if the source node belongs to the goal set. To separate the source nodes from the goal set, a path which provides a boundary of the smallest RFIS must be planned in at least two halves. The first half originating at a source node belonging to the set $\mathcal{A}_0$ and terminating in the set $\mathcal{A}_{N'}$, where $N' = \lfloor {}^{N}\!/_{2} \rfloor$. The second half originates at the node where the first half of the path terminated in the set $\mathcal{A}_{N'}$, and the goal set is the set $\mathcal{A}_0$.

Let $f(p_{ij})$ be the actual cost of an optimal path constrained to go through node p_{ij} from a given source node to a preferred goal node. The cost $f(p_{ij})$ can be expressed as the sum $f(p_{ij}) = g(p_{ij}) + h(p_{ij})$, where $g(p_{ij})$ is the actual cost of an optimal path from a source node to the node p_{ij}, and $h(p_{ij})$ is the actual cost of an optimal path from p_{ij} to a preferred goal node. An evaluation function $\widehat{f}(p_{ij})$ is required so the proposed path planning algorithm eventually makes the estimated cost $\widehat{f}(p_{ij})$ converge to the optimal cost $f(p_{ij})$. To construct the cost functions, define

$$\mathbf{N}(p_{ij}) = \Gamma_{\pi/2}(p_{ij} - p_{(i-1)k}), \tag{9}$$

for a given point $p_{(i-1)k}$. Vector $\mathbf{N}(\cdot)$ depends only on p_{ij} since $p_{(i-1)k}$ is a given point. Hence $p_{(i-1)k}$ does not appear on the left hand side of the definition in Equation (9). In Equation (9), and in the following Equations (11),(12), and (13) points p_{ij}, and $p_{(i-1)k}$ belong to the sets $\mathcal{A}_i$, and $\mathcal{A}_{i-1}$ respectively. The symbol $\Gamma_{\pi/2}$ in Equation (9) represents the standard rotation matrix in $\mathbb{R}^2$, i.e.

$$\Gamma_{\pi/2} = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) \\ \sin(\pi/2) & \cos(\pi/2) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{10}$$

The vector $\mathbf{N}(p_{ij})$ in Equation (9) is the vector $p_{ij} - p_{(i-1)k}$ rotated counterclockwise by 90°. Given a point $p_{(i-1)k}$, the cost $c(\cdot,\cdot)$ which measures the angle between the extreme ray of the pointed convex cone $\mathscr{C}_{\vartheta}(p_{ij})$ and vector $\mathbf{N}(p_{ij})$, and the distance between point p_{ij} and the equilibrium $\bar{x}$ is defined as,

$$c(p_{(i-1)k}, p_{ij}) = (\eta \circ \alpha \circ \mathscr{C}_{\vartheta})(p_{ij}) + \lambda_1 \parallel p_{ij} - \bar{x} \parallel_2 . \tag{11}$$

The following evaluation functions will be used for A^*.

$$\widehat{f}(p_{ij}) = \widehat{g}(p_{ij}) + \widehat{h}(p_{ij}) \tag{12}$$

$$\widehat{g}(p_{ij}) = \min_{0 \leq k \leq n-1} (c(p_{(i-1)k}, p_{ij}) + \widehat{g}(p_{(i-1)k})), \tag{13}$$

where $j, k \in \{0, 1, \cdots, n-1\}$, $i \in \{1, \cdots, N\}$, and the initial cost is zero, i.e. $\widehat{g}(p_{0k}) = 0$ for all $k \in \{0, 1, \ldots, n-1\}$. Note that the term *initial cost* signifies the cost at the start of a new path being planned beginning from a source node belonging to the set $\mathcal{A}_0$. The constant λ_1 in Equation (11) is non-negative and is already chosen during formulation in Equation (7). Equations (9)-(13) also work for the case of going from set $\mathcal{A}_N$ to set $\mathcal{A}_0$. For the case of going from set $\mathcal{A}_N$ to set $\mathcal{A}_0$ the index $i = 0$ and $i - 1 = N$, since the index i wraps around. For this case it is important to note that substituting $i = 0$ in Equation (13) produces

$$\widehat{g}(p_{0j}) = \min_{0 \leq k \leq n-1} (c(p_{Nk}, p_{0j}) + \widehat{g}(p_{Nk})). \tag{14}$$

For the case of going from set $\mathcal{A}_N$ to set $\mathcal{A}_0$ this cost $\widehat{g}(p_{0j})$ must not be confused for the zero initial cost, and must not be set equal to zero. The following facts from [27] about the function $\widehat{h}(\cdot)$ used in Equation (12), are important from the point of view of reducing computational effort. As long as $\widehat{h}(\cdot) \leq h(\cdot)$, the A^* algorithm produces an optimal path [27]. Here $\widehat{h}(\cdot)$ is the estimated cost to get to a goal node from a current node, and $h(\cdot)$ is the actual cost to get to a goal node from a current node. If $\widehat{h}(\cdot) > 0$ and $\widehat{h}(\cdot)$ is also a lower bound for $h(\cdot)$, then A^* provides an optimal path by searching fewer nodes compared to the case of using $\widehat{h}(\cdot) = 0$. For details see [27]. The functions $\widehat{h}(\cdot)$ and $h(\cdot)$ are known as heuristic costs. The function $\widehat{h}(\cdot)$ is usually chosen based on prior knowledge available to a system designer by way of experience, or from familiarity with a particular system. For instance, a system designer may have a heuristic estimate for a lower bound of the length of the boundary of an RFIS. Such an estimate is represented by $\widehat{h}(\cdot)$, and it may be based on the designer's knowledge of the properties of a particular vector field. The actual length of the boundary of the smallest RFIS may be thought of as the cost $h(\cdot)$. It is usually not possible to pick $h(\cdot)$, as it depends on the properties of a system.

Recall that the proposed algorithm requires planning a circular path in two halves. The first half originates at a source node belonging to the set $\mathcal{A}_0$ and terminates at a preferred goal node on the set $\mathcal{A}_{N'}$, where $N' = \lfloor {}^{N}\!/\!{}_{2} \rfloor$. The second half originates at the node where the first half of the path terminated in the set $\mathcal{A}_{N'}$, and the goal set for the second half is the set $\mathcal{A}_0$. Thus, the second half path begins at a node $p_{N'k'}$ belonging to set $\mathcal{A}_{N'}$. This node $p_{N'k'}$ is where the first half path planned by A^* terminates. For planning the second half of the required circular path beginning at the point $p_{N'k'}$ and going to the goal set $\mathcal{A}_0$, the initial cost $\widehat{g}(p_{N'k'})$ should not be set to zero, but set equal to the cost to get to point $p_{N'k'}$. This is because the point $p_{N'k'}$ is a pseudo-initial point. This point is required to be the initial point on the second half so a circular path can be planned. However, in terms of the entire circular path, this point does not represent any initial condition at all, therefore the cost to get to point $p_{N'k'}$ must be accounted for when planning the second half of the circular path beginning at a point $p_{N'k'}$.

3.3.2 The proposed algorithm for finding an approximation of the smallest RFIS

Algorithm 1 presents the pseudo code proposed to find an approximation of the smallest RFIS. The quantity r in Algorithm 1 is the iteration count with the initial value 1. The sequence P_0 is initialized to contain the points $\{b_0, b_1, \cdots, b_N\}$ belonging to the boundary of the initial guess B. Algorithm 1 generates the sequence P_r which iteratively approaches the desired boundary. The variable σ decides when Algorithm

Algorithm 1: Compute an approximation of a robust forward invariant set

```
   Data: Sets $\mathcal{A}_0$ to $\mathcal{A}_N$, graph $G'$, $\widehat{f}(\cdot)$, $p_{0j} \in \mathcal{A}_0$, $\epsilon, \widetilde{g} \in \mathbb{R}$, and $n, N \in \mathbb{N}$.
   Result: $P_r$
1  Let $r = 1, N' = \lfloor N/2 \rfloor, \sigma = 2\epsilon, P_0 = \{b_0, b_1, \cdots, b_N\}$;
2  Let $\widetilde{p}_{r,0} = p_{0j} \in \mathcal{A}_0$;
3  while $\sigma > \epsilon$ do
4      Use $A^*$ to find an optimal path from $\widetilde{p}_{r,0} \in \mathcal{A}_0$ to the goal set $\mathcal{A}_{N'}$;
5      Let $\widetilde{p}_{r,N'}$ represent the point picked by $A^*$ in line 4 from the set $\mathcal{A}_{N'}$;
6      Store the points obtained above as the sequence $P_r = \{\widetilde{p}_{r,0}, \in \mathcal{A}_0, \widetilde{p}_{r,1} \in \mathcal{A}_1, \cdots, \widetilde{p}_{r,N'} \in \mathcal{A}_{N'}\}$;
7      Use $A^*$ to find an optimal path from $\widetilde{p}_{r,N'} \in \mathcal{A}_{N'}$ to the goal set $\mathcal{A}_0$;
8      Let $\widetilde{p}_{r,N}$ represent the point picked by $A^*$ in line 7 from the set $\mathcal{A}_N$;
9      Let $\widetilde{p}_{r+1,0}$ represent the point picked by $A^*$ in line 7 from the set $\mathcal{A}_0$;
10     Update sequence $P_r$ as $P_r = \{\widetilde{p}_{r,0} \in \mathcal{A}_0, \cdots, \widetilde{p}_{r,N'} \in \mathcal{A}_{N'}, \widetilde{p}_{r,N'+1} \in \mathcal{A}_{N'+1}, \cdots, \widetilde{p}_{r,N} \in \mathcal{A}_N\}$;
11     Compute $\sigma_r = d_2(P_r, P_{r-1})$;
12     Let $\sigma = \sigma_r$, $\widetilde{p}_{r,0} = \widetilde{p}_{r+1,0} \in \mathcal{A}_0$;
13     $r = r + 1$;
14 return $P_r$;
```

1 terminates. It is initialized at a value greater than ϵ, where ϵ is a small non-zero positive constant chosen by a user. Let $\widetilde{p}_{r,0}$ represent some initial source node $p_{0j} \in \mathcal{A}_0$. During every iteration r, the first half of the resulting path is planned starting from the point $\widetilde{p}_{r,0}$ to the goal set $\mathcal{A}_{N'}$, where $N' = \lfloor N/2 \rfloor$. The points (nodes) picked by A^* from the sets $\mathcal{A}_0, \cdots, \mathcal{A}_{N'}$ are stored into the sequence P_r in the order $\{\widetilde{p}_{r,0}, \widetilde{p}_{r,1}, \widetilde{p}_{r,2}, \cdots, \widetilde{p}_{r,N'}\}$. Here a point $\widetilde{p}_{r,i}$ belongs to the set $\mathcal{A}_i, 0 \leq i \leq N'$. The second half of the resulting path is planned starting from the point $\widetilde{p}_{r,N'} \in \mathcal{A}_{N'}$ to the goal set $\mathcal{A}_0$. The sequence P_r is updated with the points picked by A^* from the sets $\mathcal{A}_{N'+1}, \cdots, \mathcal{A}_N$ as

$$P_r = \{\widetilde{p}_{r,0}, \widetilde{p}_{r,1}, \cdots, \widetilde{p}_{r,N'}, \widetilde{p}_{r,(N'+1)}, \cdots, \widetilde{p}_{r,(N-1)}, \widetilde{p}_{r,N}\}. \tag{15}$$

To start the next iteration, the point $\widetilde{p}_{r+1,0}$ is selected to be the point picked by A^* from the set $\mathcal{A}_0$ at the end of the previous iteration. The variable σ is then updated to be the value of the distance $d_2(P_r, P_{r-1})$. If σ is greater than the tolerance ϵ, the iterations continue. Otherwise the algorithm terminates. The hypothesis is that under certain conditions, the simple closed curve formed by the segments $\mathrm{seg}\,(\widetilde{p}_{r,0}; \widetilde{p}_{r,1})$, $\mathrm{seg}\,(\widetilde{p}_{r,1}; \widetilde{p}_{r,2})\,, \cdots,\ \mathrm{seg}\,(\widetilde{p}_{r,N-1}; \widetilde{p}_{r,N})\,, \mathrm{seg}\,(\widetilde{p}_{r,N}; \widetilde{p}_{r,0})$ is the closest approximation to the boundary of the smallest RFIS. A pictorial representation of the algorithmic process of picking feasible points $\widetilde{p}_{r,i}$ along a path being planned is shown in Figure 5.

3.3.3 An interpretation of the cost functions used

The definition of vector $\mathbf{N}(p_{ij})$ can be understood as follows. The goal is to construct a positively oriented simple closed curve, which forms the boundary of the smallest RFIS. The segment $\mathrm{seg}\left(p_{ij}; p_{(i-1)k}\right)$ joining points p_{ij} and $p_{(i-1)k}$ can be thought of as a small segment of some curve. This segment $\mathrm{seg}\left(p_{ij}; p_{(i-1)k}\right)$ is to be tested to find if it qualifies to be a part of the boundary of the required smallest RFIS. For such a test, the angle between the normal $\mathbf{N}(p_{ij})$ and vectors $\nu \in C_{\vartheta}(p_{ij})$ are required. This motivates the definition in Equation (9), and also the definitions in Equations (11),(12), and (13).

The choices made for the cost functions in section 3.3.1 may be interpreted as follows. Note that $\widehat{h}(p_{ij})$ is an estimate of the cost of an optimal path from node

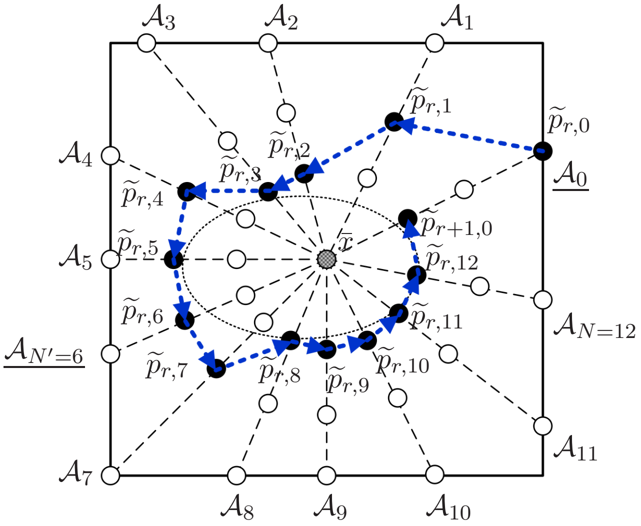


Figure 5: An illustration of planning the required path in two halves, first from $\mathcal{A}_0$ to $\mathcal{A}_{N'}$, and then from $\mathcal{A}_{N'}$ back to $\mathcal{A}_0$. Note that the set $\mathcal{A}_0$ is shown to not lie on a horizontal line segment because any set can be chosen to be the set $\mathcal{A}_0$ as desired.

p_{ij} to a preferred goal node. As per [27, Theorem 1], for A^* to produce an optimal path from a given source node to a preferred goal node, $\widehat{h}(\cdot)$ is required to be any lower bound of $h(\cdot)$, i.e. $\widehat{h}(p_{ij}) < h(p_{ij})$. This is achieved by setting $\widehat{h}(p_{ij}) = 0$ (which corresponds to the extreme case of the goal node being infinitesimally close to a current node), and assuming $h(p_{ij}) > 0$. From the definition of $c(\cdot, \cdot)$ in Equation (11) it is easy to see that the optimal cost to reach a preferred goal node from a given source node is positive as the function $\eta(\cdot)$ is positively valued, and the second term in the definition of $c(\cdot, \cdot)$ is positive. Therefore the assumption that $h(p_{ij}) > 0$, is reasonable.

Picking the function $\eta(\cdot)$ as in Equation (5) achieves the following two objectives. First, if $\widetilde{g}$ is picked to be sufficiently large, then points at which the constraint in Equation (4) is violated are never picked by the Algorithm 1. This is shown in Lemma 3.4.1. Second, the function definition $\eta(\cdot)$ ensures that the path cost $c(p_{(i-1)k}, p_{ij})$ is positive. This is required for using the A^* path planning algorithm.

3.4 Mathematical justification

The results in the following subsections justify that Algorithm 1 can be used to obtain an approximation of the smallest RFIS. A result is provided to establish conditions required to be satisfied by $\widetilde{g}$ so the constraint in Equation (4) is not violated. Further, a result is provided showing that if during an iteration r Algorithm 1 finds a grid point which lies on a unique optimal path, then Algorithm 1 terminates after two more iterations. The final result provides guidance on picking the right source node $p_{0j} \in \mathcal{A}_0$ so that Algorithm 1 provides an approximation of the boundary of the smallest RFIS on termination.

3.4.1 Picking $\widetilde{g}$ correctly

Lemma 3.4.1 specifies conditions that $\widetilde{g}$ must satisfy so the constraint in Equation (4) is not violated. The assumptions $i \in \{0, 1, \cdots, N\}$, $j \in \{0, 1, \cdots, n-1\}$, $p_{ij} \in \mathcal{A}_i$

and $h(p_{ij}) > 0$ for all i, j are made for all following results in Section 3.4.

Lemma 3.4.1. *Let a graph G', a point $p_{(i-1)k} \in \mathcal{A}_{i-1}$, and a non-zero positive constant λ_1 be given. Let the cost to reach node $p_{(i-1)k}$, which lies on a path from a given source node belonging to graph G' be $\widehat{g}(p_{(i-1)k})$, and $\widehat{g}(\cdot) \geq 0$. Suppose for all i there exists at least one j such that $\alpha \circ \mathscr{C}_\vartheta(p_{ij}) \in (0, 1]$, where $p_{ij} \in \mathcal{A}_i$. Let p_{ij^*} be a node chosen by A^*.* ***If*** *the constant $\widetilde{g} > 1 + \max_{i,j} \left(\lambda_1 \| p_{ij} - \bar{x} \|_2\right)$,* ***then*** *$\alpha \circ \mathscr{C}_\vartheta(p_{ij^*}) \in (0, 1]$.*

Proof. Let L be the cost of picking a point p_{ij} such that $\alpha \circ \mathscr{C}_\vartheta(p_{ij}) \in (0, 1]$. By the definition of $\eta(\cdot)$ in Equation (5), and the cost $c(\cdot, \cdot)$ in Equation (11), $L \leq 1 + \max_{i,j} \left(\lambda_1 \| p_{ij} - \bar{x} \|_2\right)$. Since $i \in \{0, 1, \cdots, N\}$, and $j \in \{0, 1, \cdots, n-1\}$, there exist a finite number of points p_{ij}. Therefore the quantity $\max_{i,j} \left(\lambda_1 \| p_{ij} - \bar{x} \|_2\right)$ exists. If $\widetilde{g} > 1 + \max_{i,j} \left(\lambda_1 \| p_{ij} - \bar{x} \|_2\right)$, then $\widetilde{g} > L$. The A^* algorithm picks points p_{ij} having the least cost [27]. Since for all i there exists at least one j such that $\alpha \circ \mathscr{C}_\vartheta(p_{ij}) \in (0, 1]$ and $\widetilde{g}$ is larger than the cost L, it follows from the above discussion that for any point p_{ij^*} chosen by A^* the quantity $\alpha \circ \mathscr{C}_\vartheta(p_{ij^*}) \in (0, 1]$. □

Therefore, as long as $\widetilde{g}$ satisfies the conditions in Lemma 3.4.1, Algorithm 1 picks points which do not violate the constraints in Equation (4).

3.4.2 Algorithm initialization and termination.

The following result is concerning the termination of Algorithm 1. The following result holds for any estimated cost $\widehat{h}(\cdot)$ to reach a goal node from any current node, as long as $\widehat{h}(\cdot)$ is a lower bound for the actual cost $h(\cdot)$ to reach a goal node from any current node.

Claim 3.4.2. *Let the iteration count $r \geq 1$, and let there exist a unique simple closed path $\bar{f}$ from the set $\mathcal{A}_0$ to the set $\mathcal{A}_0$ formed by directed paths belonging to the graph G' such that $\widehat{f}(\cdot)$ is minimized.* ***If*** *there exists i such that $\widetilde{p}_{r,i} \in \bar{f}$ where $\widetilde{p}_{r,i} \in P_r$*

given in Equation (15)*, and the point* $\widetilde{p}_{r,i+1} \in P_r$ *is chosen by the* A^* *algorithm to go from* $\widetilde{p}_{r,i} \in P_r$ *on the set* $\mathcal{A}_i$ *to the set* $\mathcal{A}_{i+1}$ *as shown in Algorithm 1.* ***Then*** $d_2(P_{r+1}, P_{r+2}) < \epsilon$*, where* ϵ *is a positive constant chosen by the user of Algorithm 1.*

Proof. Let there exist i such that a point $\widetilde{p}_{r,i} \in P_r$ is on the unique optimal closed path $\bar{f}$. Suppose the point $\widetilde{p}_{r,i+1} \in P_r$ is chosen by the A^* algorithm to go from $\widetilde{p}_{r,i} \in P_r$ on the set $\mathcal{A}_i$ to the set $\mathcal{A}_{i+1}$ as shown in Algorithm 1. From [27], it is known that A^* selects a point $\widetilde{p}_{r,i+1} \in \mathcal{A}_{i+1}$ such that the estimated total cost $\widehat{f}(\cdot)$ to go from $\widetilde{p}_{r,i} \in \mathcal{A}_i$ to $\widetilde{p}_{r,i+1} \in \mathcal{A}_{i+1}$ is minimized. But there exists only a single optimal closed path $\bar{f}$. This implies that the point $\widetilde{p}_{r,i+1} \in \bar{f}$. Similarly, each subsequent point picked by the A^* algorithm will lie on the optimal closed path $\bar{f}$. Therefore, the points $\widetilde{p}_{r+1,i} \in P_{r+1}$ picked by A^* during iteration $r+1$ are identical to the points $\widetilde{p}_{r+2,i} \in P_{r+2}$ picked by A^* during iteration $r+2$. Thus $d_2(P_{r+1}, P_{r+2}) = 0$ which is less than any non-zero positive ϵ, therefore completing the proof. $\square$

The above result implies the following. If during iteration r the A^* algorithm arrives at a point on the optimal path as it searches for a least cost path, and if there is only one such optimal path, then the Algorithm 1 will quickly terminate after completing the next two iterations, i.e. Algorithm 1 terminates at the end of iteration number $r+2$. Using the above result, the following proposition provides the conditions under which Algorithm 1 terminates to provide an approximation of the smallest RFIS. The following result does not provide an explicit proof, it only provides evidence that Algorithm 1 may produce points which lie on the boundary of the closest approximation of the smallest RFIS. The following proposition can also be used to speed up computation by searching for the appropriate source node for starting Algorithm 1.

Proposition 3.4.3. *Suppose that there exists a unique simple closed path* $\bar{f}$ *from the set* $\mathcal{A}_0$ *to the set* $\mathcal{A}_0$ *formed by directed paths belonging to the graph* G' *such that* $\widehat{f}(\cdot)$

is minimized. Suppose $\widetilde{g}$ *satisfies Lemma 3.4.1. Assume that there exists at least one set* $\mathcal{A}_i$ *such that a unique point* $p_{ij} \in \mathcal{A}_i$ *also belongs to* $\bar{f}$*, and such a point* p_{ij} *is the source node for Algorithm 1. Then the points* $\widetilde{p}_{r,i} \in P_r$ *produced by Algorithm 1 lie on the boundary of the closest approximation of the smallest RFIS as defined in Definition 3.2.1.*

Proof. Since p_{ij} belongs to the unique optimal path $\bar{f}$, and it is the source node, applying Claim 3.4.2 allows us to say that Algorithm 1 terminates. Further since $\widetilde{g}$ satisfies Lemma 3.4.1, it implies that the constraint of the problem in Equation (4) is not violated by the points $\widetilde{p}_{r,i} \in P_r$. From the formulation in Equation (8)-Equation (12), these points provide an optimal solution to the problem of finding an approximation of the boundary of the smallest RFIS. However, from assumptions only a single optimal path exists, hence the points $\widetilde{p}_{r,i} \in P_r$ produced by Algorithm 1 lie on the closest approximation of the boundary of the smallest RFIS, where the closest approximation of the boundary of the smallest RFIS is as given by Definition 3.2.1. $\square$

If the conditions for the above proposition are satisfied, then it is worthwhile to invest computing time to find such a source node belonging to $\bar{f}$ along some set A_0 so that less time is spent finding the boundary of the closest approximation of the smallest RFIS. As noted earlier, the above Proposition 3.4.3 does not provide an explicit proof. It only provides evidence that Algorithm 1 may produce points which lie on the boundary of the closest approximation of the smallest RFIS. This is because the the domain of interest is discretized. Hence Algorithm 1 can only pick discrete points which may lie on the boundary of the closest approximation of the smallest RFIS and therefore Algorithm 1 cannot guarantee the behavior of system trajectories in the space between two points. Algorithm 1 solves Equation 8. It is not necessary that Equation 8 always has a unique solution for each $i \in \{0, 1, \cdots, N\}$. The tie-breaker rule in Section 3.2.3.2 helps us decide which solution to pick in case multiple solutions exist. Therefore, the size of the region enclosed by the boundary produced by using Algorithm 1 is not necessarily the true size of the smallest RFIS, but only an approximate estimate of the size of the smallest RFIS.

3.5 Simulations

3.5.1 Testing the RFIS computation algorithm

Algorithm 1 is tested with three different problems and the results are presented in this section. The first test is carried out on a relatively simpler version of the curve tracking problem as used in [50]. The second test of Algorithm 1 is conducted on a slightly modified version of the curve tracking problem as used in [44]. Effects of using $\lambda_1 \in (0, 1]$ are observed in this test. The third test is carried out on a simple problem where system trajectories go around the equilibrium in circles. This test is used to verify that Algorithm 1 produces valid shapes.

The constant ϵ, length of time $T_{\max}$, and the constant $\widetilde{g}$ are used in every test. The values used are $\epsilon = 0.1$, $T_{\max} = 6.29$, and $\widetilde{g}$ is assigned the large value 4.5036×10^{15} hence the conditions required for Lemma 3.4.1 are satisfied. Such a large value for $\widetilde{g}$ is not necessary and is only chosen for convenience. A value like 100 may also work for $\widetilde{g}$ if it satisfies Lemma 3.4.1. The unit used for time is seconds. The length of time $T_{\max}$ aids the generation a cone at a given point in the domain of interest. A cone is generated by considering vectors at the given point in the domain of interest, at discrete time instances belonging to the interval $[0, T_{\max}]$. This is done to be able to compute an approximation of an RFIS, and it is conjectured that system trajectories stay in the computed approximation of an RFIS for the interval $[0, T_{max}]$. For the exact definition of an RFIS which is invariant for a finite interval of time, see Section 3.1. Both versions of the curve tracking problem require the constants ρ_0, μ and a disturbance $\delta(t)$. The values chosen are $\rho_0 = 1$, and $\mu = 6.42$, and the disturbance used is $\delta(t) = \delta_0 \sin(t)$ with $\delta_0 = 0.15$. Note that these values agree with those used in [44]. The details of each individual simulation run are provided further. Consider the following,

$$\dot{\rho} = -\sin(\phi) \tag{16}$$

$$\dot{\phi} = (\rho - \rho_0)\cos(\phi) - \mu\sin(\phi) + \delta(t). \tag{17}$$

The first test of Algorithm 1 is on the version of the curve-tracking problem used in [50], and is presented using Equations (16) and (17). Figure 6 shows the results obtained. Here ρ represents the distance of a robot from the closest point on a curve that the robot is trying to track, and ϕ is the difference in orientation between the robots heading direction and the tangent to the curve at the point closest to the robot. For this problem the values chosen for N, n, λ_1, and Δt are $56, 100, 1$, and 0.01 respectively. The square with blue circular markers is the initial guess B. Algorithm 1 terminates in four iterations and 11.15 minutes producing the sequence P_4 shown by the curve with red square markers. This is an approximation of the smallest RFIS for the system given by Equations (16) and (17), as per Algorithm 1. The black dotted curves (and the arrows) represent a few sample trajectories. The trajectories move

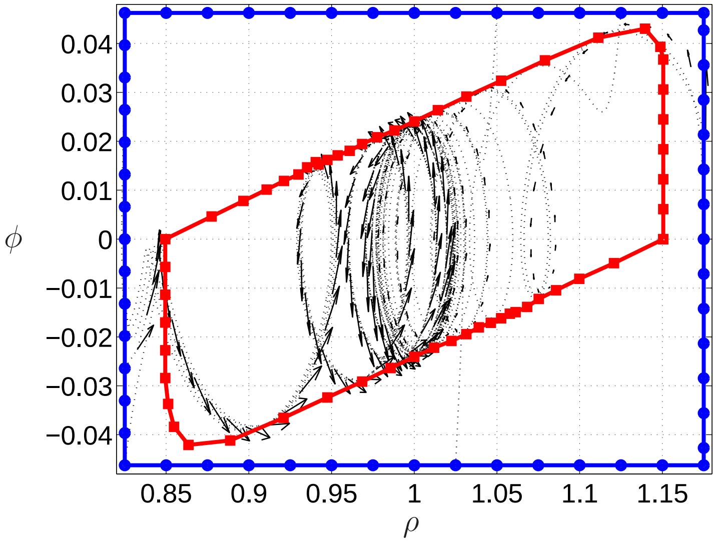


Figure 6: Results of using Algorithm 1 with the curve tracking problem given by Equations (16) and (17).

from the boundary of B to the inside of the region enclosed by points in P_4.

The second test was conducted on the following version of the curve tracking problem as used in [44]. The results obtained are shown in Figure 7.

$$\dot{\rho} = -\sin(\phi) \tag{18}$$

$$\dot{\phi} = \left(1 - \frac{\rho_0^2}{\rho^2}\right)\cos(\phi) - \mu\sin(\phi) + \delta(t). \tag{19}$$

For this problem the values chosen for N, n, and Δt are $57, 100$, and 0.1. Two tests were then carried out using Algorithm 1, one with $\lambda_1 = 1$, the other with $\lambda_1 = 0.9$. For both tests, Algorithm 1 requires approximately 47 seconds and terminates in three iterations. The square with blue circular markers is the initial guess B. The curve with red square markers is the set obtained for $\lambda_1 = 1$, and the dashed purple curve is the set obtained for $\lambda_1 = 0.9$. Reducing λ_1 from 1 to 0.9 produces a slightly larger estimate. The black dotted curves (and the arrows) show a few sample trajectories.

The third test was carried out to test that Algorithm 1 produces valid shapes. The results of this test are shown in Figure 8. For this test, Algorithm 1 is tested

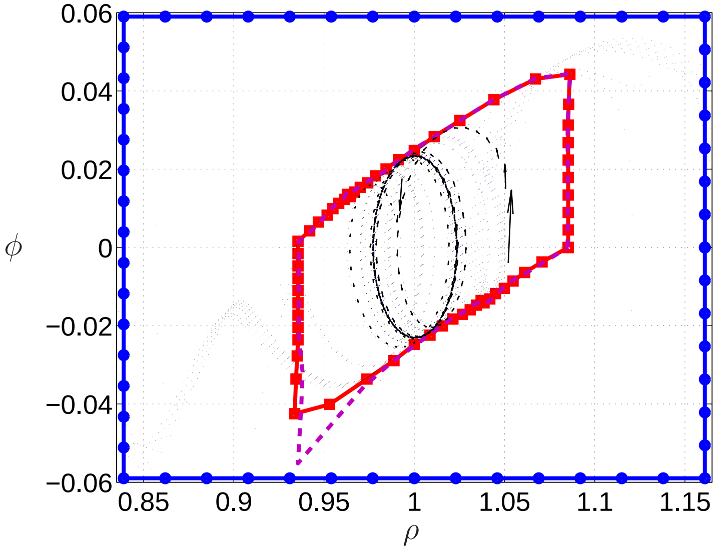


Figure 7: The effects of using $\lambda_1 \in (0, 1]$ in Algorithm 1 with the curve tracking problem given by Equations (18) and (19).

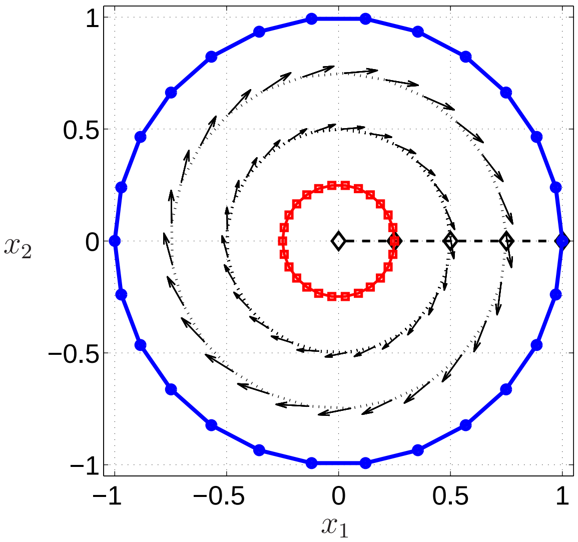


Figure 8: Using Algorithm 1 with the simple problem in Equation (20). The dashed black line shows set $\mathcal{A}_0$, the diamond shaped markers are the points p_{0j}.

with the following simple system.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \delta(t), \tag{20}$$

where $\delta(t) = [0 \quad 0]^T$. The values chosen for N, n, and λ_1 are $26, 4$ and 27. The curve with blue circular markers is the initial guess B. The curve with red square markers is the set produced by Algorithm 1 in four iterations and about 0.2 seconds. The result agrees with the system dynamics in Equation (20) as the trajectories of the system in Equation (20) are concentric circles. This verifies that Algorithm 1 produces the correct shape. The computation times reported for all above tests are obtained using a server grade desktop computer.

In summary, this chapter presented an algorithm and simulation results using the algorithm developed. From simulations, it is conjectured that the algorithm developed produces points which lie on the boundary of the closest approximation of the smallest RFIS for two dimensional systems subjected to bounded additive perturbations.

CHAPTER IV

DETECTING A LOSS OF SYSTEM STABILITY

Chapter 3 presented an algorithm for computing the smallest RFIS. However, the algorithm presented in Chapter 3 relies on measuring the vector field at distinct points, at discrete instants of time. Given a time varying nonlinear system, sampling a given vector field in a region of interest around an equilibrium point over a fixed interval of time may not be enough to guarantee that a computed invariant set is invariant for all time. This is because the stability of a time varying system can change abruptly. A given time-varying system may become unstable. This may cause a computed invariant set to not be invariant for all time. For example, a system may bifurcate, but a sophisticated model capturing such a phenomenon may not have been used for computing the smallest RFIS. In such a situation, detecting trend changes in the states of a given system may help decide if a system has become unstable and therefore if a computed RFIS is about to cease to be invariant. Detecting that a time-varying system is about to become unstable is the theme of this chapter.

As mentioned in Chapter 1, a method for monitoring a computed RFIS based on detecting trend changes in system states has computational advantages over monitoring points on the boundary of an RFIS. Prior work based on detecting Li-ion battery terminal voltage collapse, which does not need sophisticated models, can be useful for developing a strategy based on detecting trend changes in system states. This can further be used for monitoring a computed RFIS to detect that a given RFIS is about to cease to be invariant. This chapter focuses on the problem of Li-ion battery terminal voltage collapse. The material presented here is extended in Chapter 5 to algorithms for monitoring computed approximations of an RFIS.

4.1 Li-ion battery terminal voltage collapse detection; an inspiration

This chapter presents a general method for detecting Li-ion battery voltage collapses, without the requirement of a detailed model. Inspired by universal adaptive stabilization (UAS) [28, 49] a high-gain adaptive observer is developed, which is further used to detect changes in trend of the transient states of a Li-ion battery. The change in trend helps decide whether the terminal voltage of a battery is about to collapse. This novel approach, which only requires the measurement of the terminal voltage of a battery, works in the presence of measurement noise or voltage spikes due to non-smooth current discharges. It is not necessary to measure the discharge current. Thus, the cost of accurate current measurement and associated errors are eliminated. This method does not estimate the state of charge (SoC) of a battery and does not set a static threshold on the SoC or terminal voltage, so it is considerably robust to variations in the SoC and the terminal voltage. At first, this chapter introduces battery models and results concerning battery stability. A formulation of the voltage collapse detection problem, and the high-gain adaptive observer used along with a proof of convergence, are provided next. Further, a trend detection algorithm is introduced for detecting terminal voltage collapse. Simulations and experimental results form the end of this chapter, and show that the proposed approach is feasible.

4.1.1 Li-ion battery model and stability

Chen and Mora's (CM) model [14], shown in Figure 9, is an equivalent circuit representation of a Li-ion battery showing two coupled circuits. The left half models the variation of the SoC ρ (commonly known as the capacity remaining) and the right half models the variation of battery output voltage y as a function of the charge/discharge current $i(t)$. Knauff et.al. [36] derived the following state space realization for the

CM model

$$\dot{\rho}(t) = -\frac{1}{C_c} i(t) \tag{21}$$

$$\dot{x}_1(t) = -\frac{x_1}{R_{ts}(\rho) C_{ts}(\rho)} + \frac{i(t)}{C_{ts}(\rho)} \tag{22}$$

$$\dot{x}_2(t) = -\frac{x_2}{R_{tl}(\rho) C_{tl}(\rho)} + \frac{i(t)}{C_{tl}(\rho)} \tag{23}$$

$$y(t) = E_o(\rho) - x_1(t) - x_2(t) - i(t) R_s(\rho), \tag{24}$$

where y represents the voltage output from the battery, x_1 represents the voltage drop across $R_{ts}||C_{ts}$ and x_2 represents the voltage drop across $R_{tl}||C_{tl}$. The symbol $||$ represents a parallel combination of electrical components. The state $\rho \in [0, 1]$ represents the SoC. It has an initial value of 1. The states x_1, and $x_2 \in [0, \infty)$ have initial values set to 0. The circuit components $C_{ts}, C_{tl}, R_s, R_{ts}, R_{tl}$, and E_o are nonlinear functions of the SoC ρ given as follows:

$$C_{ts}(\rho) = -k_4 e^{-k_1 \rho} + k_3 \tag{25}$$

$$C_{tl}(\rho) = -k_6 e^{-k_2 \rho} + k_5 \tag{26}$$

$$R_s(\rho) = k_7 e^{-k_8 \rho} + k_9 \tag{27}$$

$$R_{ts}(\rho) = k_{10} e^{-k_{11} \rho} + k_{12} \tag{28}$$

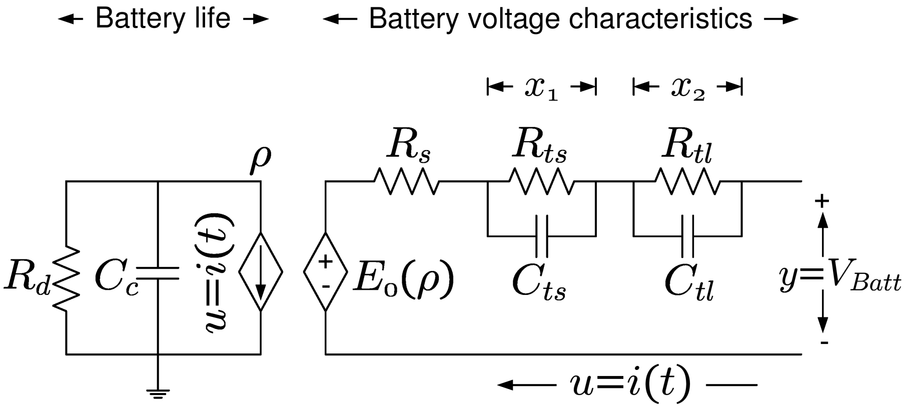


Figure 9: Chen and Mora's battery model

$$R_{tl}(\rho) = k_{13}e^{-k_{14}\rho} + k_{15} \tag{29}$$

$$E_o(\rho) = -k_{16}e^{-k_{17}\rho} + k_{18} + k_{19}\rho - k_{20}\rho^2 + k_{21}\rho^3 \tag{30}$$

$$C_c = 3600Cf_1f_2. \tag{31}$$

where $k_i > 0$ for $i = 1, 2, ..., 21$ are constants satisfying $0 < k_1 < k_2 < k_3 < k_4 < k_5 < k_6$. In Equation (31) $f_1, f_2 \in [0, 1]$ are factors accounting for the effects of temperature and charge-discharge cycles respectively. By default, $f_1 = f_2 = 1$, but their values will decrease after each charge-discharge cycle. C is the Ampere-hour capacity and E_o is the open-circuit voltage of a battery. The process of determining the constants k_1-k_{21} in Equations (25)-(30) takes considerable experimental effort as shown in [1, 14], and [18, 68], and the constants may be different for each battery.

The following facts regarding the stability of the CM model are known from [84]. There exist two constants δ_2 and δ_1 which are thresholds for the SoC ρ and satisfy $\delta_2 > \delta_1 > 0$. If the SoC satisfies $\delta_2 < \rho \leq 1$ then the subsystem in Equations (22) and (23) is asymptotically stable. If $\delta_1 \leq \rho \leq \delta_2$ then the subsystem is not asymptotically stable, and if $0 < \rho < \delta_1$, then the subsystem is unstable. Then from Equation (24) it is observed that the terminal voltage y drops very fast to 0, i.e. it collapses as the subsystem loses stability and x_1, and $x_2 \to \infty$ exponentially fast. The mathematical proofs for the above results related to Li-ion battery stability are provided below.

Considering the SoC ρ as a parameter and temporarily disregarding the input i, the system in Equations (21)-(24) can be rewritten as follows,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = A(\rho) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ where } A(\rho) = \begin{bmatrix} \dfrac{-1}{C_{ts}(\rho)R_{ts}(\rho)} & 0 \\ 0 & \dfrac{-1}{C_{tl}(\rho)R_{tl}(\rho)} \end{bmatrix}. \tag{32}$$

The above representation simplifies the nonlinear model of a battery to a parametrically varying linear system. In the proofs for battery stability provided further, the discharge current $i = 0$. If for a given value of the parameter $\rho \in [0, 1]$ the system in Equation (32) is asymptotically stable, then from results on linear system theory [13] it can be said that the system in Equation (32) is bounded-input bounded-output (BIBO) stable. For any battery, the discharge current i (which is the input in Equations (21)-(23)) is always bounded. Hence, even if the following stability results deal with the zero input case with $i = 0$, in the case with non-zero bounded input i the terminal voltage y of the battery remains bounded. This follows from the fact that if $A(\rho)$ has eigenvalues with negative real parts and the input i is bounded, then the states x_1 and x_2 of the linear parametrically varying battery system remain bounded, and the quantities $E_o(\rho)$, and $R_s(\rho)$ in Equation (24) are bounded by definition. Therefore, the simplification obtained using $i = 0$ is not necessarily very restrictive.

Consider Equation (32). The first stability result is based on the following candidate Lyapunov function and its time derivative:

$$\begin{aligned} V_1 &= \frac{1}{2}(x_1^2 + x_2^2) && (33) \\ \dot{V}_1 &= -\left(\frac{x_1^2}{R_{ts}(\rho)C_{ts}(\rho)} + \frac{x_2^2}{R_{tl}(\rho)C_{tl}(\rho)}\right). && (34) \end{aligned}$$

Lemma 4.1.1. *Consider* $C_{ts}, C_{tl}, R_{ts}, R_{tl}, V_1,$ *and* $\dot{V}_1$ *as given in Equations* (25)-(29), (33), *and* (34) *respectively. Assuming that* $\frac{1}{k_1}\ln\left(\frac{k_3}{k_4}\right) > \frac{1}{k_2}\ln\left(\frac{k_5}{k_6}\right)$, *for all SoC* $\rho \in [0, 1]$, *there exist small positive numbers* $\{(\delta_1, \delta_2) | 0 < \delta_1 < \delta_2\}$ *such that* $\dot{V}_1 > 0$ *for* $\rho \in (0, \delta_1)$ *and* $\dot{V}_1 \leq 0$ *for* $\rho \in (\delta_2, 1]$.

Proof. Observe that $V_1 > 0$, for all x_1, and $x_2 \neq 0$. Since $R_{ts}(\rho)$, and $R_{tl}(\rho)$ have the form $ae^{-b\rho} + c$, where $a, b, c > 0$, then $R_{ts}(\rho)$, and $R_{tl}(\rho) > 0$ for all $\rho \in [0, 1]$. Consider the case when $C_{ts}(\rho) < 0$. Solving Equation (25) for ρ gives,

$$\rho < -\frac{1}{k_1}\ln\left(\frac{k_3}{k_4}\right). \tag{35}$$

Similarly, considering $C_{tl}(\rho) < 0$ and solving Equation (26) for ρ gives

$$\rho < -\frac{1}{k_2}\ln\left(\frac{k_5}{k_6}\right). \tag{36}$$

Define δ_1 and δ_2 as follows:

$$\begin{aligned} \delta_1 &= -\frac{1}{k_1}\ln\left(\frac{k_3}{k_4}\right), && (37) \\ \delta_2 &= -\frac{1}{k_2}\ln\left(\frac{k_5}{k_6}\right). && (38) \end{aligned}$$

Since $k_3 < k_4$ and $k_5 < k_6$, this implies δ_1, and $\delta_2 > 0$. By our assumptions we get $0 < \delta_1 < \delta_2$. Therefore, if $\rho < \delta_1$ then C_{ts}, and $C_{tl} < 0$. That makes $\dot{V_1}$ positive. Similarly if $\rho > \delta_2$ then $C_{ts}, C_{tl} > 0$ and $\dot{V_1}$ is negative. The existence of the required constants δ_1 and δ_2 has thus been proved. $\square$

From the above proof, it is observed that the battery is unstable (in the Lyapunov sense [32]) when $\rho \in (0, \delta_1)$. When $\rho \in (\delta_2, 1]$ the battery is asymptotically stable. The constant δ_1 thus provides the worst case limit for the SoC of a battery. If the SoC ρ falls below δ_1, then the Li-ion battery is unstable. This means if the SoC ρ falls below δ_1 and $i > 0$, then the states x_1, and x_2 of the Li-ion battery will tend to infinity very quickly. The open circuit voltage $E_0(\rho)$, and the resistance $R_s(\rho)$ are bounded and positive. The discharge current $i(t)$ is positive, and in any real-life situation $i(t)$ is bounded. From the above discussion, and from Equation (24) it can be seen that the output voltage $y(t)$ will soon drop below any specified positive bound if $\rho < \delta_1$, and $i > 0$. Therefore, if the SoC falls below δ_1, one must switch a battery out of service. Note that the representation in Equation (32) simply aids in establishing the stability limits, and is not used to explicitly replicate the dynamics. Hence it does not introduce any error. These limits are applicable even to the system in Equations (21)-(24). The following claim can be made based on the previous lemma.

Claim 4.1.2. *Let δ_2 be obtained from Lemma 4.1.1, and suppose a value for the SoC ρ is given. If $\rho < \delta_2$, then the Li-ion battery system represented by equation* (32) *is not asymptotically stable.*

Proof. Approach the proof by contradiction. Suppose $\rho < \delta_2$ but the Li-ion battery system represented by equation (32) is asymptotically stable. Note that $R_{ts}(\rho), R_{tl}(\rho)$ have the form $ae^{-b\rho} + c$, where $a, b, c > 0$. This means that $R_{ts}(\rho), R_{tl}(\rho) > 0$ for all $\rho \in [0, 1]$. For a given ρ, both $C_{ts}(\rho)$ and $C_{tl}(\rho)$ must be greater than zero for the system in Equation (32) to be stable. Let $C_{tl}(\rho) > 0$. By the definition of $C_{tl}(\rho)$, this means

$$-k_6 e^{-k_2\rho} > -k_5. \tag{39}$$

Now since $\rho < \delta_2$, Equations (36) and (38) imply the following,

$$\rho < -\frac{1}{k_2} \ln\left(\frac{k_5}{k_6}\right) \tag{40}$$

$$-k_2\rho > \ln\left(\frac{k_5}{k_6}\right) \tag{41}$$

$$-k_6 e^{-k_2\rho} < -k_5. \tag{42}$$

Equation (42) contradicts Equation (39). By this contradiction, the system in Equation (32) is not asymptotically stable. □

In Equation (21), $C_c > 0$ and the discharge current $i(t) \geq 0$ for all time. From Equation (21) it can therefore be concluded that if the discharge current $i(t) = 0$ then the SoC ρ does not change, and for non-zero values of the discharge current $i(t)$ the SoC ρ decreases towards zero. Therefore, given an initial value for $\rho < \delta_2$, the SoC ρ always has a value less than δ_2. Intuitively, the above discussion and Lemma 4.1.1 indicate that removing a battery from service when $\rho < \delta_2$ is safer than removing the battery from service at a later instant of time when $\rho < \delta_1$. This is because if $\rho < \delta_2$, then the parameters of a Li-ion battery system are about to progress to a region where an impending terminal voltage collapse is expected.

4.2 A method for detecting Li-ion battery terminal voltage collapse

As a battery is gradually discharged, the value of its SoC ρ will decrease until the terminal voltage collapses. The goal is to detect when a Li-ion battery makes a transition from its stable region of operation (i.e. $\rho \in (\delta_2, 1]$) to the unstable region (i.e. $\rho \in (0, \delta_1)$) based on the measurements of its terminal voltage y. On the other hand, a temporary drop in terminal voltage does not imply that a battery is unstable. Such temporary drops in terminal voltage make it difficult to detect when a battery is about to die (i.e., that the terminal voltage is about to collapse soon). Any algorithm for detecting battery voltage collapse needs to avoid false alarms caused by temporary voltage drops, yet still be sensitive enough to detect when ρ moves out of the stability region.

An observer (or filter) can be designed to estimate the SoC ρ, and then one can use the thresholds δ_1 and δ_2 to determine whether the battery is stable or not. This approach has been taken by many in the literature [41, 61]. The challenge with this approach is that it requires the measurement of the input current i. In addition, it relies on the use of an accurate model such as the CM model for the observer, which may not always be available for all applications. The approach used in this work does not require measuring the discharge current i. Therefore, an estimate of the SoC is not produced. Instead, the states x_1 and x_2 of a Li-ion battery are estimated, and battery voltage collapse is detected based on the trend of these two states. It will also be shown that this approach does not require an accurate battery model for the observer to work satisfactorily.

4.2.1 The high-gain adaptive observer

Figure 10 shows the high-gain adaptive observer proposed for detecting the terminal voltage collapse of a Li-ion battery. It consists of three main blocks. The first block

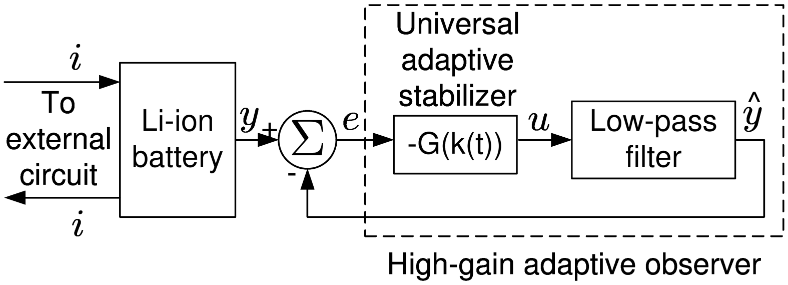


Figure 10: Battery output voltage tracking using universal adaptive stabilization (UAS).

on the left with input i and output y represents the real Li-ion battery, which is assumed to be represented by a CM model that is *unknown* to the observer. The block shown using a dashed rectangle is a high gain observer and consists of a low pass filtering block which is described in Section 4.2.1.1. The block in the middle represents the adaptive high gain, which is described in Section 4.2.1.2. Convergence analysis is provided in Section 4.2.1.3.

4.2.1.1 Lowpass filtering

The lowpass filtering block is a second order system with states $\widehat{x}_1, \widehat{x}_2$, input u, and output $\widehat{y}$ as follows

$$\dot{\widehat{x}}_1(t) = -c\widehat{x}_1(t) + u(t), \tag{43}$$

$$\dot{\widehat{x}}_2(t) = -c\widehat{x}_2(t) + u(t),\ c > 0 \tag{44}$$

$$\widehat{y}(t) = -\widehat{x}_1(t) - \widehat{x}_2(t), \tag{45}$$

where c is a positive constant chosen by the designer. The reason for using a second order system is to capture the dynamics of the states x_1 and x_2 of the CM model representing the Li-ion battery. Define $e(t) = y(t) - \widehat{y}(t)$ as the difference between the measured terminal voltage y and the output of the filter. Designing an observer will require a feedback law of the form $u(t) = -G(\cdot)e(t)$. The observer gain $G(\cdot)$ is designed to be either a simple increasing function of the error $e(t)$, or a particular function is chosen from a special class of functions called Nussbaum functions.

4.2.1.2 The adaptive gain

Nussbaum functions [54, 64, 78] are often used in the design of universal adaptive stabilizers (UAS) [28].

Definition 4.2.1. *Let $k' \in \mathbb{R}$. A piecewise right continuous function $N(\cdot) : [k', \infty) \to \mathbb{R}$ is called a Nussbaum function if it satisfies*

$$\begin{aligned} &\sup_{k>k_0} \frac{1}{k-k_0} \int_{k_0}^{k} N(\tau) d\tau = +\infty \qquad \textit{and} \\ &\inf_{k>k_0} \frac{1}{k-k_0} \int_{k_0}^{k} N(\tau) d\tau = -\infty \end{aligned} \tag{46}$$

for some $k_0 \in (k', \infty)$.

Consider the Mittag-Leffler function $E_\alpha : \mathbb{R} \to \mathbb{R}$ given by

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha + 1)}, \tag{47}$$

where α is a parameter, $z \in \mathbb{R}$, and $\Gamma(z+1) = z\Gamma(z), z > 0$ is the standard Gamma function. It is shown in [38], [39] that $E_\alpha(-\lambda t^\alpha), t \in \mathbb{R}$ is a Nussbaum function for $\lambda > 0$ and $\alpha \in (2, 3]$. Inspired by the UAS designs [28], the input $u(t)$ is selected as $u(t) = -G(k(t))e(t)$ where $G(\cdot)$ is the adaptive gain and $e(\cdot)$ is the error between the terminal voltage of a physical Li-ion battery and the output of the lowpass filter. One of the following choices are used for the adaptive gain function. Either a simple adaptive gain given by $G(k(t)) = k(t)$, or a Nussbaum type switching gain i.e. $G(k(t)) = N(k(t))$ where $N(\cdot)$ is of the form

$$N(k(t)) = E_\alpha\left(-\lambda k(t)^\alpha\right), \lambda > 0, \alpha \in (2, 3). \tag{48}$$

Thus, the adaptive high-gain observer shown in Figure 10 has the following nonlinear closed-loop dynamics

$$\left.\begin{aligned}\dot{\widehat{x}}_1(t) &= -c\widehat{x}_1(t) - G(k(t))e(t)\\ \dot{\widehat{x}}_2(t) &= -c\widehat{x}_2(t) - G(k(t))e(t)\\ \dot{k}(t) &= e^2(t)\\ e(t) &= y(t) + \widehat{x}_1(t) + \widehat{x}_2(t).\end{aligned}\right\} \tag{49}$$

The above design is structurally a high-gain observer [11, 29] with $G(k(t))$ as the adaptive high-gain. However, the goal of this work is different from the classical high-gain observer designs. The states of the observer $\widehat{x}_1$, and $\widehat{x}_2$ are not necessarily close to the actual states of the physical Li-ion battery. It is further shown that inaccuracy in state estimates will not prevent detecting battery instability under certain technical assumptions. This means accurate battery models are not necessary for observer design. The error dynamics of the system shown in Figure 10 are analyzed next.

4.2.1.3 Convergence Analysis

Lemma 4.2.2. *Consider the closed-loop dynamics in Equation (49). Let* $c > \frac{1}{2}$ *be a real positive constant. Let* $d(t) = cy(t) + \dot{y}(t)$*, and let the initial condition* $k(t_0)$ *be an arbitrary real number where* $t_0 \geq 0$*. Assume that* $\left(\int_{t_0}^{\infty} |d(\tau)|^2 \, d\tau\right)^{\frac{1}{2}} < \infty$*. If either* ***(i)*** $G(k(t)) = k(t)$ *and* $e^2(t)$ *is uniformly continuous for all* $t \in [t_0, \infty)$ *or* ***(ii)*** $G(k(t)) = N(k(t))$*, then* $\lim_{t\to\infty} k(t) = k_\infty$ *and* $\lim_{t\to\infty} e(t) = 0$*, where* k_∞ *is a finite real constant. The constant* k_∞ *depends on the initial error* $e(t_0)$*.*

Proof. By (49),

$$\dot{e}(t) = \dot{y}(t) + \dot{\widehat{x}}_1(t) + \dot{\widehat{x}}_2(t) \tag{50}$$

$$\dot{e}(t) = \dot{y}(t) - c\widehat{x}_1(t) - c\widehat{x}_2(t) - 2G(k(t))e(t). \tag{51}$$

Add and subtract $cy(t)$ to the right hand side of Equation (51) to get

$$\dot{e}(t) = \dot{y}(t) + cy(t) - c(y(t) + \widehat{x}_1(t) + \widehat{x}_2(t)) - 2G(k(t))e(t). \tag{52}$$

Using the fact that $d(t) = cy(t) + \dot{y}(t)$, and $e(t) = y(t) - \widehat{y}(t)$, the following system in the error $e(t)$ and state $k(t)$ is obtained.

$$\left.\begin{aligned} \dot{e}(t) &= -\left(c + 2G(k(t))\right) e(t) + d(t) \\ \dot{k}(t) &= e^2(t) \end{aligned}\right\} \tag{53}$$

(i) Suppose the adaptive gain $G(k(t)) = k(t)$ is used, and $e^2(t)$ is uniformly continuous for all $t \in [t_0, \infty)$. Consider the function $V_1 = \frac{1}{2}e^2(t) + k^2(t)$ with its time derivative as $\dot{V}_1 = -ce^2(t) + d(t)e(t)$. Therefore,

$$\dot{V}_1 \leq -\left(c - \frac{1}{2}\right) e^2(t) + \frac{1}{2}d^2(t). \tag{54}$$

For notational convenience let $a = c - \frac{1}{2}$. Integrating on both sides and letting $t \to \infty$ gives

$$\lim_{t\to\infty} V_1(t) - V_1(t_0) \leq -a \int_{t_0}^{\infty} e^2(\tau)d\tau + \frac{1}{2}\int_{t_0}^{\infty} d^2(\tau)d\tau. \tag{55}$$

The initial disturbance $d(t_0)$ is bounded as it depends on initial the values for $y(t_0)$, and $\dot{y}(t_0)$. The initial error $e(t_0)$ is bounded as it depends on the choice of initial states for the high-gain observer, and $V_1(t_0)$ is bounded. Since $\int_{t_0}^{\infty} d^2(\tau)d\tau$ is bounded by assumption and the first term on the right hand side of Equation (55) is non-positive, it is concluded that $V_1(\cdot)$ is bounded i.e. $\lim_{t\to\infty} V_1(t) < +\infty$. Also $V_1(\cdot)$ is bounded below by 0. Therefore, the term $\int_{t_0}^{\infty} e^2(\tau)d\tau$ must be bounded. By assumption, $e^2(t)$ is uniformly continuous for all $t \in [t_0, \infty)$. This means that $e^2(t)$ has no spikes of arbitrarily large magnitude for all $t \in [t_0, \infty)$. Further, since $\int_{t_0}^{\infty} e^2(\tau)d\tau$ is bounded, it follows that $e(t) \to 0$ as $t \to \infty$. This is because if there existed spikes in $e^2(t)$, or if $e(t)$ converged to any value other than zero, integrating $e^2(t)$ for infinite time would not lead to a bounded quantity. Thus from Equation (53), the integral $\int_{t_0}^{\infty} \dot{k}(t)dt$ converges. Therefore, $k(t)$ tends to some constant k_∞ as $t \to \infty$. By Equation (53), $\dot{k}(t)$ depends on $e(t_0)$, therefore the constant k_∞ depends on the initial error $e(t_0)$.

(ii) When the Nussbaum gain is used, let $G(k(t)) = N(k(t))$. Let the function $V_2 = \frac{1}{2}e^2(t) + 2\int_{k(t_0)}^{k(t)} N(\tau)d\tau$. The time derivative of $V_2(\cdot)$ gives $\dot{V}_2 = -ce^2(t) + d(t)e(t)$.

Similar to the previous case let $a = c - \frac{1}{2}$, so that

$$\dot{V}_2 \leq -ae^2(t) + \frac{1}{2}d^2(t). \tag{56}$$

Integrating Equation (56) on both sides provides

$$V_2(t) - V_2(t_0) \leq -a \int_{t_0}^{t} e^2(\tau)d\tau + \frac{1}{2}\int_{t_0}^{t} d^2(\tau)d\tau. \tag{57}$$

From (53), $\int_{t_0}^{t} e^2(\tau)d\tau{=}k(t) - k(t_0)$. Dividing both sides by $k(t) - k(t_0)$, letting $k(t) - k(t_0) = b(t)$ for convenience and re-arranging the terms gives

$$\frac{1}{b(t)}V_2(t) \leq \frac{1}{b(t)}V_2(t_0) - a + \frac{1}{2b(t)}\int_{t_0}^{t} d^2(\tau)d\tau. \tag{58}$$

The following is obtained by substituting $V_2 = \frac{1}{2}e^2(t) + 2\int_{k(t_0)}^{k(t)} N(\tau)d\tau$ in Equation (58).

$$\frac{e^2(t)}{2b(t)} + \frac{2}{b(t)}\int_{k(t_0)}^{k(t)} N(\tau)d\tau \leq \frac{V_2(t_0)}{b(t)} - a + \frac{1}{2b(t)}\int_{t_0}^{t} d^2(\tau)d\tau. \tag{59}$$

Re-arranging terms in Equation (59) gives the following,

$$\frac{e^2(t)}{2b(t)} \leq \frac{V_2(t_0)}{b(t)} - a + \frac{1}{2b(t)}\int_{t_0}^{t} d^2(\tau)d\tau - \frac{2}{b(t)}\int_{k(t_0)}^{k(t)} N(\tau)d\tau. \tag{60}$$

Similar to the proof in the first case $V_2(t_0)$ is bounded. The quantity $k(t_0)$ is known and bounded and $\int_{t_0}^{\infty} d^2(\tau)d\tau$ is bounded by assumption. The L.H.S. in (60) is non-negative for all time $t > t_0$. Suppose $k(t) \rightarrow \infty$ as $t \rightarrow \infty$. Then $b(t) = k(t) - k(t_0) \rightarrow \infty$ as $t \rightarrow \infty$, and as a result the following inequality holds as $k(t) \rightarrow \infty$:

$$\frac{e^2(t)}{4b(t)} \leq -\frac{a}{2} - \frac{1}{b(t)}\int_{k(t_0)}^{k(t)} N(\tau)d\tau. \tag{61}$$

But if $k(t) \rightarrow \infty$ as $t \rightarrow \infty$ then by definition of a Nussbaum function the term $\frac{1}{b(t)}\int_{k(t_0)}^{k(t)} N(\tau)d\tau$ can assume values that approach $+\infty$. This violates the fact that $\frac{e^2(t)}{4b(t)} \geq 0$. Therefore $k(t)$ must be bounded. Also from Equation (53) it is known that $k(\cdot)$ is a non-decreasing function. Hence, there exists some constant k_∞ such that $k(t) \rightarrow k_\infty$ as $t \rightarrow \infty$. This implies that $\dot{k}(t) \rightarrow 0$ as $t \rightarrow \infty$. Thus from Equation (53), $e(t) \rightarrow 0$ as $t \rightarrow \infty$. This completes the proof. □

Remark 4.2.3. *The assumption* $\left(\int_{t_0}^{\infty}|d(\tau)|^2\right)^{\frac{1}{2}} < \infty$*, is reasonable for detecting Li-ion battery voltage collapse. By definition, the disturbance* $d(t) = cy(t) + \dot{y}(t)$*. Here* $c > \frac{1}{2}$ *is a constant and* y *is the terminal voltage of a Li-ion battery. For any battery, the terminal voltage* y *has some initial value at time* $t_0 = 0$*. After some time* $T > t_0$ *the terminal voltage* $y(t) = 0$ *for all time* $t > T$*, because every battery can supply a load for a limited time after which a battery dies. Therefore the first term* $cy(t)$ *is bounded by physics and goes to zero after time* $t > T$*. Even though spikes in the terminal voltage may occur, in reality it is not possible to have a spike of infinite magnitude. Also after time* $t > T$ *we know that* $y(t) = 0$*. Therefore* $\dot{y}(t) = 0$ *for all* $t > T$*. Hence* $\dot{y}(t)$ *is bounded by physics and goes to zero after time* $t > T$ *as well. Hence the assumption is valid.*

4.2.2 An algorithm for detecting Li-ion battery terminal voltage collapse detection

Inspired by the Razumikhin theorem [86] and its use in [80], an algorithm to detect terminal voltage collapses for Li-ion batteries by monitoring trends in the states $\widehat{x}_1$ and $\widehat{x}_2$ is now proposed.

Algorithm 2: Algorithm For Battery Voltage Collapse Detection

```
Data: $j \in \{1,2\}, \widehat{x}_j, \sigma, r, \beta, \gamma, t, \epsilon, e$
Result: $S = \{0,1\}$
$\widehat{x}_{j_{min}} = \min\left(\widehat{x}_j(\tau)_{\tau\in[0,t]}\right)$;
$q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$;
$p = \frac{1}{q}$;
$p_{\max} = \max(p(\tau)_{\tau\in[t-r,t]})$;
if $\max\left(|e(\tau)|_{\tau\in[t-r,t]}\right) \leq \epsilon$ then
    if $(\dot{p}(t) < -\beta p(t))$ and $(\gamma p(t) \geq p_{\max})$ then
        $S = 1$;
    else
        $S = 0$;
return $S$;
```

In Algorithm 2, the user picks a value for j from the set $\{1, 2\}$. For monitoring both the states $\widehat{x}_1$ and $\widehat{x}_2$ two copies of the Algorithm 2 must be run in parallel, with $j = 1, 2$ respectively for each copy. Let $\widehat{x}_{j_{min}} = \min_{\tau \in [0,t]} (\widehat{x}_j(\tau))$ i.e. the running minimum for state $\widehat{x}_j$. Let $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$, σ be a positive constant, and $p = \frac{1}{q}$. Also $p_{\max}$ is the maximum value of p in a time window with length r prior to t. If the error guard condition $\max_{\tau \in [t-r,t]} (|e(\tau)|) \leq \epsilon$ is satisfied, then Algorithm 2 checks if $\dot{p}(t) < -\beta p(t)$ at time t. The discrete time approximation $\dot{p}(t) \approx \dfrac{(p(t+h) - p(t))}{h}$, with time step $h \neq 0$ is used to calculate the time derivative of p at time t. When the time derivative of p is less than $\beta p(t)$ at time t and $\gamma p(t) \geq p_{\max}$, where $\gamma > 1$, the indicator variable S is set equal to unity. The choices for the variables β, r, γ, and ϵ depend on a particular application and must be experimentally tuned. From experiments performed using real Li-ion batteries with Algorithm 2, values for $\beta \in [0, 1)$, larger window sizes (r), values for γ very close to one (i.e. say 1.001), and a choice of ϵ on the order of 0.001 have been observed to provide good performance in the presence of noise/discontinuities. From Lemma 4.2.2 the constant c for the low pass filter in Equations (43)-(45) must be chosen to be greater than 0.5. The following result proves that Algorithm 2 can be used to detect that the terminal voltage of a Li-ion battery is about to collapse.

Theorem 4.2.4. *Assume that the conditions required for Lemma 4.2.2 to hold are satisfied. Let* $j \in \{1, 2\}$, $\beta > 0$, $\gamma > 1$, *and* σ *be a small positive constant. For all* $j \in \{1, 2\}$ *let* $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$ *and* $p = \frac{1}{q}$. *Let* $\widehat{x}_{j_{min}} = \min\left(\widehat{x}_j(\tau)_{\tau \in [0,t]}\right)$, *and* $p_{max} = \max\left(p(\tau)_{\tau \in [t-r,t]}\right)$, $0 < r < t$. *Assuming* $\widehat{x}_{j_{min}}$ *is bounded for all* $j \in \{1, 2\}$, *and* $p(t)$ *is continuously differentiable for all* $t \in [0, \infty)$, *if* $\dot{p}(t) \leq -\beta p(t)$ *whenever* $\gamma p(t) \geq p_{\max}$ *then* $y(t) \to -\infty$ *as* $t \to \infty$.

Proof. By definition $\widehat{x}_j \geq \widehat{x}_{j_{min}}$ and σ is a positive number. Therefore $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$ is positive, and hence $p = \frac{1}{q}$ is also positive. Let $V(p(t)) = p(t)$ and $g(l) = \gamma l$, where $l \in \mathbb{R}$, and $\gamma > 1$. By definition $p_{max} = \max\left(p(\tau)_{\tau \in [t-r,t]}\right)$, $0 < r < t$.

Therefore, $0 < p(\tau) \leq p_{\max}$ for $\tau \in [t-r, t]$, where $0 < r < t$. Notice that $g(l)$ is non-decreasing, and from our assumptions $V(p(t))$ is continuously differentiable. Therefore the condition $V(\cdot) \leq g(V(\cdot))$ in the Razumikhin theorem [80, Theorem 5.1] holds if $p_{\max} \leq \gamma p(t)$. Let $w(p(t)) = \beta p(t)$ where $\beta > 0$. The function $w(\cdot)$ is positively valued for non-zero p. Further $\dot{V}(p(t)) = \dot{p}(t)$. Using the Razumikhin theorem [80, Theorem 5.1] we get that if $\dot{p}(t) \leq -\beta p(t)$ whenever $\gamma p(t) \geq p_{\max}$ then $p(t) \to 0$ as $t \to \infty$ for all $j \in \{1,2\}$. Since $q = \frac{1}{p}$, this gives $q \to \infty$ as $t \to \infty$ for all $j \in \{1,2\}$. However $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$, σ is a constant, and for all $j \in \{1,2\}$ $\widehat{x}_{j_{min}}$ is bounded by assumption. From the above discussion it follows that for all $j \in \{1,2\}$, $\widehat{x}_j \to \infty$ as $t \to \infty$. From the assumptions, Lemma 4.2.2 holds, i.e. as time tends to infinity, the error $e(t) = y(t) - \widehat{y}(t) \to 0$. Since $\widehat{y} = -\widehat{x}_1 - \widehat{x}_2$, and for all $j \in \{1,2\}$, $\widehat{x}_j \to \infty$ as $t \to \infty$, therefore, $y(t) \to -\infty$ as $t \to \infty$. This completes the proof. □

The above result suggests that Algorithm 2 can be used to detect that the terminal voltage of a physical battery is about to collapse by monitoring the states $\widehat{x}_1$ or $\widehat{x}_2$ of the simplified system. In any real life scenario the discharge current (even for a short circuit) is bounded, so it is not necessary to explicitly measure the discharge current $i(t)$ to verify that $i(t)$ is bounded while using Algorithm 2. It may appear that Algorithm 2 can be used directly with the terminal voltage to test for voltage collapse. Such use is prone to false alarms, because if Algorithm 2 detects a momentary terminal voltage drop (due to noise or disturbance) it does not always mean the battery is unstable. Also, using Algorithm 2 to monitor the states $\widehat{x}_1$ or $\widehat{x}_2$ of the low-pass filter provides an added layer of disturbance protection.

The condition $\max_{\tau \in [t-r,t]} (|e(\tau)|) \leq \epsilon$ is a guarding condition. It is used so that stability is judged when the output $\widehat{y}$ of the observer is sufficiently close to the terminal voltage y. This ensures that the algorithm is unaffected by transients. Algorithm 2 is a monitoring algorithm which stops running when a battery dies. The variable S in Algorithm 2 provides an indication of the estimated condition of a Li-ion battery at a particular time instant. If the indicator variable $S = 1$, then the user can decide to take appropriate actions.

4.3 Simulations and experimental results

To investigate the performance of the high-gain adaptive observer and the trend detection algorithm in real-life, the following simulations and experiments were performed.

4.3.1 Pitfalls of threshold based detection

Figure 11 shows the terminal voltage curve vs. capacity used, i.e. $(1-\rho)$, for a battery under different discharge currents. The horizontal black dashed line in Figure 11 represents a terminal voltage threshold of 3.5V. It correctly detects that the terminal voltage is about to collapse when the load current $i = 0.5A$, and $1A$. However, when $i = 2A$, using this threshold results in an incorrect detection as the battery still has a SoC = 50%. Also spikes may appear in the voltage due to sudden large current discharges causing an incorrect detection. Now consider preset thresholds on the capacity used. The vertical black dashed line in Figure 11 represents a threshold of

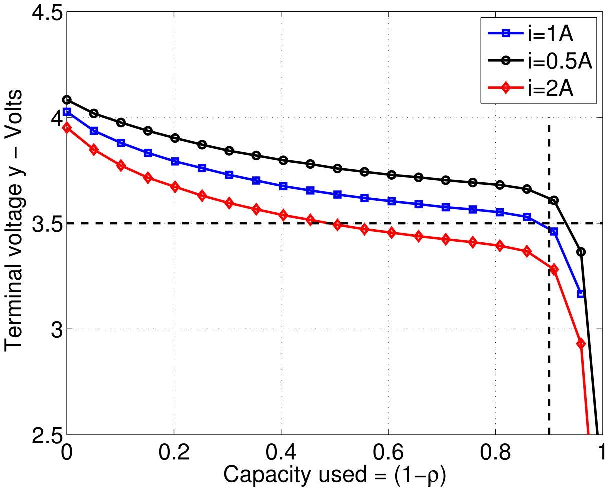


Figure 11: Terminal voltage vs. capacity used for a Li-ion battery supplying different discharge currents.

90%. For a load of $i = 0.5A$ the battery terminal voltage is above the previously used voltage threshold of 3.5V but using a threshold of 90% on the capacity used would remove the battery from service earlier than necessary.

4.3.2 Comparison of different classes of high gains

The upper half of Figure 12 shows a part of the terminal voltage curves obtained from a simulation run using the CM model in Equations (21)-(24) and the output $\widehat{y}$ of the high gain observer with different gains $G(k(t)) = k(t)$, and $G(k(t)) = N(k(t))$. The lower half of Figure 12 shows the tracking error in both cases. The simulation is carried out with a square wave discharge at 0.5 Hz, and 1A (peak to peak) with a DC offset of 3A. Battery capacity is assumed to be 100 mAh and the constant $c = 2$. Parameters for the CM model can be found in [14]. From Figure 12, the adaptive high-gain observer performs better with the Nussbaum type switching gain $G(k(t)) = N(k(t))$ as it tracks with lower error compared to the simple gain function $G(k(t)) = k(t)$. Thus the Nussbaum type switching gain is used further for simulations and experiments.

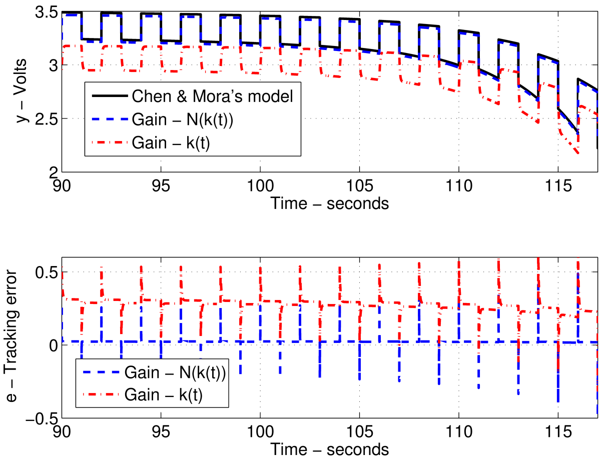


Figure 12: Comparison of Li-ion battery terminal voltage tracking performance with different adaptive high gains.

4.3.3 Voltage collapse detection

Figure 13 shows the results of using Algorithm 2, where the adaptive high-gain observer uses the gain $G(k(t)) = N(k(t))$ for detecting terminal voltage collapse. The values $2, 2.1, 200$(samples)$, 1\times10^{-3}$, and 2.9×10^{-3} are used for the variables c, γ, r, β, and ϵ respectively. The discharge current used is a square wave at 0.25 Hz in the presence of Gaussian noise with a covariance of 0.2. Algorithm 1 detects an impending voltage collapse at around $t = 98s$ when $S = 1$. At this time the terminal voltage is around 3.3V and is about to drop quickly and a sharp rise is seen in the state $\widehat{x}_1$.

Figure 14 is plotted to verify that Algorithm 2 detects that the terminal voltage of a Li-ion battery is about to collapse before the stability limit δ_2 is breached, i.e. before $\rho \leq \delta_2$. The upper half of Figure 14 shows the entire voltage curve, as opposed to the small portion of it shown in Figure 13. In both the upper and lower halves of Figure 14, the red solid vertical lines correspond to the instants at which $S = 1$. As in Figure 13, it is seen that $S = 1$ around $t = 98s, 110s$, and $120s$. The black dotted vertical line (superimposed on a red solid vertical line) represents the time instant $t \approx 119.08s$ at which $\rho(t) = \delta_2$. Note that this black dotted vertical line is

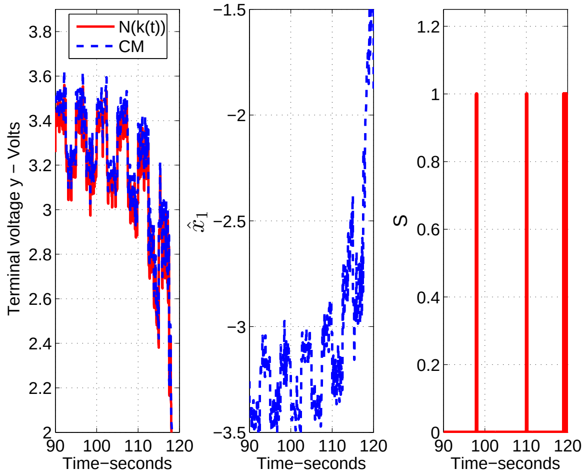


Figure 13: Falling y, $\widehat{y}$. Rising $\widehat{x}_1$ and the indicator variable S: square wave discharge with measurement noise, with $G(k(t)) = N(k(t))$.

coincident with the third vertical solid red line (another instant at which Algorithm 2 detects that the terminal voltage is about to collapse). The dashed black vertical line (superimposed on another red solid vertical line) corresponds to the time instant $t \approx 119.63s$ at which $\rho(t) = \delta_1$. The lower half of Figure 14 is a zoomed in version of the upper half so that the separation between the two distinct time instants at which $\rho(t)$ equals δ_2, and δ_1 is distinctly visible. Note that for the CM model [14], which is used here, $\delta_2 = 0.0112$ and $\delta_1 = 0.005$. From the simulation results it is found that at the time instant near $98s$ when Algorithm 2 detects that the terminal voltage is about to collapse, the value of $\rho = 0.1742$. Therefore, from the above discussion, and from Figures 13 and 14 it is verified that Algorithm 2 detects that the terminal voltage of Li-ion battery is about to collapse before a given battery becomes unstable.

4.3.4 Experiments

Figure 15 shows a schematic of the setup used to test Algorithm 2 on a Li-ion battery. Algorithm 2 runs on a computer running Matlab and Simulink. The battery tester board discharges the battery according to commands received from the computer. The Quanser Q2-USB AD/DA board interfaces the computer with the battery tester board. The tester board consists of a Li-ion battery 'B' connected in series with a relay

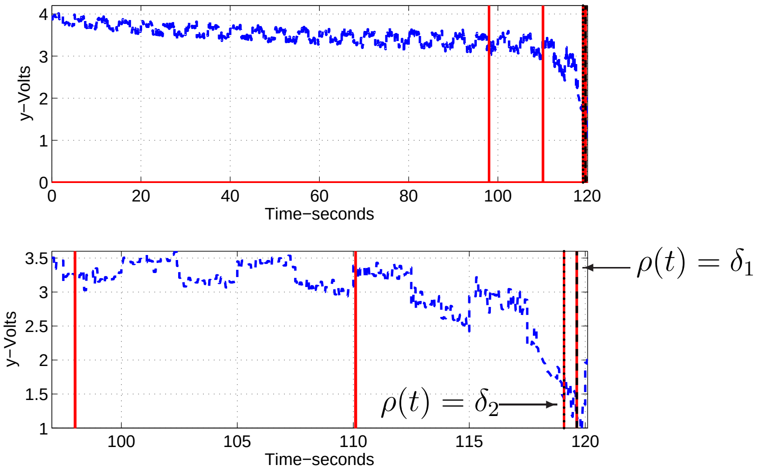


Figure 14: Verifying that Algorithm 2 detects terminal voltage collapse before $\rho \leq \delta_2$.

unit ‘R’, a light-emitting diode (LED) ‘L’, one of the resistors $R_1, \cdots, R_4$ depending upon the position of the selector switch ‘N’ and the current measurement unit ‘I’. Terminals V_1, and V_2 are used to measure the terminal voltage of the Li-ion battery. Terminals I_1, and I_2 are used to measure the current flowing through the circuit for diagnostic purposes only. The terminals T_1, and T_2 are the power and trigger signal terminals of the relay unit ‘R’. The physical circuit board is seen interfaced with the Quanser AD/DA board in Figure 16. For experiments a 3.0V RCR-123a rechargeable

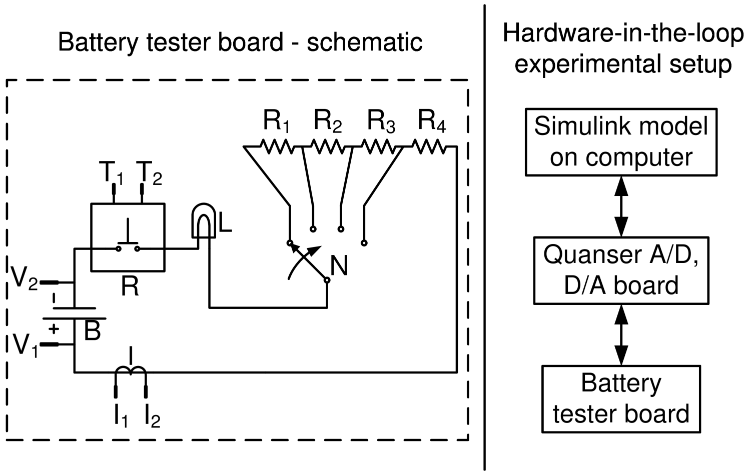

Battery tester board - schematic
R1 R2 R3 R4
T1 T2
L
N
R
V2
+
B
V1
I
I1 I2
Hardware-in-the-loop experimental setup
Simulink model on computer
Quanser A/D, D/A board
Battery tester board

Figure 15: Schematic of the experimental setup used to detect voltage collapse for a Li-ion battery using Algorithm 2.

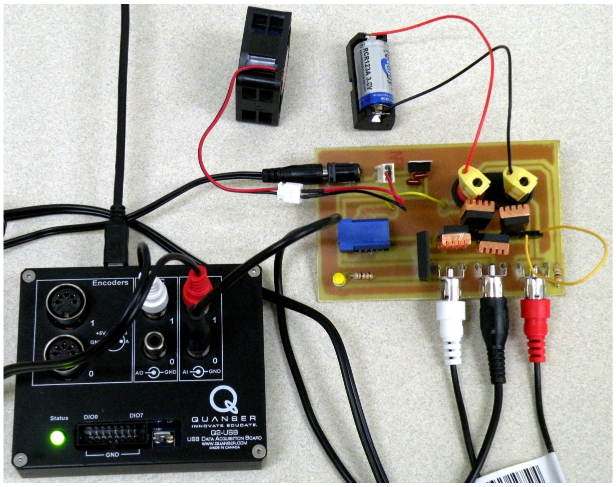


Figure 16: A simple prototype of the battery tester board connected to the Quanser Q2-USB board. (Thanks to Phillip Cheng for developing this board).

Li-ion battery is used. The relay unit is an IXYS CPC1709J. The resistors seen in Figure 16 are fitted with heat-sinks. The selector switch 'N' is formed by making the resistors individually removable. The current measurement unit is a LEM CAS-6NP current transducer. Terminal voltage measurements are obtained by connecting the battery directly to the AD/DA board. Current measurements are used to monitor proper operation of the battery tester board. An output channel on the AD/DA board is connected directly to the relay unit for providing trigger signals. The cooling fan seen in Figure 16 is an auxiliary unit for preventing the battery from heating up.

Depending on the value of the trigger signals, the relay is either open or closed. This causes one (or a combination) of the load resistors $R_1, \cdots, R_4$ selected using the selection switch to be connected to the battery 'B'. Varying the timing, and width of trigger pulses sent to the relay unit 'R' provides variable discharge frequencies and duty cycles. This allows Algorithm 2 to be tested in the presence of loads that switch on/off rapidly or very slowly. The LED 'L' acts as an indicator of proper circuit operation as it turns on when the load is connected to the battery, and off otherwise. Experimental results obtained using the above setup are presented next.

Figure 17 shows the results of one of several successful experimental runs. For

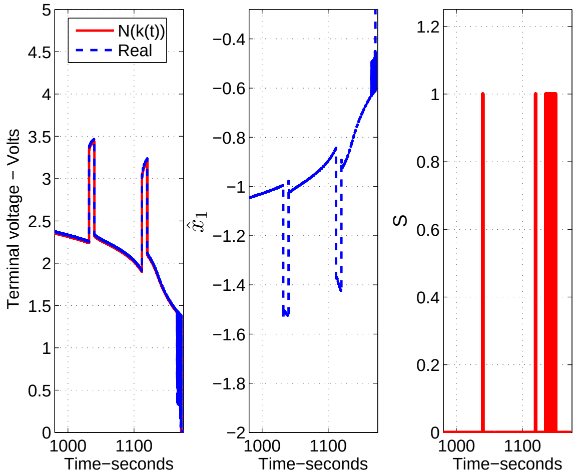


Figure 17: Results of using Algorithm 2 in a real experimental run.

this run a single 3V Li-ion battery supplies a $4\Omega, 1W$ resistive load and the discharge current is pulsed with a period of 80 seconds and duty ratio 0.9. For the duration that the load is being supplied, the terminal voltage is seen to drop (first panel of Figure 17) and then it rises back up again when the load is not drawing any current. The peak discharge current supplied was of the order of 1A as the initial terminal voltage of the battery used was 4V. The UAS strategy forces the high-gain observer to track the output voltage of a real Li-ion battery, as seen in the leftmost panel of Figure 17, where the output of the high-gain observer is plotted using a red solid line and the terminal voltage of the real Li-ion battery is plotted using a blue dashed line. The panel in the middle shows the state $\widehat{x}_1$ of the high-gain observer. As the terminal voltage of the battery begins collapsing the state $\widehat{x}_1$ begins increasing. The last panel of Figure 17 shows that Algorithm 2 detects that the terminal voltage of the battery is about to start declining sharply at $t = 1040s$ when the indicator variable $S = 1$. The values used for c, r, γ, β, and ϵ for this experiment are 2, 25000, 5, 1×10^{-3} and 0.01 respectively. The window size (r) is in terms of the number of samples and the sampling interval is 0.001 seconds.

By conducting various experimental runs, it has been confirmed that this method can be used to detect that the terminal voltage of a Li-ion battery is about to collapse regardless of the magnitude of the load current, the frequency, the duty ratio or the initial conditions of a battery. The algorithm can also be tuned to detect that the terminal voltage is about to drop slightly before the Li-ion battery becomes unstable, i.e. slightly before the terminal voltage starts declining sharply. Note that for the experimental run, the SoC ρ is never estimated as it is not required.

In conclusion, a high-gain adaptive observer and a trend filter have been used to detect changes in trend of the states of a physical Li-ion battery, which is used to detect Li-ion battery terminal voltage collapse. The method is based on classical techniques like UAS. The method presented has been theoretically justified, and also verified by simulations and experiments. This method only requires measurements of the terminal voltage, and does not require measurements of the discharge current. The method is less susceptible to false alarms which are a concern to static threshold based systems. Since the method does not require a detailed battery model, it may be robust to temperature variations, aging effects, changes in loading or other nonlinear disturbances. Since the key idea behind this method is to detect abrupt change in system behavior (which causes instability) even in the absence of a sophisticated model, it is conjectured that this method can be useful to detect that a computed RFIS is about to cease to be invariant, for systems which may bifurcate, and where a simple model fails to capture such a phenomenon.

CHAPTER V

ROBUST FORWARD INVARIANT SETS UNDER PARAMETER UNCERTAINTY

This chapter focuses on instability detection for parameter dependent systems. Recall that Chapter 3 focuses on a system of the form $\dot{x}(t) = \vartheta(x(t), \delta(t))$. In reality, there may exist parametric uncertainties in a system model. Such parametric uncertainties may arise out of a failure to obtain a highly accurate model for a system, or may be thought of as uncertainties arising out of discretization for computation purposes. Suppose an approximation of an invariant set is computed for a nominal system using the algorithm developed in Chapter 3. Further, suppose that this set is invariant over a given time interval. If parametric uncertainties are present in the system model, and an updated RFIS is not computed by correctly accounting for the effects of the parametric uncertainties, then a computed RFIS may not be invariant over the given time interval.

Suppose an RFIS is computed for a given system, and it is invariant for time $t \in [t_0, t_1]$ where $t_1 > t_0 \geq 0$. Changes in system dynamics may cause an RFIS, which is expected to be invariant over the interval $[t_0, t_1]$, to cease to be invariant. A computed RFIS may cease to be invariant within a given time interval $[t_0, t_1]$ due to system instability caused by parametric uncertainty. In this case it may not be possible to obtain a revised estimate for an RFIS. If the parametric uncertainty is bounded, and does not cause system instability, then the following case may occur. Suppose an RFIS is given which was expected to remain invariant over the interval $[t_0, t_1]$. However, it is not invariant due to the effects of parametric uncertainty, then results from perturbation theory may be used to compute a set in which the trajectories of the system with parametric uncertainty may remain for the interval $[t_0, t_1]$, if the initial conditions of the system with parametric uncertainty belong to the RFIS computed for the nominal system without parametric uncertainty. In this chapter it is shown that for a special case of a system satisfying certain conditions, if variations of the parametric uncertainty ε causes system instability, then the results developed in Chapter 4 can be used to detect such an instability. This can lead to the conclusion that a computed RFIS will soon cease to be invariant. The material presented in this chapter requires some of the notation and definitions presented in Chapter 3. The notations reused from Chapter 3 and a few new definitions are presented in the following section.

5.1 *Notation and definitions*

Consider the nominal system $\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0)$. Here $\vartheta : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$, and $\delta : \mathbb{R} \to \mathbb{R}^2$ is a time varying bounded disturbance such that standard conditions for existence and uniqueness of solutions are met. Let $\bar{x} \in \mathbb{R}^2$, and $\vartheta(\bar{x}, 0, 0) = 0$. Assume the dynamics can be written as $\vartheta(x(t), \delta(t), 0) = f(x(t)) + \delta(t)$, where $f : \mathbb{R}^2 \to \mathbb{R}^2$, and the disturbance function $\delta : [0, +\infty) \to \mathcal{U}$. The set $\mathcal{U}$ is defined as $[-\delta_0, \delta_0] \times [-\delta_0, \delta_0]$, for some known value $\delta_0 \in [0, +\infty)$. Let $\mathcal{D} \in \mathbb{R}^2$ represent the domain of interest. Let $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$ represent a parametrically perturbed version of the system given by $\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0)$, where $\vartheta : \mathcal{D} \times \mathcal{U} \times [-\varepsilon_0, \varepsilon_0] \to \mathbb{R}^2$ and time $t \in [t_0, t_1]$. Here ε_0 is a real positive constant and t_0, and t_1 represent time instants such that $0 \leq t_0 < t_1 \leq T_{\max}$, where $T_{\max}$ is a real positive constant. The constant T_{max} used here is the same as that in Chapter 3. The constant T_{max} is used because an RFIS computed for the system $\dot{x}(t) = \vartheta(x(t), \delta(t), 0)$ may not necessarily remain invariant over the entire interval $[0, T_{max}]$ in the presence of parametric uncertainties which cause the system dynamics to be given by $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$. Given $\varepsilon \in \mathbb{R}$ and $p \in \mathbb{R}^2$, let the ball $\mathcal{B}_\varepsilon(p) = \{x \in \mathbb{R}^2 : \| x - p \|_2 \leq |\varepsilon|\}$. Further, let $\delta_1(\varepsilon) \in \mathbb{R}$

and $\delta_2(\varepsilon) \in \mathbb{R}$ represent two quantities. Then as in [32], $\delta_1(\varepsilon)$ is said to be of the *order of* $\delta_2(\varepsilon)$, which is represented as $\delta_1(\varepsilon) = O(\delta_2(\varepsilon))$, if there exist positive constants k and c such that $|\delta_1(\varepsilon)| \leq k\,|\delta_2(\varepsilon)|$ for all $|\varepsilon| < c$.

5.2 Detecting a loss of system stability due to parameter uncertainty

Given a system of the form $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$, it is possible that a perturbation causes a fundamental behavioral change in a system. For example, a bifurcation may occur such that a parametric uncertainty makes the system unstable. In such situations it is important to identify that an RFIS, which was computed earlier and was invariant for a given interval of time, is about to cease to be invariant. This is because system trajectories may leave the computed RFIS within the given time interval. For a particular type of systems, the techniques developed for detecting Li-ion battery terminal voltage collapse, which were developed in Chapter 4, can be useful for detecting that a given system is approaching instability and therefore an RFIS computed earlier is about to cease to be invariant. The details of the high-gain adaptive observer shown in Figure 18, which is proposed to be used to detect that an RFIS is about to cease to be invariant, are similar to the one used in Chapter 4 and are presented further.

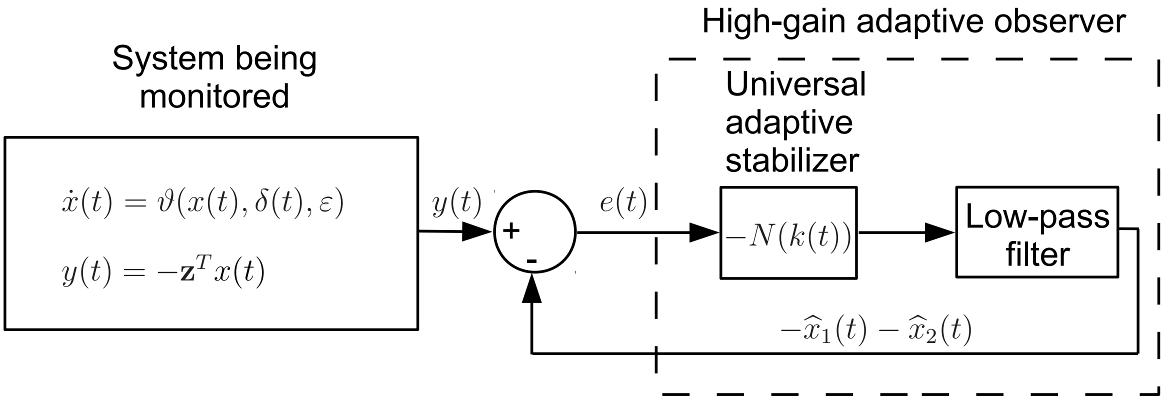

High-gain adaptive observer
System being monitored
$\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$
$y(t) = -\mathbf{z}^T x(t)$
$y(t)$
+
-
$e(t)$
Universal adaptive stabilizer
$-N(k(t))$
Low-pass filter
$-\widehat{x}_1(t) - \widehat{x}_2(t)$

Figure 18: Monitoring system states using a high-gain observer based on universal adaptive stabilization (UAS).

5.2.1 A high gain adaptive observer for monitoring the reliability of an RFIS

An observer is considered for monitoring an RFIS for the following reasons. The RFIS being monitored may be computed based on a nominal system model. Such a model may account for perturbations, but may not necessarily have accounted for parametric uncertainties. An accurate model of how the parametric uncertainty enters a system may not be available. In such a situation, relying on the states of the nominal system model may not necessarily show that an RFIS computed using a nominal system model is about to cease to be invariant. This is because the nominal system model may be very different from the actual perturbed system with parametric uncertainties. Constructing an observer by assuming a simple model for the system being monitored, so that the states of the observer provide a reliable indication of system behavior even in the presence of modeling errors is the motivation behind the observer design presented further.

5.2.1.1 A perturbed system model, with a specific type of output equation

Consider the following system,

$$\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon) \tag{62}$$

$$y(t) = -\mathbf{z}^T x(t), \tag{63}$$

here $x(t) \in \mathbb{R}^2$ and $\mathbf{z} = [z_1 \;\; z_2]^T$ where z_1, z_2 are positive real numbers, and $\vartheta(\cdot, \cdot, \cdot)$ is as defined in Section 5.1.

5.2.1.2 Developing a high-gain adaptive observer

Along the lines of the material in Chapter 4, consider the Mittag-Leffler function $E_\alpha \colon \mathbb{R} \to \mathbb{R}$, given by

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha + 1)}, \tag{64}$$

where α is a parameter, $z \in \mathbb{R}$, and $\Gamma(z+1) = z\Gamma(z), z > 0$ is the standard Gamma function. It is shown in [38], [39] that $E_\alpha(-\lambda t^\alpha), t \in \mathbb{R}$ is a Nussbaum function for $\lambda > 0$ and $\alpha \in (2, 3]$. Now consider the following Nussbaum function $N(k(t))$ given by,

$$N(k(t)) = E_\alpha\left(-\lambda k(t)^\alpha\right), \lambda > 0, \alpha \in (2, 3). \tag{65}$$

The definitions in Equations (64) and (65) can be utilized to formulate the following high-gain adaptive observer

$$\left.\begin{aligned} \dot{\widehat{x}}_1(t) &= -c\widehat{x}_1(t) - N(k(t))e(t) \\ \dot{\widehat{x}}_2(t) &= -c\widehat{x}_2(t) - N(k(t))e(t) \\ \dot{k}(t) &= e^2(t) \\ e(t) &= y(t) + \widehat{x}_1(t) + \widehat{x}_2(t). \end{aligned}\right\} \tag{66}$$

Note that here $y(t)$ is as given in Equation (63).

5.2.1.3 Convergence analysis

Note that the following result, which is related to the convergence of the high-gain adaptive observer presented in Equation (66), is similar to that presented in Lemma 4.2.2. The only difference is that here the dynamics of the system being monitored are of the form $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$ with the output $y(t)$ given by Equation (63). In Lemma 4.2.2, the dynamics represent the real Li-ion battery being monitored, and the output $y(t)$ represents the battery terminal voltage.

Lemma 5.2.1. *Consider the closed-loop dynamics in Equation (66). Let $c > \frac{1}{2}$ be a real positive constant. Let $d(t) = cy(t) + \dot{y}(t)$, and let the initial condition $k(t_0)$ be an arbitrary real number where $t_0 \geq 0$. If $\left(\int_{t_0}^{\infty} |d(\tau)|^2 \, d\tau\right)^{\frac{1}{2}} < \infty$, then $\lim_{t\to\infty} k(t) = k_\infty$ and $\lim_{t\to\infty} e(t) = 0$, where k_∞ is a finite real constant. The constant k_∞ depends on the initial error $e(t_0)$.*

Proof. By (66),

$$\dot{e}(t) = \dot{y}(t) + \dot{\widehat{x}}_1(t) + \dot{\widehat{x}}_2(t) \tag{67}$$

$$\dot{e}(t) = \dot{y}(t) - c\widehat{x}_1(t) - c\widehat{x}_2(t) - 2N(k(t))e(t). \tag{68}$$

Add and subtract $cy(t)$ to the right hand side of Equation (68) to get,

$$\dot{e}(t) = \dot{y}(t) + cy(t) - c(y(t) + \widehat{x}_1(t) + \widehat{x}_2(t)) - 2N(k(t))e(t) \tag{69}$$

Using the fact that $d(t) = cy(t) + \dot{y}(t)$, and $e(t) = y(t) + \widehat{x}_1(t) + \widehat{x}_2(t)$ the following system in the error $e(t)$ and state $k(t)$ is obtained.

$$\left.\begin{aligned} \dot{e}(t) &= -\left(c + 2N(k(t))\right)e(t) + d(t) \\ \dot{k}(t) &= e^2(t) \end{aligned}\right\} \tag{70}$$

The remainder of this proof is identical to the proof of the second result in Lemma 4.2.2, and therefore the details have not been repeated. For the steps of the proof, readers are requested to refer to Lemma 4.2.2. □

The above result shows that if $\left(\int_{t_0}^{\infty} |d(\tau)|^2\, d\tau\right)^{\frac{1}{2}} < \infty$, then $k(t) \to k_\infty$, and $e(t) \to 0$ as $t \to \infty$. In general the condition $\left(\int_{t_0}^{\infty} |d(\tau)|^2\, d\tau\right)^{\frac{1}{2}} < \infty$ may not be true for a given system, therefore the requirement of such a condition makes the above result restrictive.

5.2.1.4 Detecting system instability

The following result is similar to Theorem 4.2.4. While Theorem 4.2.4 is related to detecting Li-ion battery terminal voltage collapse using Algorithm 2, the following result aims at detecting system instability for the type of system shown in Equations (62) and (63) using Algorithm 2. The variables β, r, γ, and ϵ are required to be chosen for using Algorithm 2. As described in Section 4.2.2, choices for the variables β, r, γ, and ϵ depend on a particular application and must be experimentally tuned.

Theorem 5.2.2. *Assume that the conditions required for Lemma 5.2.1 to hold are satisfied. Let* $\mathbf{z} \in \mathbb{R}^2$, *and* $x(t)$ *be a solution to the system* $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$. *Let* $j \in \{1, 2\}$, $\beta > 0$, $\gamma > 1$, *and* σ *be a small positive constant. For all* $j \in \{1, 2\}$ *let* $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$ *and* $p = \frac{1}{q}$. *Let* $\widehat{x}_{j_{min}} = \min\left(\widehat{x}_j(\tau)_{\tau \in [0,t]}\right)$, *and* $p_{max} = \max\left(p(\tau)_{\tau \in [t-r,t]}\right)$, $0 < r < t$. *Assuming* $\widehat{x}_{j_{min}}$ *is bounded for all* $j \in \{1, 2\}$, $\| \mathbf{z} \|_2$ *is bounded, and* $p(t)$ *is continuously differentiable for all* $t \in [0, \infty)$, *if* $\dot{p}(t) \leq -\beta p(t)$ *whenever* $\gamma p(t) \geq p_{\max}$, *then* $\| x(t) \|_2 \to \infty$ *as* $t \to \infty$.

Proof. By definition $\widehat{x}_j \geq \widehat{x}_{j_{min}}$ and σ is a positive number. Therefore $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$ is positive, and hence $p = \frac{1}{q}$ is also positive. Let $V(p(t)) = p(t)$ and $g(l) = \gamma l$, where $l \in \mathbb{R}$, and $\gamma > 1$. By definition $p_{max} = \max\left(p(\tau)_{\tau \in [t-r,t]}\right)$, $0 < r < t$. Therefore, $0 < p(\tau) \leq p_{\max}$ for $\tau \in [t - r, t]$, where $0 < r < t$. Notice that $g(l)$ is non-decreasing, and from our assumptions $V(p(t))$ is continuously differentiable. Therefore the condition $V(\cdot) \leq g(V(\cdot))$ in the Razumikhin theorem [80, Theorem 5.1] holds if $p_{\max} \leq \gamma p(t)$. Let $w(p(t)) = \beta p(t)$ where $\beta > 0$. The function $w(\cdot)$ is positively valued for non-zero p. Further $\dot{V}(p(t)) = \dot{p}(t)$. Using the Razumikhin theorem [80, Theorem 5.1] we get that if $\dot{p}(t) \leq -\beta p(t)$ whenever $\gamma p(t) \geq p_{\max}$ then $p(t) \to 0$ as $t \to \infty$ for all $j \in \{1, 2\}$. Since $q = \frac{1}{p}$, this gives $q \to \infty$ as $t \to \infty$ for all $j \in \{1, 2\}$. However $q = \widehat{x}_j - \widehat{x}_{j_{min}} + \sigma$, σ is a constant, and for all $j \in \{1, 2\}$ $\widehat{x}_{j_{min}}$ is bounded by assumption. From the above discussion it follows that for all $j \in \{1, 2\}$, $\widehat{x}_j \to \infty$ as $t \to \infty$. From the assumptions, Lemma 5.2.1 holds, i.e. as time tends to infinity, the error $e(t) = y(t) + \widehat{x}_1(t) + \widehat{x}_2(t) \to 0$. Therefore, $y(t) \to -\infty$ as $t \to \infty$. From Equation (63) $y(t) = -\mathbf{z}^T x(t)$, where $x(t) \equiv [x_1(t) \;\; x_2(t)]^T$. By the Cauchy-Schwarz inequality we have that $\| y(t) \|_2 \leq \| \mathbf{z} \|_2 \| x(t) \|_2$. Since $y(t) \to -\infty$, therefore $\| y(t) \|_2 \to \infty$ with time. Further since $\| \mathbf{z} \|_2$ is bounded by assumption, this implies that $\| x(t) \|_2 \to \infty$ as $t \to \infty$. This completes the proof. □

Suppose a bounded RFIS $\mathcal{S}$ is computed based on a nominal model of a perturbed system without parametric uncertainties given by $\dot{x}(t) = \vartheta(x(t), \delta(t), 0)$, and it is known that $\mathcal{S}$ is invariant for the time interval $[t_0, t_1]$. Suppose the original system given by $\dot{x}(t) = \vartheta(x(t), \delta(t), 0)$ is no longer valid, but the system has evolved into a perturbed system with parametric uncertainties of the form $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$ given by Equations (62) and (63). Further suppose that monitoring the states $\widehat{x}_1$ and $\widehat{x}_2$ of the high-gain adaptive observer in Equation (66) using Algorithm 2 from Chapter 4 causes the indicator variable S to be set to 1 at some time instant within the interval $[t_0, t_1]$. This means that $x_1(t)$ and $x_2(t)$ tend to infinity as $t \to \infty$. Since $\mathcal{S}$ is a bounded set and $x_1(t)$ and $x_2(t)$ tend to infinity as $t \to \infty$, therefore Algorithm 2 has detected that states $x_1(t)$ and $x_2(t)$ will eventually leave the RFIS $\mathcal{S}$. Therefore, for a specific type of systems given by Equations (62) and (63), Algorithm 2 can be used to detect that an RFIS $\mathcal{S}$ is about to cease to be invariant.

If parametric uncertainty does not cause instability, but is bounded, then it is possible to compute an updated set which may provide an estimate of the region in which trajectories of a perturbed system with parametric uncertainty may stay for a given time interval. The following section discusses such a case.

5.3 Estimating an RFIS in the presence of bounded parametric perturbations

Consider a system of the form

$$\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon), \;\; x(t_0) = \eta(\varepsilon) \tag{71}$$

where $\delta(\cdot)$ represents a perturbation term affecting the system and ε is an unknown parametric uncertainty, and $\eta : \mathbb{R} \to \mathbb{R}^2$ and is C^1 in ε. Suppose an RFIS is computed for the system $\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0)$. An estimate of the region (i.e. a set) in which trajectories of the perturbed system with parametric uncertainties given by $\dot{x}(t) = \vartheta(x(t), \delta(t), \varepsilon)$ lie, is desired. Note that the system $\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0) = f(x(t)) + \delta(t)$ is identical to the system used in Chapter 3.

If the parametric perturbation is bounded, then ideally we desire to use results from perturbation theory [32] to quickly estimate the shape/size of a new RFIS, given an RFIS for the nominal system $\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0)$. This has not been achieved in this work yet and remains an open problem. Suppose $\vartheta : \mathcal{D} \times \mathcal{U} \times [-\varepsilon_0, \varepsilon_0] \to \mathbb{R}^2$ is continuous in (t, x, ε), and locally Lipschitz in (x, ε), uniformly in t. Suppose η is continuous, and locally Lipschitz in ε. Let $x(t) \in \mathcal{D} \subset \mathbb{R}^2$ be a solution to Equation (71). Suppose the nominal system

$$\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0), \;\; x(t_0) = \eta_0, \tag{72}$$

has a unique solution $x_0(t) \in \mathcal{D}$ for all initial conditions $x_0(t_0) \in \mathcal{D}$, and all $t \in [t_0, t_1]$. Here $\eta_0 = \eta(0)$. An approximation of the solution $x(t)$ for the system in Equation (71) can be constructed using a finite Taylor series expansion as follows,

$$x(t) = \sum_{l=0}^{L-1} x_l(t)\varepsilon^l + \varepsilon^l R_x(t, \varepsilon). \tag{73}$$

For finding the coefficients for the terms ε^l, and to determine the remainder term $\varepsilon^l R_x(t, \varepsilon)$, take the time derivative of terms in Equation (73) and use Equation (71) to get

$$\sum_{l=0}^{L-1} \dot{x}_l(t)\varepsilon^l + \varepsilon^l \dot{R}_x(t, \varepsilon) = \vartheta(x(t), \delta(t), \varepsilon). \tag{74}$$

Let $L = 1$, and then set $\varepsilon = 0$. This gives the following

$$\dot{x}_0(t) = \vartheta(x(t), \delta(t), 0), \tag{75}$$

which is the nominal system in Equation (72). Now let $\mathcal{S} \subset \mathbb{R}^2$, with a simple closed curve $\partial\mathcal{S}$ as its boundary, represent an RFIS for the system in Equation (75). Given positive real numbers k and $\varepsilon^\star$, let the set $\mathcal{S}' = \mathcal{S} \bigcup_{p \in \partial\mathcal{S}} \mathcal{B}_{k\varepsilon^\star}(p)$, for all $p \in \partial\mathcal{S}$. Letting $N = 1$ in [32, Theorem 10.1] it can be shown that $x(t) - x_0(t) = O(\varepsilon)$. This leads to the conclusion that the set $\mathcal{S}'$ may not be an RFIS because the trajectories of a perturbed system with parametric uncertainty stay in the set $\mathcal{S}'$, only if the initial conditions are in $\mathcal{S}$.

Suppose $\varepsilon^\star$ and the constant k can be known and a set $\mathcal{S}'$ is found as shown above. This provides a straightforward way to find the size of a set in which the solutions of a perturbed system with parametric uncertainty stay, given an RFIS for a nominal perturbed system without parametric uncertainty, provided initial conditions of the perturbed system with parametric uncertainty are in the given RFIS.

5.4 Simulations

Consider the following perturbed system with parametric uncertainties where $\delta(t) = (\delta_0 + \varepsilon)\sin(t)$ is the perturbation with $\delta_0 = 0.15$ and the parametric uncertainty $\varepsilon = 0.05$.

$$\dot{\rho} = -\sin(\phi) \tag{76}$$

$$\dot{\phi} = (\rho - \rho_0)\cos(\phi) - \mu\sin(\phi) + (\delta_0 + \varepsilon)\sin(t), \rho_0 = 1, \mu = 6.42. \tag{77}$$

Note that for $\varepsilon = 0$, the system in Equations (76) and (77) is identical to the nominal perturbed system without parametric uncertainties in Equations (16) and (17), which is shown below for convenience.

$$\dot{\rho} = -\sin(\phi) \tag{78}$$

$$\dot{\phi} = (\rho - \rho_0)\cos(\phi) - \mu\sin(\phi) + \delta(t), \rho_0 = 1, \mu = 6.42, \tag{79}$$

where $\delta(t) = \delta_0\sin(t)$ and $\delta_0 = 0.15$. For the system in Equations (76) and (77), Figure 19 shows the results obtained based on the application of [32, Theorem 10.1] to find an approximation of the set $\mathcal{S}'$ as discussed in Section 5.3.

In Figure 19 the red curve with square markers represents an approximation of the smallest RFIS computed using Algorithm 1 for the nominal system in Equations (78) and (79). The black curve with circular markers represents an approximation of the smallest RFIS computed using Algorithm 1 for the perturbed system with parametric uncertainty in Equations (76) and (77). The blue solid curve shows an estimated set in which trajectories of the system in Equations (76) and (77) remain, if

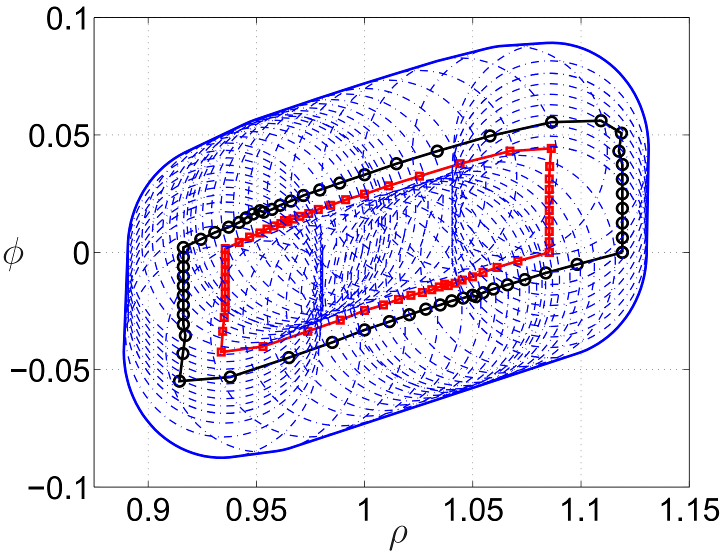


Figure 19: Using perturbation theory to estimate the size of a set in which trajectories of a perturbed system with parametric uncertainty may remain, given an approximation of the smallest RFIS for the nominal system.

initial conditions the system in Equations (76) and (77) belong to the region enclosed by the red curve with square markers. The convex hull (blue solid curve) is obtained by considering the convex hull of the set formed by the union of all the balls (circles represented by dashed blue curves) centered at points on the boundary of the red curve with square markers. The red curve with square markers is the set $\mathcal{S}$ and the blue solid curve is the set $\mathcal{S}'$. The values for k and ε^* used are 0.9 and 0.05 respectively.

Figure 20 shows the results in Figure 19 overlaid with a trajectory of the nominal perturbed system without parametric uncertainty, and a trajectory of the perturbed system with parametric uncertainty. In Figure 20, the dashed red curve is a trajectory of the nominal perturbed system without parametric uncertainty given in Equations (78) and (79). The dotted black curve in Figure 20, is a trajectory of the perturbed system with parametric uncertainty in Equations (76) and (77). From Figure 20 it is seen that the trajectory of the perturbed system with parametric uncertainty stays within the approximation of the smallest RFIS for the perturbed system with parametric uncertainty (black curve with circular markers). Similarly, the trajectory

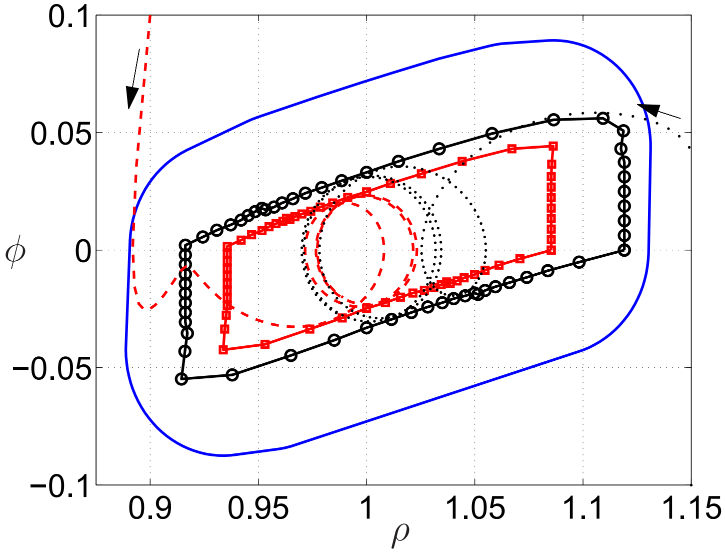


Figure 20: Comparing sizes of approximations smallest RFISs for a perturbed system with parametric uncertainty and a perturbed system without parametric uncertainty to a set computed based on perturbation theory, in which trajectories of a perturbed system with parametric uncertainty are expected to belong.

for the nominal perturbed system without parametric uncertainty stays within the approximation of the smallest RFIS for the nominal perturbed system without parametric uncertainty (red curve with square markers). Both trajectories remain within the set with boundary given by the solid blue curve, which is the estimate obtained using perturbation theory. Note that the size of this estimate depends on the choices of k, and ε^*, and as visible in Figures 19 and 20 the estimate obtained using methods based on perturbation theory is much more conservative compared to the results obtained using Algorithm 1. This can be attributed to the fact that an estimate based on perturbation theory is developed based on the zeroth order term of the Taylor expansion of the solution $x(t)$ for the perturbed system with parametric uncertainty.

5.4.1 Detecting that a computed approximation of an RFIS is about to cease to be invariant

The problem used in this section is the nominal curve tracking problem as given in Equations (78) and (79), with a few changes. For notational consistency with the material presented in Section 5.2.1, the states of a system being monitored are represented using x_1, and x_2. The version of the curve tracking problem including perturbations and parametric uncertainty, which is used for the simulations in this section, is as follows.

$$\dot{x}_1 = -\varepsilon(t)\sin(x_2) \tag{80}$$

$$\dot{x}_2 = (x_1 - \bar{x}_1)\cos(x_2) - \mu\sin(x_2) + \delta(t), \bar{x}_1 = 1, \mu = 6.42, \tag{81}$$

where $\delta(t) = \delta_0 \sin(t)$ and $\delta_0 = 0.15$. The parametric uncertainty ε is defined as,

$$\dot{\varepsilon}(t) = \begin{cases} 0,\ t \leq \tau, \tau \in \mathbb{R},\ \text{and}\ \tau > t_0 > 0 \\ -t,\ t > \tau \end{cases} \tag{82}$$

with $\varepsilon(t_0) = 1$.

As long as the time $t \leq \tau$, the problem in Equation (80) and (81) is almost similar to the nominal problem in Equations (78) and (79). The parametric uncertainty $\varepsilon(t)$ does not appear in Equation (78). Equation (80) and Equation (78) are identical at time t_0 when $\varepsilon = 1$.

Suppose that the model used to calculate an approximation of the smallest RFIS considered the parametric uncertainty ε to be added to δ_0 as in Equations (76) and (77). Then the approximation of the smallest RFIS obtained using Algorithm 1 is shown using the red curve with square markers in Figure 21. However, suppose that the parametric uncertainty ε is not constant but is given by Equation (82), and enters the system as given in Equations (80) and (81). The trajectory for the system in Equations (80) and (81) is shown in Figure 21 using the black dashed curve for $t \leq \tau$, and using black circular markers when $t > \tau$. It is seen that when $t < \tau$, the system trajectory enters the approximation of the smallest RFIS (red curve with square markers), inside which the trajectory keeps circling until $t = \tau$. When $t > \tau$, x_1 is seen to increase rapidly causing the system trajectory to leave the computed approximation of the smallest RFIS. The aim of the simulation results presented in this section is to detect that the behavior of the system in Equations (80) and (81)

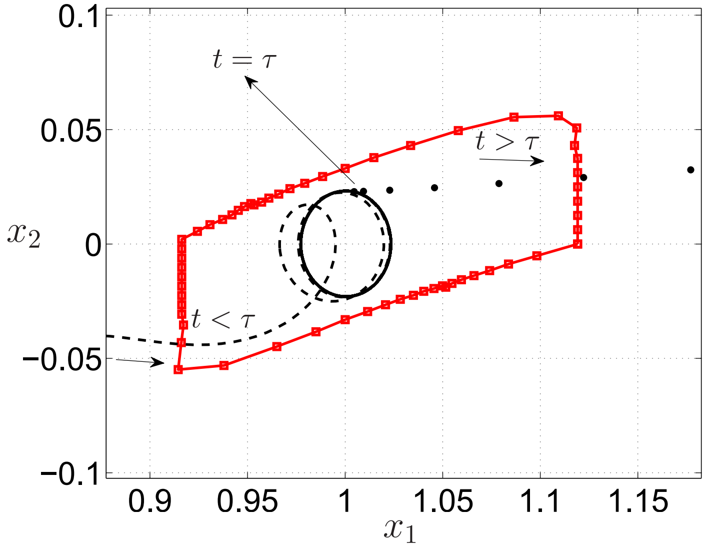


Figure 21: An illustration showing that modeling errors can cause a computed approximation of the smallest RFIS to cease to be invariant. Here $\tau = 950$ and $t \in [0, 1000]$ seconds.

is about to change before $t = \tau$. This is desired, because such knowledge may be useful in updating other decision algorithms which may rely on a system trajectory staying within a computed approximation of the smallest RFIS. To achieve this, the high-gain adaptive observer in Equation (66), which is shown in Figure 18, is used with the system in Equations (80) and (81) and the following output equation:

$$y(t) = -\mathbf{z}^T x(t), \text{ where } \mathbf{z}^T = [1\ \ 1]^T. \tag{83}$$

The results are presented in Figure 22 and Figure 23. Figure 22 shows a comparison of the trajectories of the high-gain adaptive observer in Equation (66), and the perturbed version of the curve tracking problem with parametric uncertainty, given in Equations (80) and (81). The dash-dotted blue curve shows the trajectory of the high-gain adaptive observer, i.e. $\widehat{x}_1$ vs $\widehat{x}_2$. The dashed red curve shows the trajectory x_1 vs x_2 of the system in Equations (80) and (81). The portions of both trajectories for time $t > \tau$ is plotted using circular markers. When $t \leq \tau$ the trajectory x_1 vs x_2 is seen to circle around the point $(1, 0)$. The adaptive high-gain observer tries to mimic such behavior, which is why a portion of the trajectory $\widehat{x}_1$ vs $\widehat{x}_2$ near $t = \tau$ appears solid.

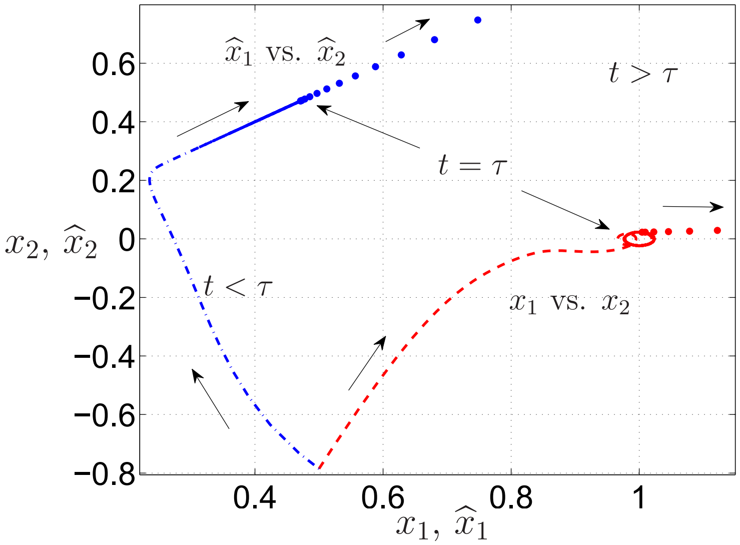


Figure 22: Comparing the trajectory of the high-gain adaptive observer with the trajectory of a perturbed version of the curve tracking problem with parametric uncertainty given in Equations (80) and (81), where $\tau = 950$ and $t \in [0, 1000]$ seconds.

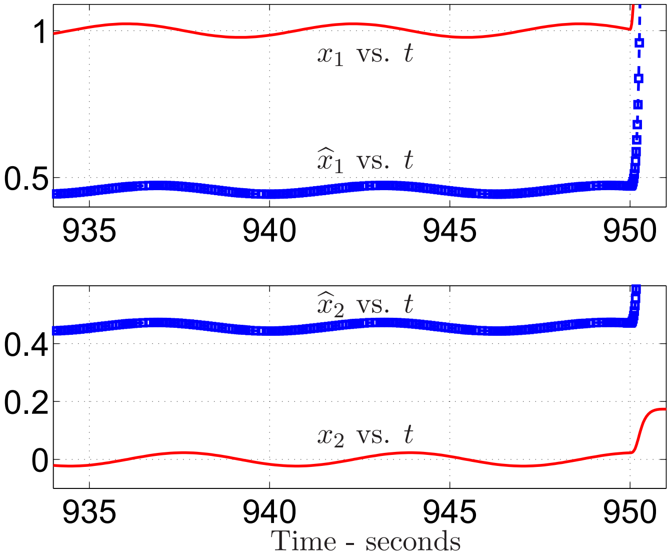


Figure 23: An illustration examining the states of the high-gain adaptive observer, and the states of a perturbed version of the curve tracking problem with parametric uncertainty given in Equations (80) and (81), where $\tau = 950$ and $t \in [0, 1000]$ seconds.

This is because the states $\widehat{x}_1$, and $\widehat{x}_2$ oscillate around $(0.45, 0.45)$ as seen in Figures 22 and 23. From Figure 23 it is obvious that the lack of a sophisticated model used in the high-gain adaptive observer causes some discrepancy between the real states x_1, and x_2 and the estimated states $\widehat{x}_1$, and $\widehat{x}_2$. However this does not prevent detecting that the computed approximation of an RFIS (which may not always necessarily be an RFIS) shown in Figure 21 is at risk of ceasing to be invariant due to a change in behavior near time $t = \tau$. Figure 22 clearly shows that the states of the high-gain adaptive observer experience a sudden increase after time $t = \tau$.

This effect is examined more clearly in Figure 24. The state $\widehat{x}_1$ of the high-gain adaptive observer is used with Algorithm 2 to detect if the computed approximation of an RFIS (which may not always necessarily be an RFIS) for the system in Equations (76) and (77) is about to cease to be invariant in the time interval $[0, 1000]$ due to modeling errors, given that the actual system dynamics are represented by Equations (80) and (81) where ε is given by Equation (82). The leftmost panel of Figure 24 shows the state x_1 of the system in Equations (80) and (81) by a solid red curve. The state $\widehat{x}_1$ of the high-gain adaptive observer given by Equation (66) is shown in this panel by the blue dashed curve with square markers. After $t = 950s$ both x_1, and $\widehat{x}_1$ appear to go to infinity with time. The middle panel in Figure 24 shows the tracking error $e(t)$ of the high-gain adaptive observer in Equation (66). For this simulation, the value 0.09 is used for ϵ in the error guard condition $\max\left(|e(\tau)|_{\tau\in[t-r,t]}\right) \le \epsilon$ in

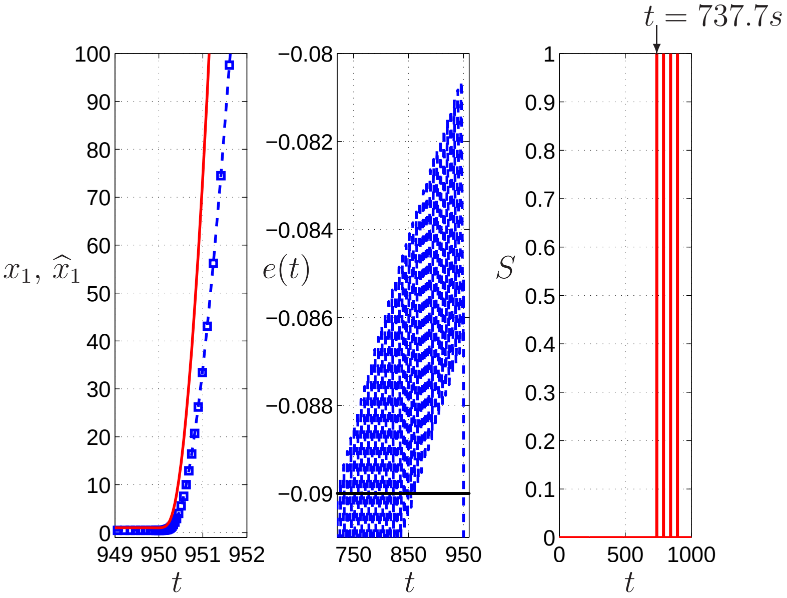


Figure 24: The states $x_1, \widehat{x}_1$, tracking error $e(t)$ and the indicator variable S used in Algorithm 2.

Algorithm 2. The solid black horizontal line in the middle panel of Figure 24 shows the lower bound of acceptable values for $e(t)$ satisfying the error guard condition. After approximately time $t = 720s$ there exist many time instances at which the error $e(t)$ satisfies the error guard condition. Further, from the right most panel of Figure 24, it is observed that at time $t = 737.7s$, Algorithm 2 detects that the behavior of the system in Equations (80) and (81) has changed drastically so that $\widehat{x}_1$ and therefore x_1 will go to infinity with time (for justification as to why x_1 will go to infinity with time, the readers are requested to refer to the proof of Theorem 5.2.2). Therefore Algorithm 2, sets the indicator variable $S = 1$. The values used for the constants r, γ, β, and ϵ for use in Algorithm 2 are $50s$, 1.0001, 0.5 and 0.09 respectively. Also, the value $c = 2$ is used for the high-gain adaptive observer in Equation (66).

This (the indicator variable S being set to 1) can alert a system designer who computed the approximation of the smallest RFIS for the system in Equations (76) and (77) with limited model information (the actual model for which obeys Equations (80) and (81) with ε given by Equation (82)) that the computed approximation of the smallest RFIS is about to cease to be invariant. Note from the right most panel in Figure 24, Algorithm 2 detects that a computed approximation of the smallest RFIS is about to cease to be invariant at time $t = 737.7s$. This is before time $t = 950s$, i.e., before system behavior changes and the states leave the computed approximation of the smallest RFIS. This indicator variable S produced by Algorithm 2 can therefore be used by a higher-level monitoring algorithm to shut down appropriate devices or processes, which may be jeopardized by such an abrupt change in system behavior.

As a summary, this chapter presented methods for monitoring RFISs (or approximations of RFISs) computed for systems with bounded additive perturbations, in the presence of parametric uncertainties. If such uncertainties are bounded, then a method based on perturbation theory may be used to generate an estimate of the size of a region in which trajectories of a given perturbed system with parametric uncertainties may stay, if the initial conditions belong to an RFIS of a nominal perturbed system without parametric uncertainties.

If an RFIS (or an approximation of an RFIS) has been computed for a system with limited model information, it is possible that some critical aspects of system behavior have gone undetected. For example, an unsophisticated model may not be able to tell that parametric uncertainties may cause system instability. If a system designer relies solely on an approximation of the smallest RFIS computed using Algorithm 1, and is unaware of such an impending loss in system stability, then the system under consideration can experience adverse effects. The high-gain adaptive observer, coupled with the change in trend detection algorithm developed in Chapter 4, has been successfully used to detect such a loss in system stability, for a particular type of systems. In the event of a loss of stability for such types of systems, it is possible to detect that a computed RFIS (or an approximation of an RFIS) is about to cease to be invariant, by using Algorithm 2 along with the high-gain adaptive observer in Equation (66), even if a sophisticated system model is unavailable.

CHAPTER VI

CONCLUSION AND FUTURE DIRECTION

The main contribution of this work is a formulation of the problem of computing an approximation of the smallest robust forward invariant sets (RFISs) for two dimensional systems subjected to bounded additive perturbations, and using path planning algorithms for solving the problem. Leveraging a path planning algorithm for computing an approximation of the smallest RFIS for a given nonlinear system is a key idea utilized in this work.

The secondary contribution of this work includes an algorithm to monitor the validity of an approximation of an RFIS. Inspiration for this comes from a high-gain adaptive observer and a trend filter used to detect changes in trend in the states of a physical Li-ion battery. This method is used to detect Li-ion battery terminal voltage collapse. The method is based on classical techniques like universal adaptive stabilization. The method presented has been theoretically justified, and also verified by simulations and experiments. This method only requires measurements of the terminal voltage, and does not require measurements of the discharge current. The method is less susceptible to false alarms which are a concern to static threshold based systems. Since the method does not require a detailed battery model, it may be robust to temperature variations, aging effects, changes in loading or other nonlinear disturbances.

The idea behind the above method is to detect abrupt changes in system behavior, which causes instability, even in the absence of a sophisticated model. This method has been successfully used, with a particular type of perturbed systems with parametric uncertainty, to detect that a computed approximation of an RFIS is about to cease to be invariant. This work focuses on parametric uncertainties which cause system instability. If such an instability can be detected, then it can be said that an RFIS (or an approximation of an RFIS) computed at an earlier instant of time is not reliable anymore and should not be trusted, because detecting system instability implies that a computed RFIS (or an approximation of an RFIS) is about to cease to be invariant. The method presented in Chapter 5 detects such an instability even in the absence of a sophisticated model.

6.1 Future work

The ideas developed in this work may be applied to compute approximations of the smallest RFIS for higher dimensional systems in future.

The terminal voltage collapse detection algorithm for Li-ion batteries, which is used to detect that a computed approximation of an RFIS is about to become unreliable, may be generalized further to detect abrupt changes in system behavior for a wider class of systems. It may be specialized to detect specific types of failures studied in different fields of engineering. For example detecting a bridge collapse may require a fundamentally different type of system model/algorithm compared to a model/algorithm used for detecting that wireless network connectivity is soon to be lost. It is conjectured that the idea of terminal voltage collapse detection based on limited model information can be extended to deal with such situations.

REFERENCES

[1] Abu-Sharkh, S. and Doerffel, D., "Rapid test and non-linear model characterisation of solid-state Lithium-ion batteries," *Journal of Power Sources*, vol. 130, no. 1-2, pp. 266–274, 2004.

[2] Alamo, T., Fiacchini, M., Cepeda, A., Limon, D., Bravo, J., and Camacho, E., "On the computation of robust control invariant sets for piecewise affine systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control* (Findeisen, R., Allgöwer, F., and Biegler, L., eds.), vol. 358 of *Lecture Notes in Control and Information Sciences*, pp. 131–139, Springer Berlin Heidelberg, 2007.

[3] Baier, R., Dellnitz, M., Molo, M.-V., Kevrekidis, I. G., and Sertl, S., "The computation of convex invariant sets via Newton's method," Technical Report, University of Paderborn, Germany, May 2010.

[4] Baier, R. and Molo, M.-V., "Newtons method and secant method for set-valued mappings," in *Large-Scale Scientific Computing* (Lirkov, I., Margenov, S., and Waniewski, J., eds.), vol. 7116 of *Lecture Notes in Computer Science*, pp. 91–98, Springer Berlin Heidelberg, 2012.

[5] Balakrishnan, V., Boyd, S., and Balemi, S., "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *International Journal of Robust and Nonlinear Control*, vol. 1, no. 4, pp. 295–317, 1991.

[6] Barreiro, A., Aracil, J., and Pagano, D., "Detection of attraction domains of non-linear systems using bifurcation analysis and Lyapunov functions," *International Journal of Control*, vol. 75, no. 5, pp. 314–327, 2002.

[7] Beyn, W.-J., "The effect of discretization on homoclinic orbits," in *Bifurcation: analysis, algorithms, applications: Proceedings of the conference at the University of Dortmund*, vol. 79 of *International series of numerical mathematics*, pp. 1–8, Birkhaüser, Basel, Aug. 1986.

[8] Beyn, W.-J., "Numerical methods for dynamical systems," in *Advances in numerical analysis*, vol. 1, pp. 175–236, Oxford science publications, Oxford: Clarendon press, 1991.

[9] Beyn, W.-J. and Lorenz, J., "Center manifolds of dynamical systems under discretization," *Numerical Functional Analysis and Optimization*, vol. 9, no. 3-4, pp. 381–414, 1987.

[10] Brockett, R., "The status of stability theory for deterministic systems," *IEEE Transactions on Automatic Control*, vol. 11, pp. 596–606, Jul. 1966.

[11] Bullinger, E. and Allgower, F., "An adaptive high-gain observer for nonlinear systems," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, pp. 4348–4353, Dec 1997.

[12] Camilli, F., Grüne, L., and Wirth, F., "A regularization of Zubov's equation for robust domains of attraction," in *Nonlinear control in the Year 2000* (Isidori, A., Lamnabhi-Lagarrigue, F., and Respondek, W., eds.), vol. 258 of *Lecture Notes in Control and Information Sciences*, pp. 277–289, Springer London, 2000.

[13] Chen, C.-T., *Linear System Theory and Design.* Oxford University Press, 3rd edition ed., 1998.

[14] Chen, M. and Mora, R., "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Transcations on Energy Conversion*, vol. 21, pp. 504–512, June 2006.

[15] Chesi, G., "Estimating the domain of attraction for uncertain polynomial systems," *Automatica*, vol. 40, no. 11, pp. 1981–1986, 2004.

[16] Chiasserini, C.-F. and Rao, R., "Energy efficient battery management," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1235–1245, Jul 2001.

[17] Clarke, F., Ledyaev, Y., Stern, R., and Wolenski, P., *Nonsmooth Analysis and Control Theory.* Springer, 1998.

[18] Coleman, M., Hurley, W., and Lee, C., "An improved battery characterization method using a two-pulse load test," *IEEE Transactions on Energy Conversion*, vol. 23, pp. 708–713, June 2008.

[19] Dattoro, J., *Convex Optimization and Euclidean Distance Geometry.* $\mathcal{M}\varepsilon\beta oo$ Publishing, 2005.

[20] Davison, E. J. and Kurak, E. M., "A computational method for determining quadratic Lyapunov functions for non-linear systems," *Automatica*, vol. 7, pp. 627–636, Sep. 1971.

[21] Ding, S., *Model Based Fault Diagnosis Techniques.* Springer, 2008.

[22] Garay, B. M., "Discretization and Morse-Smale dynamical systems on planar discs.," *Acta Mathematica Universitatis Comenianae. New Series*, vol. 63, no. 1, pp. 25–38, 1994.

[23] Genesio, R., Tartaglia, M., and Vicino, A., "On the estimation of asymptotic stability regions: State of the art and new proposals," *IEEE Transactions on Automatic Control*, vol. 30, pp. 747–755, Aug. 1985.

[24] Gold, S., "A PSPICE macromodel for Lithium-ion batteries," in *Proceedings of the Twelfth Annual Battery Conference on Applications and Advances*, pp. 215–222, jan 1997.

[25] Grüne, L., "Attractors under perturbation and discretization," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, pp. 2118–2122, 2000.

[26] Grüne, L., *Asymptotic Behavior of Dynamical and Control Systems under Pertubation and Discretization*, vol. 1783. Springer, 2002.

[27] Hart, P. E., Nilsson, N. J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[28] Ilchmann, A., *Non-Identifier Based High Gain Adaptive Control.* Lecture Notes in Control and Information Sciences 189, Springer-Verlag, 1993.

[29] Jankovic, M., "Adaptive nonlinear output feedback tracking with a partial high-gain observer and backstepping," *IEEE Transactions on Automatic Control*, vol. 42, pp. 106–113, Jan 1997.

[30] Katok, A. and Hasselblatt, B., *Introduction to the Modern Theory of Dynamical Systems.* Cambridge University Press, 1996.

[31] Kelley, C. T., *Iterative Methods for Linear and Nonlinear Equations.* Society for Industrial and Applied Mathematics, 1995.

[32] Khalil, H. K., *Nonlinear Systems.* Prentice Hall, 3 ed., 2002.

[33] Kim, H. and Shin, K. G., "On dynamic reconfiguration of a large-scale battery system," in *RTAS '09: Proceedings of the 2009 15th IEEE Symposium on Real-Time and Embedded Technology and Applications*, (Washington, DC, USA), pp. 87–96, IEEE Computer Society, 2009.

[34] Klein, R., Chaturvedi, N. A., Christensen, J., Ahmed, J., Findeisen, R., and Kojic, A., "Electrochemical model based observer design for a Lithium-ion battery," *IEEE Transactions on Control Systems Technology*, vol. X, no. 99, pp. 1–13, 2012.

[35] Kloeden, P. E. and Lorenz, J., "Stable attracting sets in dynamical systems and in their one-step discretizations," *SIAM Journal on Numerical Analysis*, vol. 23, no. 5, pp. 986–995, 1986.

[36] Knauff, M., Dafis, C., Niebur, D., Kwatny, H., and Nwankpa, C., "Simulink model for hybrid power system test-bed," in *Proceedings of the IEEE Electric Ship Technologies Symposium*, pp. 421–427, May 2007.

[37] Kolmanovsky, I. and Gilbert, E. G., "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering*, vol. 4, no. 4, pp. 317–367, 1998.

[38] Li, Y. and Chen, Y., "When is a Mittag-Leffler function a Nussbaum function?," *Automatica*, vol. 45, no. 8, pp. 1957–1959, 2009.

[39] Li, Y., Chen, Y., and Cao, Y., "Fractional order universal adaptive stabilization," *Third IFAC Workshop on Fractional Differentiation and its Applications, Ankara, Turkey,* Nov. 2008.

[40] Linden, D. and Reddy, T., *Handbook of Batteries.* McGraw-Hill, 3rd ed., 2002.

[41] Liu, L., *Integrated system identification and state-of-charge estimation of battery systems.* PhD thesis, Wayne State University, January 2011. Paper AAI1503103.

[42] Loparo, K. and Blankenship, G., "Estimating the domain of attraction of nonlinear feedback systems," *IEEE Transactions on Automatic Control*, vol. 23, pp. 602–608, Aug. 1978.

[43] Malisoff, M., Mazenc, F., and Zhang, F., "Input-to-state stability for curve tracking control: A constructive approach," in *Proceedings of the American Control Conference*, (Baltimore, MD), pp. 1984–1989, 2011.

[44] Malisoff, M., Mazenc, F., and Zhang, F., "Stability and robustness analysis for curve tracking control using input-to-state stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1320–1326, 2011.

[45] Malisoff, M. and Mazenc, F., *Constructions of Strict Lyapunov Functions.* Springer, 2009.

[46] Márquez, H. J., *Nonlinear Control Systems: Analysis and Design.* Wiley-Interscience, 1 ed., 2003.

[47] Michel, A., Miller, R., and Nam, B., "Stability analysis of interconnected systems using computer generated Lyapunov functions," *IEEE Transactions on Circuits and Systems*, vol. 29, pp. 431–440, Jul. 1982.

[48] Moiola, J. L. and Berns, D. W., "On the detection of period doubling bifurcations in nonlinear feedback systems," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 465–468, 1997.

[49] Mukhopadhyay, S., Li, Y., and Chen, Y., "Experimental studies of a fractional order universal adaptive stabilizer," in *Proceedings of International Conference on Mechatronic and Embedded Systems and Applications*, pp. 591–596, Oct. 2008.

[50] Mukhopadhyay, S., Wang, C., Bradshaw, S., Bazie, V., Maxon, S., Hicks, L., Patterson, M., and Zhang, F., "Controller performance of marine robots in reminiscent oil surveys," in *Proceedinggs of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1766–1771, 2012.

[51] Mukhopadhyay, S. and Zhang, F., "Adaptive detection of terminal voltage collapses for Li-ion batteries," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 4799–4804, 2012.

[52] Mukhopadhyay, S. and Zhang, F., "A high-gain adaptive observer for detecting Li-ion battery terminal voltage collapse," *Automatica*, vol. 50, no. 3, pp. 896–902, 2014.

[53] Mukhopadhyay, S. and Zhang, F., "A path planning approach to compute the smallest robust forward invariant sets," in *Proceedings of the American Control Conference*, p. accepted, 2014.

[54] Nussbaum, R. D., "Some remarks on a conjecture in parameter adaptive control," *Systems & Control Letters*, vol. 3, no. 5, pp. 243–246, 1983.

[55] Pai, M., Mohan, M., and Rao, J., "Power system transient stability: Regions using Popov's method," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 788–794, May. 1970.

[56] Panik, M. J., *Fundamentals of Convex Analysis.* Springer, 1993.

[57] Parrilo, P. A., "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, pp. 293–320, 2003.

[58] Pedram, M. and Wu, Q., "Design considerations for battery-powered electronics," in *Proceedings of the 36th Design Automation Conference*, pp. 861–866, 1999.

[59] Plett, G. L., "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background," *Journal of Power Sources*, vol. 134, no. 2, pp. 252–261, 2004.

[60] Plett, G. L., "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification," *Journal of Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.

[61] Plett, G. L., "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation," *Journal of Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.

[62] Pop, V., Bergveld, H., Danilov, D., and Regtien, P., *Battery Management Systems: Accurate State-of-Charge Indication for Battery Powered Applications.* Springer, 2008.

[63] Prajna, S., Papachristodoulou, A., and Wu, F., "Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach," in *5^{th} Asian Control Conference*, vol. 1, pp. 157–165, Jul. 2004.

[64] Prätzel-Wolters, D., H, O. D., and Achim, I., "Robust adaptive stabilization by high gain feedback and switching," *International Journal of Control*, vol. 49, no. 6, pp. 1861–1868, 1989.

[65] Rakovic, S., Grieder, P., Kvasnica, M., Mayne, D., and Morari, M., "Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 2, pp. 1418–1423, Dec 2004.

[66] Rakovic, S., Kerrigan, E., Kouramas, K., and Mayne, D., "Invariant approximations of the minimal robust positively invariant set," *IEEE Transactions on Automatic Control*, vol. 50, pp. 406–410, March 2005.

[67] Rao, R., Vrudhula, S., and Rakhmatov, D., "Battery modeling for energy-aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003.

[68] Schweighofer, B., Raab, K., and Brasseur, G., "Modeling of high power automotive batteries by the use of an automated test system," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 4, pp. 1087–1091, 2003.

[69] Smyshlyaev, A., Krstic, M., Chaturvedi, N., Ahmed, J., and Kojic, A., "PDE model for thermal dynamics of a large Li-ion battery pack," in *Proceedings of the American Control Conference*, pp. 959–964, Jul 2011.

[70] Strogatz, S. H., *Nonlinear Dynamics and Chaos.* Perseus Press, 1994.

[71] Tibken, B. and Fan, Y., "Computing the domain of attraction for polynomial systems via BMI optimization method," in *Proceedings of the American Control Conference*, pp. 117–122, Jun. 2006.

[72] Topcu, U. and Packard, A., "Local stability analysis for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1042–1047, May 2009.

[73] Topcu, U., Packard, A., Seiler, P., and Balas, G., "Robust region-of-attraction estimation," *IEEE Transactions on Automatic Control*, vol. 55, pp. 137–142, Jan. 2010.

[74] Vannelli, A. and Vidyasagar, M., "Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems," *Automatica*, vol. 21, no. 1, pp. 69–80, 1985.

[75] Walker, J. A. and McClamroch, N. H., "Finite regions of attraction for the problem of Luré," *International Journal of Control*, vol. 6, no. 4, pp. 331–336, 1967.

[76] Weissenberger, S., "Application of results from the absolute stability problem to the computation of finite stability domains," *IEEE Transactions on Automatic Control*, vol. 13, pp. 124–125, Feb. 1968.

[77] Willems, J., "Direct method for transient stability studies in power system analysis," *IEEE Transactions on Automatic Control*, vol. 16, pp. 332–341, Aug. 1971.

[78] Willems, J. and Byrnes, G., "Global adaptive stabilization in the absence of information on the sign of the high frequency gain," in *Analysis and Optimization of Systems* (Bensoussan, A. and Lions, J., eds.), vol. 62 of *Lecture Notes in Control and Information Sciences*, pp. 49–57, Springer Berlin Heidelberg, 1984.

[79] Willems, J., "Improved Lyapunov function for transient power-system stability," *Proceedings of the Institution of Electrical Engineers*, vol. 115, pp. 1315–1317, Sep. 1968.

[80] Wu, W. and Zhang, F., "Robust cooperative exploration with a switching strategy," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 828–839, 2012.

[81] Yaghoobi, H., Hassouneh, M. A., and Abed, E. H., "Detection of impending bifurcation using a near-resonant probe signal," in *Proceedings of the American Control Conference*, pp. 2285–2291, 2001.

[82] Young, P. M., Newlin, M. P., and Doyle, J. C., "μ analysis with real parametric uncertainty," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 1251–1256, 1991.

[83] Zhang, F., Justh, E., and Krishnaprasad, P., "Boundary following using gyroscopic control," in *Proceedings of the 43^{rd} IEEE Conference on Decision and Control*, vol. 5, pp. 5204–5209, Dec. 2004.

[84] Zhang, F., Shi, Z., and Mukhopadhyay, S., "Robustness analysis of battery supported cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 3, pp. 69:1–69:27, 2012.

[85] Zhang, F. and Leonard, N. E., "Coordinated patterns of unit speed particles on a closed curve," *Systems & Control Letters*, vol. 56, no. 6, pp. 397–407, 2007.

[86] Zhang, S. and Chen, M., "A new Razumikhin theorem for delay difference equations," *Computers and Mathematics with Applications*, vol. 36, pp. 10–12, 1998.

[87] Zubov, V. I., *Methods of A.M. Lyapunov and their application.* Groningen, P. Noordhoff, 1964.

VITA

At the time of writing this thesis, Shayok Mukhopadhyay is a Ph.D. candidate in electrical engineering at the Georgia Institute of Technology, Atlanta, Georgia, USA. He received his B.E. in electrical engineering from the College of Engineering Pune (C.O.E.P), at the University of Pune, India in 2006. He was ranked third in the University of Pune at the B.E. degree examination, in 2006. He received his M.Sc. in electrical engineering (majoring in systems and control) from Utah State University, Logan, Utah, USA in 2009. His research interests include nonlinear systems and computational methods in general. He also likes computer programming. More information about him can be found on his website at http://sites.google.com/site/shayok.