



Theses and Dissertations

2005-11-10

Verification of Digital Controller Verifications

Xuan Wang

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Computer Sciences Commons](#)

BYU ScholarsArchive Citation

Wang, Xuan, "Verification of Digital Controller Verifications" (2005). *Theses and Dissertations*. 681.
<https://scholarsarchive.byu.edu/etd/681>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

VERIFICATION OF DIGITAL CONTROLLER IMPLEMENTATIONS

by

Xuan Wang

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

December 2005

Copyright © 2005 Xuan Wang

All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Xuan Wang

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Mike Jones, Chair

Date

Sean Warnick

Date

Yiu-Kai Dennis Ng

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Xuan Wang in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Mike Jones
Chair, Graduate Committee

Accepted for the Department

Parris Egbert
Graduate Coordinator

Accepted for the College

G. Rex Bryce, Associate Dean
College of Physical and Mathematical Sciences

ABSTRACT

VERIFICATION OF DIGITAL CONTROLLER IMPLEMENTATIONS

Xuan Wang

Department of Computer Science

Master of Science

This thesis presents an analysis framework to verify the stability property of a closed-loop control system with a software controller implementation. The usual approach to verifying stability for software uses experiments which are costly and can be dangerous. More recently, mathematical models of software have been proposed which can be used to reason about the correctness of controllers. However, these mathematical models ignore computational details that may be important in verification. We propose a method to determine the instability of a closed-loop system with a software controller implementation under l^2 inputs using simulation. This method avoids the cost of experimentation and the loss of precision inherent in mathematical modeling. The method uses small gain theorem compute a lower bound on the 2-induced norm of the uncertainty in the software implementation; if the lower bound is greater than $\frac{1}{\|G\|_{2-ind}}$, where G is the feedback system consisting of the mathematical model of the plant and the mathematical model of the controller, the closed-loop

system is unsafe in certain sense. The resulting method can not determine if the closed-loop system is stable, but can only suggest instability.

ACKNOWLEDGMENTS

I am particularly thankful to my advisor Dr. Mike Jones for funding my education, patiently guiding me throughout my research and, consideration. I would like to thank Dr. Sean Warnick for his efforts and contributions to my thesis work. I would like to thank Dr. Tom Sederberg for the research guidance and financial aid during my education in BYU in fall 2000 and winter 2001. I would like to thank Dr. Dennis Ng and Dr. Eric Mercer for their time and efforts on my thesis. I would like to thank all members of the Verification and Validation lab for their help; they made my experience in VV lab a very enjoyable one.

I would like to thank my friends, Rina, Cui, Zonghui, Dan Su and Tonga for their help and encouragement during my education in BYU.

This thesis work is done while I am away with my family. I am grateful for the love, support and understanding that my whole family provide to me. Without their support, education at BYU would be an impossible mission for me. My special thanks are given to my husband, Yue, my parents, my parents-in-law, my sisters, my brother and my cutest son, Longlong, and my adorable daughter, Yingying.

TABLE OF CONTENTS

Section	Page
1 Introduction	1
1.1 Digital Control Systems	1
1.2 Robustness in Controller Verification	5
1.3 Thesis Statement	9
2 Related Work	11
2.1 Hybrid Systems	11
2.2 Controller Uncertainty	15
2.3 Modeling of Software for Verification	17
3 Verification of Digital Controller Implementations	19
3.1 Problem Definition	19
3.2 Small Gain Theorem	21
3.2.1 Signal Norms	21
3.2.2 System Norms	24
3.2.3 Small Gain Theorem	25
3.3 Verification of Digital Controller Implementation	26
3.3.1 Computation of Linear Operator Norms	28
4 Computation of 2-induced Norm of Controller Uncertainty	37
4.1 Computing the 2-induced Norm of Software Controller Uncertainty	38
4.2 Algorithm to Compute A Lower Bound of the Software Controller Uncertainty	44
5 Conclusions and Future Work	51
5.1 Future Work	52

LIST OF TABLES

Table

Page

LIST OF FIGURES

Figure	Page
1.1 A typical feedback control system	2
1.2 Control diagram of a digital controller	4
1.3 Feedback connection	7
1.4 A feedback system with plant uncertainty	8
1.5 A feedback system with controller uncertainty	9
2.1 Finite automata model of hybrid systems as used in [5].	12
2.2 Response of a stable system with step input	14
2.3 Response of an unstable system with step input	15
2.4 Decomposition of a hybrid system used to show input-to-state stability in [11]	16
3.1 A closed-loop system without controller uncertainty	20
3.2 A closed-loop system with controller uncertainty	21
3.3 Norms over R^2	23
3.4 An interconnected system of an ideal control system and controller uncertainty	27
4.1 A nonlinear operator with truncated input	38
4.2 A nonlinear operator with truncated input and truncated output	41
4.3 Component and variables needed to compute the lower bound of $\ \Delta_k\ $.	45
4.4 Algorithm to compute a lower bound of $\ \Delta_C\ $	46
4.5 State trajectory expansion due to non-determinism.	48

Chapter 1

Introduction

This thesis establishes a framework for determining if a software implementation of a digital controller may destabilize a closed-loop control system. In this chapter, we present the problem to be solved and give some background knowledge.

1.1 Digital Control Systems

Digital control systems are pervasive in industry and in our everyday lives. They can be found in a wide variety of applications including microwaves, automobiles, machines and airplanes. Control systems are often feedback closed-loop systems, where signals produced by the plants of particular interest are compared to their references, and the control actions are computed based on the differences. Figure 1.1 shows a feedback closed-loop system.

The goal in a digital control problem is to design and implement a computer-based controller according to the dynamic behavior of the plant and the required control objectives. Generally, the control objectives of a control system are to decrease the output errors, increase the speed of the system response and increase the system bandwidth¹. Among all control objectives, stability is the most important because it

¹System bandwidth is a measure of the frequency range over which the closed-loop magnitude of the response to a unit amplitude input exceeds $\frac{1}{\sqrt{2}}$.

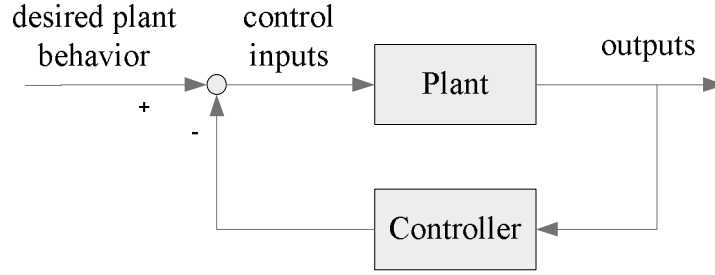


Figure 1.1: A typical feedback control system

determines the safety of a system. A system is stable if every bounded input produces a bounded output [1].

This thesis work considers the systems that can be represented with finite dimensional, linear and time-invariant system models. We choose this class of systems because it includes the simplest models which cover fundamental issues in control systems and a linear ordinary differential equation with constant coefficients is the most common type of control law. There are many ways to design and implement a digital controller. Here we illustrate the control procedure on which this thesis will be based. The first step in the design procedure is to model the physical plant by applying natural laws to their ideal building blocks, such as masses, beams, electrons and so on. Most often, the model is composed of nonlinear differential equations. The method can then be reduced by linearization to convert the nonlinear differential equations into linear differential equations. Equations 1.1 and 1.2 are the state-space representation of a linearized model, which consists of a set of first-order differential equations in terms of state variables.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.1)$$

$$y(t) = Cx(t) + Du(t) \quad (1.2)$$

where

- $t \in R$ is time,
- $x(t) \in R^n$ is the state vector at time t ,
- $\dot{x}(t) = dx(t)/dt$ is the first derivative of x ,
- $u(t) \in R^m$ is the input vector at time t ,
- $y(t) \in R^p$ is the output vector at time t ,
- $A \in R^{n \times n}$ is the dynamics matrix,
- $B \in R^{n \times m}$ is the input matrix,
- $C \in R^{p \times n}$ is the output matrix,
- $D \in R^{p \times m}$ is the feedthrough matrix.

Digital control systems operate on discrete signals rather than continuous signals. For discrete control systems, the continuous plant model needs to be converted to discrete model using zero-order hold operation. The resulting discrete linear difference equations can be written as

$$x(k+1) = A_k x(k) + B_k u(k) \quad (1.3)$$

$$y(k) = C_k x(k) + D_k u(k) \quad (1.4)$$

where

- $k \in Z = \{0, 1, 2, \dots\}$ is the number of sampling,
- x, u, y are vector sequences, which represents state, input and output,
- $A_k \in R^{n \times n}, B_k \in R^{n \times m}, C_k \in R^{p \times n}, D_k \in R^{p \times m}$ are real valued matrices.

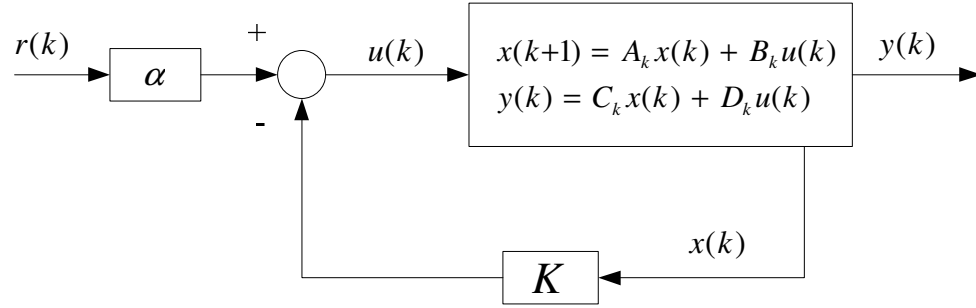


Figure 1.2: Control diagram of a digital controller

The next step is to design a control law, which is a mathematical model of a controller, such that the resulting feedback control system achieves the control objectives. Equation 1.5 gives a typical full state feedback controller.

$$u(k) = \alpha r(k) - Kx(k) \quad (1.5)$$

where

- $\alpha \in R$ is a feedforward scalar factor for the reference,
- $r(k) \in R^m$ is the reference input at sample k ,
- $K \in R^{m \times n}$ is the control matrix.

Figure 1.2 shows the control diagram of the closed-loop system. The system, so far, is amenable to analysis by well-defined mathematic theories which can be used to prove that the digital control system is controllable and stable. In the system described by 1.3, 1.4 and control input 1.5, the eigenvalues of matrix $(A_k - B_k \times K)$ are the poles of the closed-loop system. The discrete-time system is BIBO (Bounded Input and Bounded Output) stable if the eigenvalues of $(A_k - B_k \times K)$ are inside the unit circle.

After the control law is designed, it is implemented in software so that the control program can be run on the target machine to fulfill the control objectives. The controller implementation in software introduces extra requirements, and errors, such as finite number representations, finite resolution in numerical computation, programmer errors, and so on.

For such a controller implementation, the verification question is: “can the software be run on the target machine to achieve system stability?” The common method for answering this question is to do a set of experiments to verify the stability of the software. The problem with this approach is that doing experiments to verify the program on target machine is costly and may be dangerous, for example, when verifying a flight controller. In this thesis, simulation is used to determine the instability of the software controller implementation instead of experiments with real hardware. Simulation is much cheaper and not dangerous.

1.2 Robustness in Controller Verification

In reality, a digital control law is usually realized in a digital processor with constraints such as finite number representations and finite precision, and with unknown programmer errors. Even though the ideal control law is designed to be stable, an implementation under these constraints and errors may decrease the desired performance of the closed-loop system or even make the system to become unstable. Given a controller implementation, which is derived from an ideal control law, how can we verify the closed-loop stability?

The method we will use to verify a digital controller implementation is inspired by methods used to handle plant uncertainties in control problems. Before discussing how models of uncertainty can help in reasoning about software implementations, let us first study how plant uncertainties are dealt with in control problems. As described in section 1.1, most control designs start by modeling the plant that is to be controlled. In reality, no mathematical model can exactly represent a physical

system, and, sometimes, in order to make the problem more convenient, a plant model is simplified. For example, a nonlinear model might be linearized to create a new problem which is similar to the original problem in order to facilitate control design. These modeling errors may adversely affect the performance and stability of a control system. Because of this, robust control, which deals with various models of plant uncertainties, is studied. The basic idea in robust control is to model the plant as a set P , and a controller is designed such that it provides robust stability for every plant in the set P .

The small gain theorem is extensively used in robust control problems with plant uncertainties [2][3][4]. This theorem gives a sufficient condition for stability in the kinds of feedback systems shown in Figure 1.3. The closed-loop control system shown in Figure 1.2 is an example of this kind of feedback systems. In this system, both subsystems G_1 and G_2 can be arbitrary nonlinear time-varying systems, which map signals from vector spaces with compatible dimensions. These vector spaces are equipped with p -norms. The inputs and outputs of the subsystems satisfy the following relations.

$$y_1 = G_2 y_2 + u_1; \quad (1.6)$$

$$y_2 = G_1 y_1 + u_2; \quad (1.7)$$

where

- $u_1 \in R^n$ is the external input to G_1 ,
- $y_1 \in R^n$ is the control input to G_1 ,
- $u_2 \in R^m$ is the external input to G_2 ,
- $y_2 \in R^m$ is the control input to G_2 .

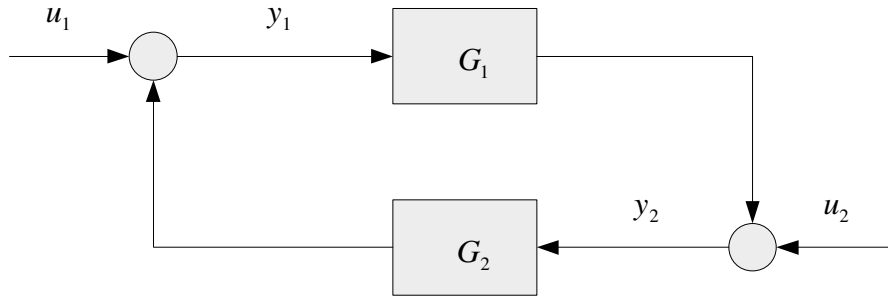


Figure 1.3: Feedback connection

Theorem 1.2.1 (Small Gain Theorem [2]) *Let $G_1 : l_p^n \rightarrow l_p^m$ and $G_2 : l_p^m \rightarrow l_p^n$ be two stable operators of bounded l_p -gain and assume that the closed loop system is well posed. Then the closed loop system is stable if $\|G_1\| \|G_2\| < 1$.*

The detailed introduction of small gain theorem together with signal norms, system norms and their computations are presented in chapter 3. Intuitively, the small gain theorem says that if both G_1 and G_2 are stable and if the multiplication of the induced norm of G_1 by the induced norm of G_2 is less than 1, then the interconnected system is stable.

The Small Gain Theorem provides a test for robust stability. In the presence of plant uncertainty, a feedback system \hat{G} can be viewed as a nominal feedback system G , which is a system that includes the controller and nominal plant without plant uncertainty, and plant uncertainty Δ_p as shown in figure 1.4. Assume the nominal feedback system G is stable under controller C and Δ_p is also stable, then based on small gain theorem the controller C stabilizes the system \hat{G} if

$$\|G\| \|\Delta_p\| < 1;$$

In the case of controller uncertainty, the small gain theorem can also be used to

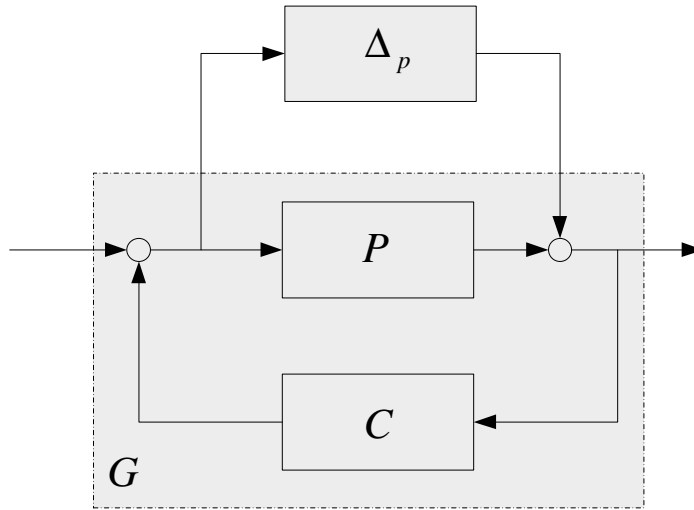


Figure 1.4: A feedback system with plant uncertainty

verify system stability. The feedback system \hat{M} with a software controller implementation can be viewed as a nominal feedback system M , which is a system without controller uncertainty, together with controller uncertainty Δ_c as shown in figure 1.5. Assume that both the nominal feedback system M and the uncertainty Δ_c are stable, then by checking

$$\|M\|\|\Delta_c\| < 1$$

we can verify if the feedback system \hat{M} is stable.

In this manner, the small gain theorem is used to show that a set of controllers can stabilize a given plant. This is important because the software implementation of a controller will never exactly match the mathematical model of a controller in much the same way that a mathematical model of a plant will never match the actual plant. In this thesis, significant progress is made toward modeling uncertainty in the

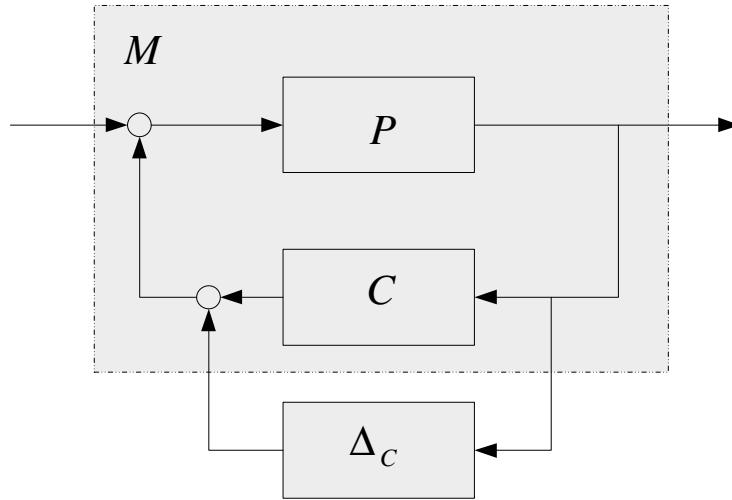


Figure 1.5: A feedback system with controller uncertainty

controller implementation. Further work is needed to derive a theory that supports uncertainty in both the plant and the controller.

1.3 Thesis Statement

The small gain theorem can be used to determine if a software implementation of a digital controller may destabilize the closed-loop system by giving a bound on the 2-induced norm of the uncertainty allowed in the software implementation.

Chapter 2

Related Work

This work focuses on the verification of digital controller software. It relates to hybrid system verification, controller uncertainty and computation for software verification. This chapter presents the research work done in these three areas.

2.1 Hybrid Systems

A hybrid system consists of a collection of digital programs and a physical environment in which the digital programs interact with each other and with the environment. Hybrid system design combines the fields of engineering control theory and computer science verification. Digital controller verification, as studied in this thesis, lies in the area of hybrid systems design. The modeling and verification of hybrid systems, which exhibit very challenging problems, have received much research attention in the past decade.

Alur *et al.* model hybrid systems as finite automata with continuous variables that evolve continuously with respect to time based on dynamic laws [5]. The model consists of six components: locations, variables, synchronization labels, transitions, activities and invariant conditions; as shown in Figure 2.1. The hybrid system in Figure 2.1 consists of two locations, *Location1* and *Location2*. Each location has a set of activities, such as *Activity1* in *Location1*. At each location there are

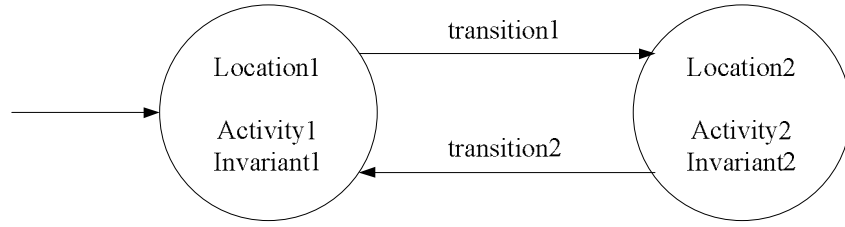


Figure 2.1: Finite automata model of hybrid systems as used in [5].

invariant conditions. Invariant conditions are predicates over variables that must be satisfied in that location, such as *Invariant1* in *Location1*. Transitions are jump conditions under which the system switches from one location to another location. For example, *transition1* is the transition condition from *Location1* to *Location2*. Alur *et al.* prove that the reachability problem for simple multirate timed systems is decidable, while the reachability problem for *2-rate* timed systems is undecidable, where *2-rate* timed systems are systems with two distinct clock rates. They present a methodology for verifying linear hybrid systems using forward analysis, backward analysis and approximate analysis.

A symbolic model checker, HyTech, is implemented for verifying linear hybrid systems based on the finite automata model [6]. HyTech computes the reachable state set of a linear hybrid automaton, then checks if the state assertion is false for this set. HyTech requires that, given a state assertion, all the reachable states must converge to a fixed point within a finite number of transitions and flows. Because of these requirements, HyTech can only verify a restricted classes of linear hybrid automata. This class includes, for example, timed automata.

Because checking reachability is undecidable for general hybrid systems, many researchers have focused on finding decidable subclasses of hybrid systems. Puri and Varaiya consider a subclass of hybrid systems with constant rectangular differential

inclusions [7]. They show that verification for this class of hybrid systems is decidable. Alur *et al.* consider classes of hybrid systems, either with simple continuous dynamics, such as timed automata, multirate automata and rectangular automata, or with simple discrete dynamics, such as order-minimal hybrid systems [8]. They abstract these subclasses of hybrid systems to a class of purely discrete systems which preserve all properties definable in temporal logic. Based on this result, they present a unified way to collectively define the boundary between decidability and undecidability for hybrid systems.

Tomlin *et al.* develop a reachability computation algorithm based on level set techniques and viscosity solutions [9]. It employs an implicit surface function representation of the reachable set and, computes its evolution using constrained level set methods and discrete mappings through transition functions. This algorithm increases the ability to compute reachable sets of hybrid systems. Therefore it is able to represent, analyze and verify nonlinear hybrid systems.

An explicit discrete model checker, called Mur φ , extended with a long double floating point type is used to verify a Turbogas Control System [10]. The differential equations of the control system are discretized with a certain sampling time and an uncontrollable disturbance and its derivatives are restricted by constant bounds. Reachable states are explicitly computed in Mur φ based on the discrete time models. This method neglects reachable states between sample points, thus cannot guarantee whether the control system satisfies requirements between sample points. This problem can be solved by assuming that the system dynamics are piecewise continuous between sampling points and that every pair of contiguous sampling points contain the maximum and minimum values between them.

All of these works in hybrid system verification implicitly assume that all hybrid systems are stable. Verification is done by computing reachable states based on the ideal models of the hybrid systems using various methods and then checking if unsafe

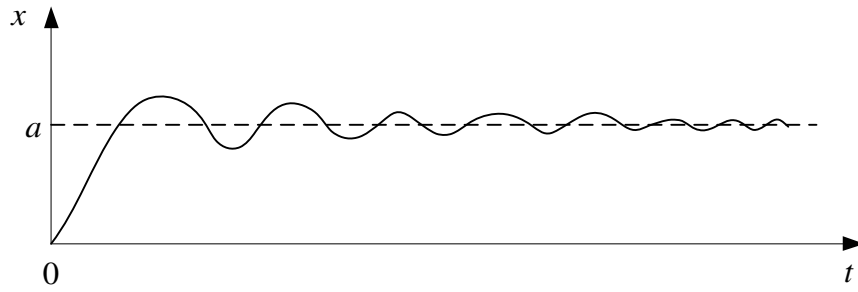


Figure 2.2: Response of a stable system with step input

properties are satisfied in these reachable states. The stability of hybrid systems is an important property in hybrid automata. In the hybrid automata model of hybrid systems, the activities at each location are either systems natural responses or are achieved by control devices. If a system is stable, then the system will follow the trajectory as described in the activities. If a system is unstable, the system response goes to infinity as time goes to infinity. An unstable system does not follow the trajectories as described in the activities and it may reach every state in the state space. In other words, every unstable system is unsafe. For example, if an activity of a system in one location is to follow a step input a , Figure 2.2 shows a response of a stable system, the reachable state is $x = a$. Figure 2.3 shows a response of an unstable system, the reachable x could be any value, even unsafe value.

Nesic *et al* briefly introduce the concept of input-to-state stability (ISS). In an ISS system, bounded inputs and inputs converging to zero produce states which are also bounded and converging to zero. The authors propose to use the ISS small-gain theorem to analyze the stability of hybrid systems [11]. The authors view a hybrid system as a feedback closed-loop system consisting of two subsystems and each subsystem in the decomposition is a continuous, discrete, or hybrid system as shown in figure 2.4. Then the small gain theorem is applied to reduce the problem of

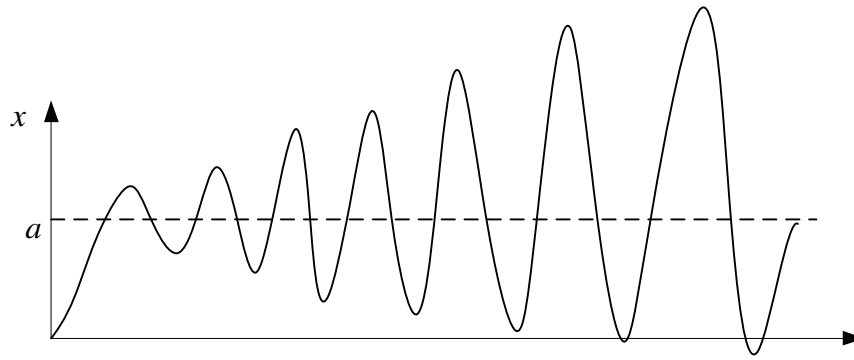


Figure 2.3: Response of an unstable system with step input

verifying ISS of the hybrid system to be a problem of verifying ISS of its subsystems and checking a condition of the subsystems' gains. The authors provide a small-gain analysis framework for hybrid systems, however, the analysis is conducted only with mathematical models of hybrid systems and no concrete implementation is considered.

2.2 Controller Uncertainty

In control theory, plant uncertainty has long been studied [2] [3] [12]. It is known that no mathematical system can exactly model a physical system and these modeling errors may lead to the instability of a control system. Robust stability is a kind of stability that allows the presence of modeling errors. The basic idea is that instead of modeling the plant as a single fixed model, the plant is modeled as a set. This set can be structured or less structured, that is, plant uncertainties are modeled as a finite number of uncertain parameters (structured), or the frequency response of the plant for every frequency lies in a set in the complex plane (less structured). In robust control, the small gain theorem is used to design and verify a controller such that the controller provides robust stability to every closed-loop system in the plant model set. In other words, the controller provides internal stability for every plant in the plant model set.

In robust control, controller uncertainty had long been ignored as it is noted

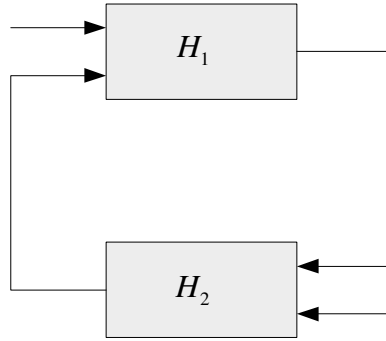


Figure 2.4: Decomposition of a hybrid system used to show input-to-state stability in [11]

that plant uncertainty is the most significant source of uncertainties in control systems, while controllers are generally implemented with high precision hardware and software. Keel and Bhattacharyya raise the problem that a controller may not be implemented exactly [13]. They also point out that controller uncertainty exists due to imprecision inherent in A/D and D/A conversion, finite word length, finite resolution of measuring instruments and roundoff errors in numerical computations, and that useful design procedures require a controller to have sufficient room for readjustment of its coefficients. Through examples, they show that optimum and robust controllers, designed by using H_2 , H_∞ , l^1 and μ formulations, can be extremely “fragile”, that is, they can result in unstable closed-loop control systems even if very small perturbations are applied to the coefficients of the designed controllers.

Whidborne *et al.* present a pole sensitivity approach to reduce controller fragility through a state space parameterization of the controller [14]. This method is based on a weighted norm of the closed-loop pole/eigenvalue sensitivities to controller parameter perturbations. They provide conditions for the optimal state space realization of the controller and a numerical method to obtain the solution is introduced. Wu *et*

al. apply this pole sensitivity approach to investigate stability of discrete time control systems, where the digital controllers are implemented with finite word length (FWL) [15]. The authors derive an improved stability measure which estimates the closed-loop stability robustness of an FWL implemented controller more accurately.

Controller uncertainty is considered when designing an optimal mathematically modeled controller so as to reduce the closed-loop control system's sensitivity to controller parameter perturbations. No software implementation of the controller is studied to check if given this software implementation, the closed-loop control system is stable or not. In this thesis, uncertainties of the software controller implementations are studied. we present a method which applies the small gain theorem to determine the instability of a closed-loop control system by computing the norm of the controller uncertainty.

2.3 Modeling of Software for Verification

Reasoning about the stability of a closed-loop control system with software controller implementation requires either modeling the software implementation and then computing the norm of its uncertainty based on the mathematical model, or using simulation to compute the norm of the uncertainty directly from the software implementation.

Roosbehani *et al.* present a framework for modeling and analysing real-time safety-critical software [16]. They model software as a dynamical system and then convert verification of software properties, such as bounded-ness of variables and termination of the program in finite time, to an optimization-based search for system invariants. This modeling approach ignores software implementation details, such as discretization and scheduling etc, which is good because verification is focused on proving the desired properties of the software and this method is scalable to large-size computer programs. The problem of this approach is that the run-time errors caused by implementation details can not be detected. In contrast, in this thesis

work, we verify the instability property of a closed-loop control system with software controller implementation using simulation. Implementation details are reserved in the simulation, therefore, run-time errors caused by these details can be detected. But, can not guarantee absence of errors.

Chapter 3

Verification of Digital Controller Implementations

In this chapter, the small gain theorem is applied to the verification of a digital controller implementation. We start by defining the verification problem, then give a detailed explanation of the small gain theorem. After that, the method which uses the small gain theorem to verify the stability of a closed-loop system built from a digital controller implementation is presented.

3.1 Problem Definition

Consider a discrete-time closed-loop control system which consists of a linear, finite-dimensional, time-invariant plant P and a digital controller C as shown in Figure 3.1. The plant is assumed to be strictly proper¹ with state-space representation as shown in equations 1.3 and 1.4, which are,

$$\begin{aligned}x(k+1) &= A_k x(k) + B_k u(k) \\ y(k) &= C_k x(k) + D_k u(k)\end{aligned}$$

¹In control theory, a strictly proper system is a system in which the numerator of the transfer function has lower degree than the denominator. It implies that the system's outputs approach zero as the input frequency approaches infinity.

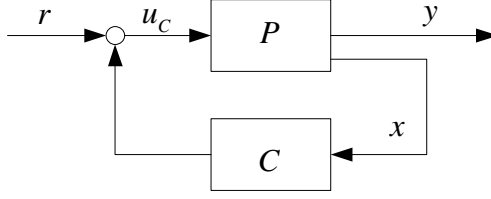


Figure 3.1: A closed-loop system without controller uncertainty

and the digital controller is represented as in equation 1.5 which is,

$$u(k) = \alpha r(k) - Kx(k)$$

The stability of this closed-loop system depends on the eigenvalues of the matrix

$$\widehat{A}_k = A_k - B_k * K$$

If we assume that the digital controller was correctly designed to stabilize the closed-loop system, then the eigenvalues of \widehat{A}_k satisfy

$$|\lambda_i(\widehat{A}_k)| < 1, \quad \forall i \in \{1, 2, \dots, n\}$$

Even though the digital controller C stabilizes the closed-loop system, its realization C_r , in its software implementation, which includes other factors, such as finite word length and inevitable programmer errors, may destabilize the system. Now given an controller realization C_r which is an implementation of the digital controller C to be run on a target processor, we want to verify that the closed-loop system which consists of the plant P and the controller implementation C_r on the target processor is stable.

The basic idea of our method is that we model C_r to be a nominal controller C with some uncertainty Δ_c as shown in Figure 3.2. By applying the small gain theorem, we can compute the maximum controller uncertainty Δ such that any controller, if

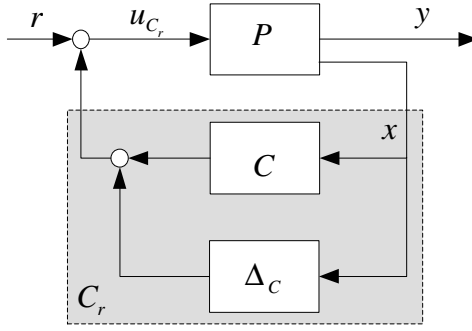


Figure 3.2: A closed-loop system with controller uncertainty

its uncertainty is within the uncertainty set Δ , can stabilize the closed-loop system. Then, by checking whether or not the uncertainty Δ_c of the controller implementation C_r sits inside the maximum uncertainty set Δ , we can verify that the implementation C_r stabilizes the closed-loop system.

3.2 Small Gain Theorem

The small gain theorem provides a method to quantify or measure “robust stability”, which is the stability under uncertainty. The stability measurement is expressed in system norms, or operator norms. In order to understand small gain theorem, we start by introducing signal norms and system norms.

3.2.1 Signal Norms

A signal norm measures the “size” of a signal. There are many different norm definitions, we begin with the general norm.

Definition 3.2.1 (General Norm) *Let V be a linear space. A norm on V is a bounded function $\| * \|$ mapping V into R^+ which satisfies the following conditions.*

1. $\|x\| \geq 0$
2. $\|x\| = 0 \leftrightarrow x(t) = 0, \forall t$

$$3. \|\alpha x\| = |\alpha|\|x\|, \forall \alpha \in \mathbb{R}$$

$$4. \|x + y\| \leq \|x\| + \|y\|$$

where x and y are any signals on V .

Definition 3.2.2 (Normed Linear Space) A normed linear space is a linear space equipped with a norm.

The p -norm of the finite-dimensional vector space $x \in \mathbb{R}^n$ is defined as

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

for any $1 \leq p \leq \infty$.

The p -norm of a vector provides different measures of its length. The most commonly used p -norms are the 1-norm, 2-norm and ∞ -norm. They are

- **1-norm** $\|x\|_1 = \sum_{i=1}^n |x_i|$,
- **2-norm** $\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}$,
- **∞ -norm** $\|x\|_\infty = \max_i |x_i|$.

To illustrate the meaning of these norms, let us consider a vector space with $n = 2$ and let u be a point with coordinates (x_u, y_u) as shown in Figure 3.3. Then the 1-norm of u is $|x_u| + |y_u|$, which is the sum of the absolute values of its coordinates. The 2-norm is $\sqrt{|x_u|^2 + |y_u|^2}$, which is the Euclidean distance from the origin to u . The ∞ -norm is $\max(|x_u|, |y_u|)$. In this example, we see that $\|u\|_1 \geq \|u\|_2 \geq \|u\|_\infty$. In general, this is true in finite dimensional spaces.

Let us now extend the finite-dimensional space to an infinite sequence. Let Z denote the set of all integers and $l_p^n(Z)$ denote the space of all vector-valued real sequences with dimension n and with bounded p -norm defined as

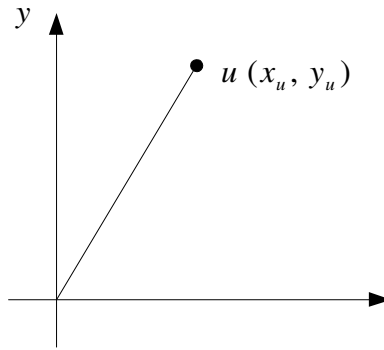


Figure 3.3: Norms over R^2

$$\|x\|_p = \left(\sum_{k=0}^{\infty} |x(k)|^p \right)^{\frac{1}{p}}$$

for any $1 \leq p \leq \infty$.

When $p = 1$, $p = 2$, $p = \infty$, the respective norms are,

- **1-norm** $\|x\|_1 = \sum_{k=0}^{\infty} |x(k)|$,
- **2-norm** $\|x\|_2 = \left(\sum_{k=0}^{\infty} |x(k)|^2 \right)^{\frac{1}{2}}$,
- **∞ -norm** $\|x\|_{\infty} = \sup_{k \geq 0} |x(k)|$.

If $p = 2$, the norm $\|x\|_2$ is the amount of energy contained in the signal. If $p = \infty$, the norm $\|x\|_{\infty}$ is the maximum magnitude that the signal attains over all time. In general, every finite energy signal is also a finite magnitude signal, but not vice versa.

Lebesgue integrable functions² will be needed to define a class of operators used later. Therefore, we introduce the p -norm of Lebesgue integral functions here. Let $\mathcal{L}_p(B)$ denote the space of all Lebesgue integrable functions on closed set B with

²In mathematics, the integral of a function of one real variable can be regarded as the area of a plane region bounded by the graph of that function. Lebesgue integration is a mathematical theory that extends the integral to a very large class of functions.

bounded p -norm defined as

$$\|x\|_p = \left(\int_B |x(t)|^p dt \right)^{\frac{1}{p}}$$

for any $1 \leq p \leq \infty$. Respectively, the 1-norm, 2-norm and ∞ -norm of this type of signals are

- **1-norm** $\|x\|_1 = \int_B |x(t)| dt$
- **2-norm** $\|x\|_2 = \left(\int_B |x(t)|^2 dt \right)^{\frac{1}{2}}$
- **∞ -norm** $\|x\|_\infty = \sup_{t \in B} |x(t)|$

3.2.2 System Norms

If signals can be defined to be in some suitable signal space, a system which maps input signals to output signals can be defined as an operator between two signal spaces. Let T be an operator mapping signals from X to Y , where X and Y are two normed linear spaces, then the norm of the system is defined as

Definition 3.2.3 (System Norm) *The norm of a system T is its induced norm which is defined as*

$$\|T\| = \sup_{x \neq 0} \frac{\|Tx\|}{\|x\|} \quad (3.1)$$

where $x \in X$, $Tx \in Y$.

The norm of a system is also called the gain of the system or gain of the operator.

We say a system T a bounded operator if and only if its induced norm is finite, i.e.,

$$\|T\| = \sup_{x \neq 0} \frac{\|Tx\|}{\|x\|} < \infty$$

Definition 3.2.4 *A linear or nonlinear system T is stable with respect to some input/output space if it is a bounded operator.*

3.2.3 Small Gain Theorem

Consider the interconnected system as shown in Figure 1.3 where G_1 and G_2 can be nonlinear time-varying systems. The inputs and outputs of the system satisfy equations 1.6 and 1.7.

Definition 3.2.5 (Well Posedness) *The feedback system shown in Figure 1.3 is well posed if for any given input u_1, u_2 in a normed linear space, there is a unique output y_1, y_2 in that normed linear space.*

Intuitively, well-posedness means that the system has a unique response for every input. In software terms, well-posedness implies that a system is deterministic but well-posedness is stronger than determinism because it also implies unique outputs for every input.

Definition 3.2.6 (causal) *An operator is causal if its current output does not depend on future inputs. An operator is strictly causal if its current output depends on past inputs, not including the current input.*

The following theorem provides a way to determine if a feedback system is well posed and well-posedness is a condition used in the small gain theorem.

Theorem 3.2.1 [2] *The feedback system shown in Figure 1.3 is well posed if the operator G_1G_2 is strictly causal, where G_1G_2 is an operator that cascades G_1 and G_2 .*

Stability of the interconnected system is defined in the following definition.

Definition 3.2.7 (Stability) *Let the closed-loop feedback system be expressed as*

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = H(G_1, G_2) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

where $H(G_1, G_2)$ is the operator mapping input u_1, u_2 to output y_1, y_2 , then the closed-loop system is l_p -stable if

$$\|H(G_1, G_2)\|_{l_p-ind} < \infty$$

Now we are ready to introduce the small gain theorem which will be used in digital controller verification.

Theorem 3.2.2 (Small Gain Theorem [2]) *Let $G_1 : l_p^n \rightarrow l_p^m$ and $G_2 : l_p^m \rightarrow l_p^n$ be two stable operators of bounded l_p -gain and assume that the closed loop system is well posed. Then the closed loop system is stable if $\|G_1\| \|G_2\| < 1$.*

The small gain theorem says that in the closed-loop system as shown in Figure 1.3, if both G_1 and G_2 are stable and the feedback system is well posed, and if the multiplication of the induced norm of G_1 with the induced norm of G_2 is less than 1, then the feedback closed-loop system is stable. The small gain theorem provides a foundation for the verification of digital controller implementation. In the presence of controller uncertainty, a feedback system G_r can be viewed as an ideal feedback system G , which is a system without controller uncertainty, together with controller uncertainty Δ_c , if both G and Δ_c are stable and the feedback system is well posed, then by checking if the multiplication of $\|G\| \|\Delta_c\|$ is less than 1, we can verify the stability of the closed-loop system.

3.3 Verification of Digital Controller Implementation

Returning to digital controller verification, the closed-loop system with controller uncertainty shown in Figure 3.2 can be viewed as an interconnected system with subsystem G and Δ_C as shown in Figure 3.4, where

- G consists of the plant P and the nominal controller C (the ideal controller without uncertainty),
- Δ_C is the controller uncertainty,
- r is the external input,
- u is control input to the plant,

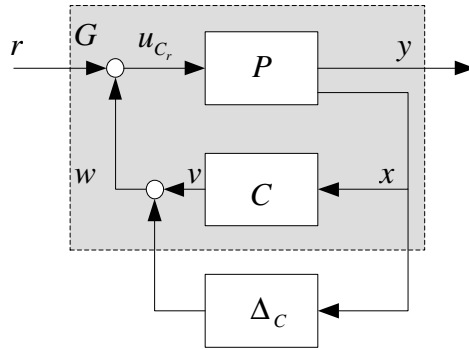


Figure 3.4: An interconnected system of an ideal control system and controller uncertainty

- y is the plant output,
- x is the state of the closed-loop system G ,
- $u_C = v + r$ is the output of the state-feedback controller C and
- $u_{C_r} = w + r$ is the output of the controller realization C_r , that is, the output of the controller program.

From section 3.1, we know the system G is designed to be stable, which means G is a stable operator on l_p space. Here we consider input and output signals in l_2 space, that is, G is a stable operator on l_2 space. Δ_C is a stable system because if the controller program properly handles “divide by zero”, given any finite input the program will produce finite output. This means Δ_C is a stable operator on l_2 space. We assume that the interconnected system shown in Figure 3.4 is well posed. Now we want to verify the stability of the feedback closed-loop system. By applying the small gain theorem, we conclude that the interconnected system is stable on l_2 if

$$\|G\|_{2-ind} \|\Delta_C\|_{2-ind} < 1 \quad (3.2)$$

which is

$$\|\Delta_C\|_{2-ind} = \sup_{x \neq 0} \frac{\|u_{C_r} - u_C\|_2}{\|x\|_2} < \frac{1}{\|G\|_{2-ind}} \quad (3.3)$$

Equation 3.3 provides a constraint on $\|\Delta_C\|_{2-ind}$. The equation means that: given any implementation of the controller C , if the 2-induced norm of uncertainty is less than this bound, the controller implementation will stabilize the interconnected system in the l_2 signal space. The problem then becomes computing $\|G\|_{2-ind}$. In the following section, we briefly introduce the background knowledge necessary to compute $\|G\|_{2-ind}$ and then give the computation of $\|G\|_{2-ind}$ using Fourier transformations.

3.3.1 Computation of Linear Operator Norms

We will introduce inner product space, linear operator, and frequency domain signal and system space in this section.

3.3.1.1 Inner Product Space

Definition 3.3.1 (General Inner Product) [17] *An inner product on a linear space \mathcal{V} is a function that maps each ordered pair of sequences x, y to a real (or complex) value denoted as $\langle x, y \rangle$ and satisfies the following axioms.*

1. $\langle x, y \rangle = \overline{\langle x, y \rangle}$ ³.
2. $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0$ if and only if $x = 0$.
3. $\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle$
4. $\langle x, \alpha y \rangle = \alpha \langle x, y \rangle$ for all $\alpha \in R$ or $\alpha \in C$.

Definition 3.3.2 *An inner product space is a linear space equipped with an inner product.*

Theorem 3.3.1 *All inner product spaces are normed linear spaces.*

³ $\overline{\langle x, y \rangle}$ is the conjugate of the complex number $\langle x, y \rangle$. The conjugate of a complex number is given by changing the sign of its imaginary part, for example, $\overline{a + jb} = a - jb$

Proof:

If X is an inner product space, then $\|x\| = \sqrt{\langle x, x \rangle}$ defines a norm on X . **End of**

Proof.

Let $x = \{x(0), x(1), \dots, x(k), \dots\}$ be a sequence in a normed linear space X . The sequence converges to an element $x_c \in X$ if

$$\lim_{k \rightarrow \infty} \|x_c - x(k)\| = 0$$

Intuitively, if x converges to x_c , then its elements get closer and closer to each other as k becomes larger. This property is captured in the following definition.

Definition 3.3.3 (Cauchy sequence) [2] A sequence $x = \{x(0), x(1), \dots, x(k), \dots\}$ is a Cauchy sequence if for every $\epsilon > 0$, there exists an N such that

$$\|x(k_1) - x(k_2)\| < \epsilon$$

for all $k_1, k_2 \geq N$

We say a normed linear space X is a complete normed linear space if every Cauchy sequence in X has a limit in X . A complete normed linear space is also called a Banach space.

Definition 3.3.4 A Hilbert space is a complete inner product space.

From the definition of the Hilbert space, we can see that the space of l_2^n is a Hilbert space with the inner product defined as

$$\langle x, y \rangle = \sum_{k=-\infty}^{\infty} y^T(k)x(k).$$

And the resulting norm is exactly the 2-norm as defined earlier.

3.3.1.2 Frequency Domain Signal Spaces

In section 3.2.1, we introduced time domain signal norms and spaces. Sometimes it is advantageous to present problems in frequency domain so as to make the problem simpler to solve and easier to conceptualize, especially the l_2 space. Here we introduce the frequency domain signal spaces and their norms.

The Fourier Transform replaces a signal defined in the time domain into one defined in the frequency domain. Let the Fourier Transform of a sequence x be

$$\hat{x}(e^{i\theta}) = \sum_{k=-\infty}^{\infty} x(k)e^{ik\theta},$$

then the space of all Fourier Transforms of $l_2^n(Z)$ with inner product defined as

$$\langle \hat{x}, \hat{y} \rangle = \frac{1}{2\pi} \int_0^{2\pi} \hat{y}^*(e^{i\theta}) \hat{x}(e^{i\theta}) d\theta$$

is also an inner product space. Here $*$ denotes the conjugate-transpose⁴ of a complex-valued matrix. The useful fact about Fourier Transform is that the inner product of elements in $l_2^n(Z)$ is preserved under the transform, which is

$$\langle x, y \rangle = \langle \hat{x}, \hat{y} \rangle$$

Let $\mathcal{L}_2^n[0, 2\pi]$ be the space of all complex vector-valued Lebesgue-integrable functions on $[0, 2\pi]$ with $\langle \hat{x}, \hat{x} \rangle < \infty$. Because the inverse Fourier Transform exists for every element in $\mathcal{L}_2^n[0, 2\pi]$, the following statements hold.

- If $x \in l_2^n(Z)$, then $\hat{x} \in \mathcal{L}_2^n[0, 2\pi]$.
- If $\hat{x} \in \mathcal{L}_2^n[0, 2\pi]$, then $x \in l_2^n(Z)$.

Let S be a subset of normed linear space, we say a point $x_0 \in S$ is an interior point [2] of S if there exists a ball of radius ϵ such that

$$B(x_0, \epsilon) = \{x \mid \|x - x_0\| < \epsilon, x \in S\}$$

⁴the conjugate transpose of an m-by-n matrix A with complex entries is the n-by-m matrix A^* obtained from A by taking the transpose and then taking the complex conjugate of each entry.

We say S is open if every point in S is an interior point. A function $f(x)$ on an open space S is said to be analytic at a point $z_0 \in S$ if it is differentiable at Z_0 and also at each point in some neighborhood of z_0 . A function is analytic in S if it is analytic at each point of S .

Let Z_+ be the set of all nonnegative integers and the space $l_2^n(Z_+)$ is the subset of $l_2^n(Z)$ such that its elements are zero for $t < 0$. Fourier Transforms of elements of $l_2^n(Z_+)$ are analytic continuous in the open unit disc. Let \mathcal{H}_2^n denote the space of all functions which are analytic in the open unit disc, then $\hat{x} \in \mathcal{H}_2^n$ means $\hat{x} \in \mathcal{L}_2^n[0, 2\pi]$ and \hat{x} is analytic in the open unit disc.

3.3.1.3 Linear Operators

T is a linear operator if it satisfies

$$T(\alpha x_1 + \beta x_2) = \alpha T(x_1) + \beta T(x_2)$$

for all $\alpha, \beta \in R$.

Definition 3.3.5 (time-invariant) *An operator is time-invariant if its action does not depend on the starting time.*

A linear operator on l_p signal space can be represented as a multiplication operator with infinite matrix R ,

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \end{bmatrix} = \begin{bmatrix} R(0,0) & R(0,1) & R(0,2) & \cdots \\ R(1,0) & R(1,1) & R(1,2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \end{bmatrix} \quad (3.4)$$

where $R(i, j)$ is the i th, j th block of matrix R . This matrix representation of the operator acts on inputs of l^n by multiplication, that is

$$y(k) = \sum_j^{\infty} R(k, j)u(j) \in R^m$$

If the system R is causal, then

$$R(i, j) = 0, \text{ for all } i < j.$$

If it is strictly causal, then

$$R(i, j) = 0, \text{ for all } i \leq j.$$

If it is time-invariant, then

$$R(i, j) = R(i + 1, j + 1).$$

Therefore, a linear, time-invariant, causal operator has the following infinite matrix form,

$$\begin{bmatrix} R(0) & 0 & \cdots & 0 \\ R(1) & R(0) & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ R(k) & R(k-1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.5)$$

Let $\mathcal{L}^{m \times n}$ denote the space of all linear, causal operators, then the infinite matrix in equation 3.4 becomes

$$\begin{bmatrix} R(0,0) & 0 & \cdots & 0 \\ R(1,0) & R(1,1) & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ R(k,0) & R(k,1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.6)$$

and the output of the operator can be obtained by

$$y(k) = \sum_j^k R(k, j)u(j) \in R^m$$

Operators in $\mathcal{L}^{m \times n}$ may or may not be bounded. We say an operator R is bounded on l_2 if and only if

$$\|R\| = \sup_k \sigma_{max}(R_k) < \infty.$$

where R_k is the k^{th} dimensional block of the matrix shown in equation 3.6, which is

$$\begin{bmatrix} R(0,0) & 0 & \cdots & 0 \\ R(1,0) & R(1,1) & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ R(k,0) & R(k,1) & \cdots & R(k,k) \end{bmatrix}$$

and $\sigma_{max}(R_k)$ is maximum singular value of R_k .⁵

3.3.1.4 Frequency Domain Operator Spaces

Just as the time domain signal space \mathcal{L}_2^n can be identified by the frequency domain signal space \mathcal{H}_2^n , the time domain operator space with bounded norms can also be identified by the frequency domain signal space.

Let $\mathcal{L}_\infty^{m \times n}$ denote the space of all complex-valued matrix functions on the unit circle with bounded norms, then $\hat{R} \in \mathcal{L}_\infty^{m \times n}$ means

$$\|\hat{R}\|_\infty = \text{ess sup}_\theta \sigma_{max}[\hat{R}(e^{i\theta})] < \infty. \quad (3.7)$$

where $\text{ess sup } f(x)$ is the essential supremum of function $f(x)$; essential supremum of $f(x)$ is the smallest number $a \in R$ for which $f(x)$ only exceeds a on a set of measure zero.

Define a multiplication operator $\hat{G} \in \mathcal{L}_\infty^{m \times n}$ on the space $\mathcal{L}_2^n[0, 2\pi]$ as

$$\hat{G} : \mathcal{L}_2^n[0, 2\pi] \rightarrow \mathcal{L}_2^n[0, 2\pi],$$

we have

$$\|\hat{G}\hat{x}\|_2 = \left(\frac{1}{2\pi} \int_0^{2\pi} |\hat{G}(e^{i\theta})\hat{x}(e^{i\theta})|_2^2 d\theta \right)^{\frac{1}{2}} \leq \text{ess sup}_\theta \sigma_{max}[\hat{G}(e^{i\theta})] \|\hat{x}\|_2. \quad (3.8)$$

Therefore, the induced norm of this operator \hat{G} satisfies

$$\|\hat{G}\| = \sup_{\hat{x} \neq 0} \frac{\|\hat{G}\hat{x}\|}{\|\hat{x}\|}$$

⁵The singular values of a matrix A are the positive square roots of the nonzero eigenvalues of $A * A$

$$\begin{aligned}
&= \operatorname{ess\,sup}_{\theta} \sigma_{\max}[\hat{G}(e^{i\theta})] \\
&\leq \|\hat{G}\|_{\infty}.
\end{aligned}$$

Let $\mathcal{H}_{\infty}^{m \times n}$ denote the subspace of $\mathcal{L}_{\infty}^{m \times n}$ in which all the elements are analytic continuous in the unit disc. Then $\hat{R} \in \mathcal{H}_{\infty}^{m \times n}$ means \hat{R} is analytic in the open unit disc and bounded on the unit circle. The following theorem shows when the equality in equation 3.8 holds.

Theorem 3.3.2 [2] *For every bounded linear time-invariant, causal operator G on $l_2(Z_+)$, there exists a multiplication operator $\hat{G} \in \mathcal{H}_{\infty}^{m \times n}$ on \mathcal{H}_2^n such that $y = Gx$ satisfies $\hat{y}(e^{i\theta}) = \hat{G}(e^{i\theta})\hat{x}(e^{i\theta})$. The induced norm of this operator is equal to $\|\hat{G}\|_{\infty}$.*

The usefulness of theorem 3.3.2 is that it provides a way to compute the norm of an operator in $\mathcal{L}^{m \times n}$ which equals the norm of the corresponding multiplication operator. That is, to compute the 2-induced norm of linear time-invariant, causal operator G , we first transform G to a frequency domain operator \hat{G} , then compute $\|\hat{G}\|$ using equation 3.7.

3.3.1.5 Computing $\|G\|_{2-ind}$

Now let us compute $\|G\|_{2-ind}$. Recall that the closed-loop system G which consists of plant P and nominal controller C as described in equation 1.3 and 1.5 is

$$\begin{aligned}
x(k+1) &= A_k x(k) + B_k u(k) \\
&= A_k x(k) + B_k (\alpha r(k) - Kx(k)) \\
&= (A_k - B_k K)x(k) + \alpha B_k r(k) \\
&= (A_k - B_k K)^k x(0) + \alpha \sum_{i=0}^{k-1} (A_k - B_k K)^{k-i-1} B_k r(i)
\end{aligned}$$

Assume $x(0) = 0$, then the input to state system equation is

$$x(k+1) = \alpha \sum_{i=0}^{k-1} (A_k - B_k K)^{k-i-1} B_k r(i),$$

which can be represented in a matrix form as follows,

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \alpha B_k & 0 & \cdots & 0 \\ \alpha(A_k - B_k K) & \alpha B_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(A_k - B_k K)^k & \alpha(A_k - B_k K)^{k-1} & \cdots & 0 \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ r(2) \\ \vdots \\ r(k) \end{bmatrix} \quad (3.9)$$

The infinite matrix in equation 3.9 has the same form as equation 3.5 except that $R(0) = 0$, which shows that closed-loop system G is a linear, strictly causal and time-invariant system.

From theorem 3.3.2, we get

$$\|G\|_{2-ind} = \|\hat{G}\|_{\infty} \quad (3.10)$$

There are many algorithms which can be used to compute the \mathcal{H}_{∞} norm of a linear, time-invariant operator [18][19][20]. In this thesis, we make use of a free scientific software package, Scilab, to compute the $\|\hat{G}\|_{\infty}$.

Now that we have the value of $\|G\|_{2-ind}$ which is equal to $\|G\|_{\infty}$, we need to compute the 2-induced norm of the software uncertainty, which is

$$\sup_{x \neq 0} \frac{\|u_{C_r} - u_C\|_2}{\|x\|_2}. \quad (3.11)$$

Then by verifying that

$$\sup_{x \neq 0} \frac{\|u_{C_r} - u_C\|_2}{\|x\|_2} < \frac{1}{\|G\|_{\infty}} \quad (3.12)$$

we can verify that the controller program C_r stabilizes the closed-loop system.

The computation of 2-induced norm of controller uncertainty Δ_k is presented in the following chapter.

Chapter 4

Computation of 2-induced Norm of Controller Uncertainty

In chapter 3, we reduced the verification problem to the problem of computing the norm of controller uncertainty Δ_C . The problem we now face is that given a piece of controller software, we do not have the mathematical model of this software which can be used in established algorithms to compute the 2-induced norm.

Instead, we will compute the 2-induced norm of Δ_C through its definition, which is equation 3.11.

$$\sup_{x \neq 0} \frac{\|u_{C_r} - u_C\|_2}{\|x\|_2}.$$

The definition requires that the 2-induced norm is computed over all state sequences $\forall x \in R$ and over all time. In general, since the state space consists of infinite number of states and the time domain is infinite, these two infinite ranges make the induced norm uncomputable. In this chapter, we provide a method to approximate a lower bound of this 2-induced norm and this lower bound can be used to determine when the system may be unstable.

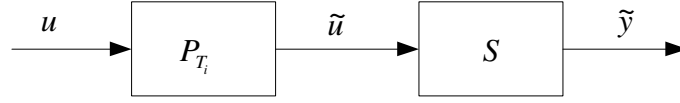


Figure 4.1: A nonlinear operator with truncated input

4.1 Computing the 2-induced Norm of Software Contronller Uncertainty

Let P_k , $k \in Z_+$, denote the standard truncation operator on l^n [2], then

$$P_k(x(0), x(1), \dots) = (x(0), x(1), \dots, x(k), 0, 0, \dots). \quad (4.1)$$

Let S be a nonlinear operator, let P_{T_i} be an input truncation operator, and let u be an input from S 's input space and its truncation \tilde{u} be

$$\begin{aligned} \tilde{u} &= P_{T_i}u \\ &= P_{T_i}(u(0), u(1), \dots) \\ &= (u(0), u(1), \dots, u(T_i), 0, 0, \dots). \end{aligned} \quad (4.2)$$

The following definition describes the norm of an operator with truncated input.

Definition 4.1.1 Let $S \diamond P_{T_i}$ be the operator S with truncated inputs as shown in figure 4.1, and let \tilde{y} be the output of operator S under truncated input \tilde{u} , then

$$\|S \diamond P_{T_i}\| = \sup_{\tilde{u} \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \quad (4.3)$$

The norm under truncated inputs is defined because we can only feed inputs to the system in finite time in simulation.

If S is a linear operator, then

$$S \diamond P_{T_i} = S \times P_{T_i}, \quad (4.4)$$

which is the multiplication of matrices S and P_{T_i} .

The norm of an operator with truncated inputs from a subset of the operator's input space is,

Definition 4.1.2 *Let I denote a subset of operator S 's input space then*

$$\|S \diamond P_{T_i}\|_I = \sup_{u \neq 0, u \in I} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \quad (4.5)$$

The norm under truncation is defined over a limited input space because we can only visit a finite subset of the inputs in simulation. We are now ready to prove that the norm over truncated inputs is less than or equal to the true norm.

Theorem 4.1.1 *Let S be a nonlinear operator, let P_{T_i} be an input truncation operator, let I denote a subset of S 's input space, then*

$$\|S \diamond P_{T_i}\|_I \leq \|S\|. \quad (4.6)$$

Proof:

We consider two cases, the first case is that the operator S has bounded norm, the second case is that the operator S 's norm is not bounded.

Case 1: the operator S has bounded norm. First, note that if any input u is in S 's input space, and its truncation \tilde{u} is also in its input space, so that

$$\|S \diamond P_{T_i}\| \leq \|S\| \quad (4.7)$$

Since S has bounded norm, we know that S has finite gain by Definition 3.2.4, i.e.

$$\|S\| = \sup_{u \neq 0} \frac{\|y\|}{\|u\|} < \infty \quad (4.8)$$

where u, y are the input and output of operator S and $u \in S$'s input space.

Because the operator S has finite gain, there must exist an input $u^* \in S$'s input space such that with this input, S obtains its maximum gain, that is,

$$\frac{\|y\|}{\|u\|} \leq \|S\| = \frac{\|y^*\|}{\|u^*\|} \quad (4.9)$$

where y^* is the output of S with input u^* .

Now we truncate any input u in S 's input space, before feeding it into operator S . Since u and \tilde{u} are both in S 's input space, based on the hypothesis that u^* gives S its maximum gain, we obtain

$$\sup_{u \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \leq \frac{\|y^*\|}{\|u^*\|} = \|T\| \quad (4.10)$$

where \tilde{y} is the output of S with truncated input \tilde{u} .

Because

$$\|S \diamond P_{T_i}\| = \sup_{u \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|}, \quad (4.11)$$

we have

$$\|S \diamond P_{T_i}\| \leq \|S\|.$$

Since I is a subset of S 's input space, it is clear that

$$\begin{aligned} \|S \diamond P_{T_i}\|_I &= \sup_{u \neq 0, u \in I} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\ &\leq \sup_{u \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\ &= \|S \diamond P_{T_i}\|. \end{aligned}$$

Based on equation 4.7, we conclude that

$$\|S \diamond P_{T_i}\|_I \leq \|S\|.$$

Case 2: the operator S 's norm is not bounded. It is clear that

$$\|S \diamond P_{T_i}\|_I \leq \infty = \|S\|.$$

End of Proof.

Let P_{T_o} be an output truncation operator, let \tilde{y} be the output of a non-linear operator under truncated input \tilde{u} which is an infinite sequence, then $\tilde{\tilde{y}}$ denotes the

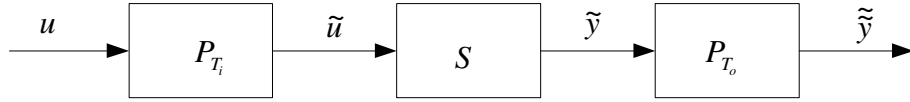


Figure 4.2: A nonlinear operator with truncated input and truncated output

truncation of \tilde{y} , that is,

$$\begin{aligned}
 \tilde{\tilde{y}} &= P_{T_o} \tilde{y} \\
 &= P_{T_o}(\tilde{y}(0), \tilde{y}(1), \dots) \\
 &= (\tilde{y}(0), \tilde{y}(1), \dots, \tilde{y}(T_o), 0, 0, \dots).
 \end{aligned} \tag{4.12}$$

The norm of an operator with only truncated input and truncated output is characterized in the following definition.

Definition 4.1.3 Let S be a nonlinear operator, let P_{T_i} be an input truncation operator and P_{T_o} be an output truncation operator. Then $P_{T_o} \diamond S \diamond P_{T_i}$ denotes the operator S with only truncated inputs and truncated outputs as shown in Figure 4.2, and

$$\|P_{T_o} \diamond S \diamond P_{T_i}\| = \sup_{u \neq 0} \frac{\|\tilde{\tilde{y}}\|}{\|\tilde{u}\|} \tag{4.13}$$

If S is a linear operator, then

$$P_{T_o} \diamond S \diamond P_{T_i} = P_{T_o} \times S \times P_{T_i}, \tag{4.14}$$

which is a multiplication of matrices P_{T_o} , S and P_{T_i} .

Definition 4.1.4 Let S be a nonlinear operator, let P_{T_i} be an input truncation operator and P_{T_o} be an output truncation operator. Let I denote a subset of S 's input space, then

$$\|P_{T_o} \diamond S \diamond P_{T_i}\|_I = \sup_{u \neq 0, u \in I} \frac{\|\tilde{\tilde{y}}\|}{\|\tilde{u}\|} \tag{4.15}$$

The norm under truncation outputs is defined because we can only observe the system outputs in finite time in simulation. Now we are ready to prove that the norm over truncated inputs and truncated outputs is less than or equal to the true norm.

Theorem 4.1.2 *Let S be a nonlinear operator which is causal, let P_{T_i} be an input truncation operator and P_{T_o} be an output truncation operator. Let I denote a subset of S 's input space and \tilde{I} denote the subset which consists of truncations of all inputs from I . If \tilde{I} is in S 's input space*

$$\|P_{T_o} \diamond S \diamond P_{T_i}\|_I \leq \|S\| \quad (4.16)$$

Proof:

First we prove that, if any input u is in S 's input space, and its truncation $\tilde{u} = P_{T_i}u$ is also in S 's input space, then

$$\|P_{T_o} \diamond S \diamond P_{T_i}\| \leq \|S\| \quad (4.17)$$

Since operator S is causal, its output does not depend on its future inputs. Therefore for any output y , if truncated, its norm becomes smaller, in other words,

$$\|P_{T_o}y\| \leq \|y\| \quad (4.18)$$

Dividing both sides with the norm of input u , we have

$$\frac{\|P_{T_o}y\|}{\|u\|} \leq \frac{\|y\|}{\|u\|}. \quad (4.19)$$

If input u is replaced by truncated input \tilde{u} , where $\tilde{u} = P_{T_i}u$, and output is replaced with \tilde{y} , which is the output of S under input \tilde{u} , equation 4.19 becomes,

$$\begin{aligned} \frac{\|P_{T_o}\tilde{y}\|}{\|\tilde{u}\|} &= \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\ &\leq \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \end{aligned} \quad (4.20)$$

From equation 4.20, we have

$$\begin{aligned}
\|P_{T_o} \diamond S \diamond P_{T_i}\| &= \sup_{u \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\
&\leq \|S \diamond P_{T_i}\| \\
&\leq \|S\|.
\end{aligned} \tag{4.21}$$

Since I is a subset of S 's input space, it is clear that

$$\begin{aligned}
\|P_{T_o} \diamond S \diamond P_{T_i}\|_I &= \sup_{u \neq 0, u \in \tilde{I}} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\
&\leq \sup_{u \neq 0} \frac{\|\tilde{y}\|}{\|\tilde{u}\|} \\
&= \|P_{T_o} \diamond S \diamond P_{T_i}\|.
\end{aligned} \tag{4.22}$$

Based on equations 4.21 and 4.22, we conclude that

$$\|P_{T_o} \diamond S \diamond P_{T_i}\|_I \leq \|S\|.$$

End of Proof.

The norm of an operator with truncated inputs and truncated outputs have the following properties.

1. For two input truncation operators $P_{T_{i,1}}$ and $P_{T_{i,2}}$, if $T_{i,1} < T_{i,2}$, then

$$\|P_{T_o} \diamond S \diamond P_{T_{i,1}}\| \leq \|P_{T_o} \diamond S \diamond P_{T_{i,2}}\| \tag{4.23}$$

that is, $\|P_{T_o} \diamond S \diamond P_{T_i}\|$ increases monotonically as T_i increases.

2. For two output truncation operators $P_{T_{o,1}}$ and $P_{T_{o,2}}$, if $T_{o,1} < T_{o,2}$, then

$$\|P_{T_{o,1}} \diamond S \diamond P_{T_i}\| \leq \|P_{T_{o,2}} \diamond S \diamond P_{T_i}\|, \tag{4.24}$$

that is, $\|P_{T_o} \diamond S \diamond P_{T_i}\|$ increases monotonically as T_o increases.

3. For two sets I_1 and I_2 of S 's input space, if $I_1 \subset I_2$, then

$$\|P_{T_o} \diamond S \diamond P_{T_i}\|_{I_1} \leq \|P_{T_o} \diamond S \diamond P_{T_i}\|_{I_2}, \tag{4.25}$$

that is, $\|P_{T_o} \diamond S \diamond P_{T_i}\|_I$ increases monotonically as the cardinality of I increases.

4. In theorem 4.1.2, as T_i and T_o go infinity and I approaches S 's input space, $\|P_{T_o} \diamond S \diamond P_{T_i}\|_I$ will approach $\|S\|$, that is, the following equation holds:

$$\lim_{\substack{T_i \rightarrow \infty, \\ T_o \rightarrow \infty, \\ I \rightarrow S's \text{ input space}}} \|P_{T_o} \diamond S \diamond P_{T_i}\|_I = \|S\| \quad (4.26)$$

These properties ensure that tighter bounds on the norm can be found at the expenses of more computation.

Theorem 4.1.2 provides a theoretical basis to compute a lower bound of the 2-induced norm of Δ_C . In the following section, we introduce an algorithm to compute a lower bound of the 2-induced norm of Δ_C .

4.2 Algorithm to Compute A Lower Bound of the Software Controller Uncertainty

Figure 4.3 shows the components and variables needed to compute a lower bound of $\|\Delta_C\|_{2-ind}$. Figure 4.4 shows the algorithm. The basic idea of the algorithm is, for each reference input $r \in I_r$, first obtain the trajectories of x and u_{C_r} up to truncation time T of the closed-loop system G_r which consists of P and C_r , then compute the trajectories of u_r of C using the same state trajectories as obtained in the closed-loop system G_r . After that compute

$$\delta_{j,k} = \frac{\|u_{C_r} - u_C\|_2}{\|x\|_2} \quad \text{for each } j = \{1, \dots, M\}, k = \{1, \dots, N\}$$

Then the maximum of $\delta_{j,k}$ is a lower bound of $\|\Delta_C\|$. The larger the truncation value T and the larger the input set I_r , the closer this lower bound is to $\|\Delta_C\|$.

Three components are necessary to compute a lower bound on $\|\Delta\|_{2-ind}$. The first is the mathematical model of the plant P that is to be controlled. The second is the mathematical model of a digital controller C which is designed to stabilize the closed-loop system. Both of these models are discrete and the sampling time is t_s . The third is the software C_r which implements the digital controller model.

- 1: **components**
- 2: P : plant to be controlled;
- 3: C : mathematical model of the digital controller;
- 4: C_r : software implementation of the digital controller;
- 5: **variables**
- 6: r : external reference input;
- 7: I_r : a subset of external reference inputs;
- 8: M : the cardinality of I_r ;
- 9: N : the total number of trajectories of x ;
- 10: u_C : output of C and also control input to plant P ;
- 11: x : state of plant P which is fed back to C and C_r ; $x(0) = 0$;
- 12: X_{state} : a set trajectories of state x , spanned from the initial condition $x(0)$;
- 13: u_{C_r} : output of C_r and also control input to plant P ;
- 14: T : time at which signals are truncated;
- 15: t_s : sampling time;

Figure 4.3: Component and variables needed to compute the lower bound of $\|\Delta_k\|$


```

1: begin
2: /* compute the trajectories of  $u_{C_r}$  and  $x$  at each sampling time */
3: for each  $r$  in  $I_r$  do
4:   initialize the execution of  $C_r$ ;
5:   for each sampling time  $0 \leq i \leq a = \frac{T}{t_s}$  do
6:     for each trajectory  $x$  in  $X_{state}$  do
7:       feed  $r(i)$  and  $x(i)$  to  $C_r$ , observe its output  $u_{C_r}(i)$ ;
8:       save data  $(u_{C_r}(i), x(i))$ ;
9:       feed  $u_{C_r}$  to  $P$ , compute  $x(i+1)$ ;
10:      consider non-determinism (due to A/D errors or sensor errors), by mapping
           $x(i+1)$  into a set  $(x_1(i+1), x_2(i+1), \dots, x_n(i+1))$ ; and each trajectory of  $x$ 
          is expanded into  $n$  trajectories
11:     end for
12:   end for
13:   stop execution of  $C_r$ ;
14: /* using the above obtained  $x$ , compute the trajectories of  $u_C$  at each sampling
          time */
15: for each sampling time  $0 \leq i \leq a = \frac{T}{t_s}$  do
16:   for each trajectory  $x$  in  $X_{state}$  do
17:     feed  $r(i)$  and  $x(i)$  to  $C$ , compute its output  $u_C$ ;
18:     save data  $(u_C(i), x(i))$ ;
19:   end for
20: end for
21: /* compute  $\frac{\|u_{C_r} - u_C\|_2}{\|x\|_2}$  for each trajectory */
22: for each trajectory  $x$  in  $X_{state}$  do
23:   compute  $\delta_{j,k} = \frac{\sqrt{|u_{C_r}(0) - u_C(0)|^2 + |u_{C_r}(1) - u_C(1)|^2 + \dots + |u_{C_r}(a) - u_C(a)|^2}}{\sqrt{|x(0)|^2 + |x(1)|^2 + \dots + |x(a)|^2}}$ 
24: end for
25: end for
26: /* compute a lower bound of  $\|\Delta_k\|$  */
27:  $\|P_T \diamond \Delta_C \diamond P_T\|_{I_r} = \max\{\delta_{1,1}, \delta_{1,2}, \dots, \delta_{1,N}, \dots, \delta_{M,1}, \delta_{M,2}, \dots, \delta_{M,N}\}$ 
28: end

```

Figure 4.4: Algorithm to compute a lower bound of $\|\Delta_C\|$

Then we choose a set of external reference inputs I_r which we want the closed-loop system to follow. At each sampling time $i \times t_s$, every variable is referenced by using index i , for example, state value at sampling time i is $x(i)$ as in line 7. Due to the non-determinism of A/D devices and sensors, given any reference input r , there is a set $(x(i), u_{C_r}(i))$ corresponding to a given, that is, there are a set of state trajectories x and a set of control input trajectories u_{C_r} corresponding to each reference input r as in line 10. Trajectory set X_{state} contains all of these trajectories. Since the initial state is assumed 0, given any reference input r , at the first sampling time, X_{state} only contains $x(0) = 0$ in line 7. After feeding $r(0)$ and $x(0)$ into C_r , we get $u_{C_r}(0)$. We then compute $x(1)$ based on the mathematical model of P . Then taking non-determinism into account, the value of $x(1)$ is mapped into a set $(x_1(1), x_2(1), \dots, x_n(1))$. Now

$$X_{state} = \{(x(0), x_1(1)), (x(0), x_2(1)), \dots, (x(0), x_n(1))\},$$

which means X_{state} contains n trajectories. For each $x_i(1)$, $i = \{1, 2, \dots, n\}$ together with $r(i)$ fed into C_r , there is a corresponding output $u_{C_r,i}(1)$, $i = \{1, 2, \dots, n\}$ for each $x(1)$. Similarly, at the third sampling time,

$$X_{state} = \{(x(0), x_1(1), x_1(2)), (x(0), x_1(1), x_2(2)), (x(0), x_1(1), x_n(2)), \\ (x(0), x_2(1), x_1(2)), \dots, (x(0), x_n(1), x_n(2))\},$$

which means after 2 transitions, X_{state} contains n^2 trajectories. Repeat this process up to the $a = \frac{T}{t_s}$ th sampling time, where T is the point at which time is truncated, at which time X_{state} contains n^a trajectories. The expansion of the trajectories is shown in Figure 4.5.

The steps to check if a digital controller implementation may destabilize a plant are, first compute $\|G\|_{2-ind}$ using equation 3.10 based on the mathematical model of the plant and the mathematical model of the controller. Then compute a lower bound of $\|\Delta_C\|$ using $\|P_{T_o} \diamond \Delta_C \diamond P_{T_i}\|_{I_r}$, according to the algorithm as shown in Figure 4.4.

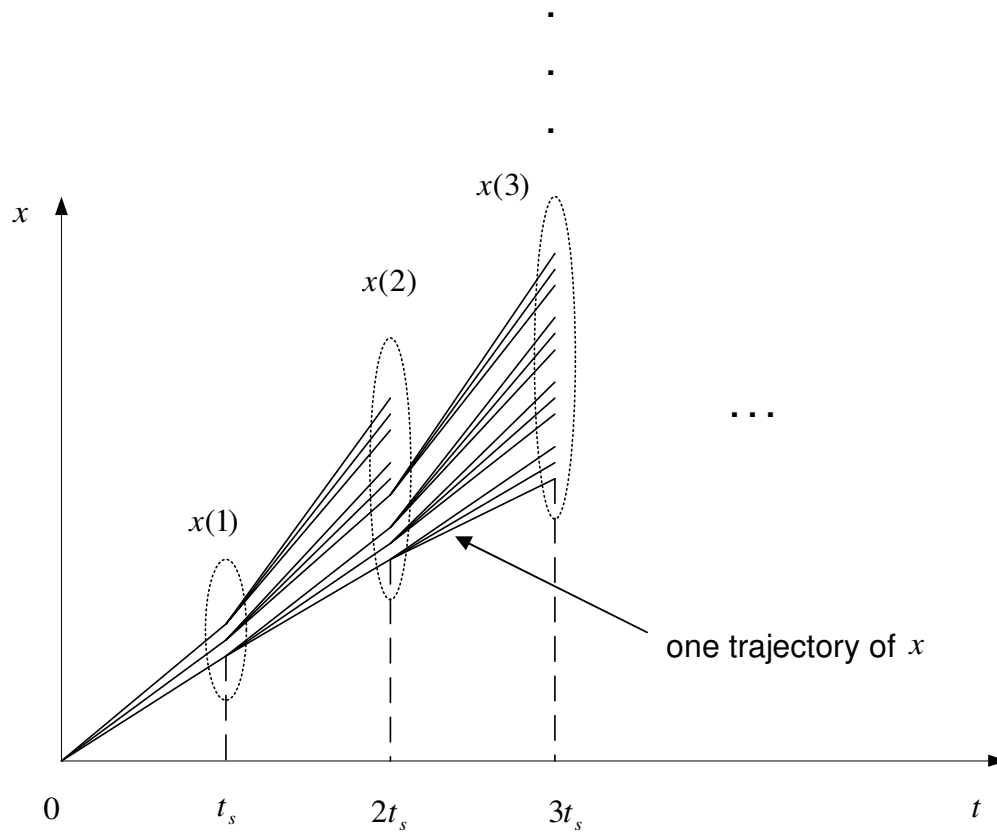


Figure 4.5: State trajectory expansion due to non-determinism.

Then check if $\|P_{T_o} \diamond \Delta_C \diamond P_{T_i}\|_{I_r} < \frac{1}{\|G\|_{2-ind}}$. If $\|P_{T_o} \diamond \Delta_C \diamond P_{T_i}\|_{I_r} \leq \frac{1}{\|G\|_{2-ind}}$, the result is inconclusive because we don't know if $\|\Delta_C\| < \frac{1}{\|G\|_{2-ind}}$. If the result is $\|P_{T_o} \diamond \Delta_C \diamond P_{T_i}\|_{I_r} > \frac{1}{\|G\|_{2-ind}}$, the digital controller implementation may destabilize the closed-loop system in the l^2 input space, and the closed-loop system is unsafe in certain sense. A conservative approach suggests that the controller designer needs to either redesign the mathematical controller or adjust the control parameters.

Chapter 5

Conclusions and Future Work

This work provides a theoretical basis to compute a lower bound of a system's 2-induced norm for black box systems. Based on the theory, an algorithm to compute a lower bound of the 2-induced norm of the uncertainty in the software implementation is presented. After the lower bound is computed, it is compared with $\frac{1}{\|G\|_{2-ind}}$, where G is the closed-loop system consisting mathematical model of the plant to be controlled and the mathematical model of the digital controller designed. If the lower bound is greater, the software implementation of the digital controller may destabilize the closed-loop system under l^2 inputs, in other words, the closed-loop system is unsafe, otherwise, the result is inconclusive. The unsafeness of the closed-loop system suggests that the controller designer needs to redesign the mathematical controller.

This verification method does not require experiments which are expensive and can be dangerous. This method does not require a mathematical approximation of the software implementation which ignores implementation details and which may cause false errors. The presented algorithm can be implemented in an explicit model checker, such as Estes¹. If the algorithm is implemented in an explicit model checker,

¹Estes is an explicit model checker developed by Verification and Validation Lab, Computer Science Department, Brigham Young University.

then not only can stability property of the closed-loop system be verified, but software bugs due to implementation details can also be found. Furthermore, a software implementation which controls several plants concurrently can also be verified.

5.1 Future Work

The future work related to this work is:

- We know that the small gain condition in the small gain theorem is sufficient to guarantee internal stability if Δ_C is a nonlinear and time-varying stable operator [3]. We don't know if the small gain condition is necessary to guarantee internal stability when Δ_C is nonlinear, time-varying and stable. If the condition is necessary, then this work provides a method to determine if a software implementation of a controller will definitely destabilize the closed-loop system.
- Implement the algorithm in an explicit model checker to compute a lower bound of the 2-induced norm of the uncertainty in a software implementation of a digital controller.
- Complete a case study. Design or obtain a digital controller to control a linear time-variant system. Implement or obtain a software implementation of this digital controller. Compute a lower bound of the 2-induced norm of the uncertainty in this controller software, and check if this controller software may destabilize the closed-loop system.
- Consider a controller software implementation that controls several plants concurrently. Verify if this implementation may destabilize any one of the closed-loop systems. And verify bugs related to concurrency issues. This would be particularly difficult using related methods that compute mathematical models of software because the models would need to include an accurate model of concurrency.

- Consider plant uncertainty when verifying a software implementation. Include plant uncertainty in the analysis framework.

LIST OF REFERENCES

- [1] R. T. Stefani, B. Shahian, C. J. Savant, and G. H. Hostetter, Design of Feedback Control Systems. Oxford University Press, 2002.
- [2] M. A. Dahleh and I. J. Diaz-Bobillo, Control of Uncertain Systems A Linear Programming Approach. Prentice Hall, 1995.
- [3] K. Zhou and J. C. Doyle, Essentials of Robust Control. Prentice Hall, 1998.
- [4] C. A. Desoer and M. Vidyasagar, Feedback System: Input-Output Properties. Prentice Hall, 1975.
- [5] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, “The algorithmic analysis of hybrid systems,” in Theoretical Computer Science, 1995, pp. 138:3–34.
- [6] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, “Hytech: A model checker for hybrid systems,” in Journal on Software Tools for Technology Transfer, 1997.
- [7] A. Puri and P. Varaiya, “Decidability of hybrid systems with rectangular differential inclusions,” in Proceeding of 6th Workshop on Computer Aided Verification, Springer-Verlag, 1994, pp. 95–104.
- [8] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas, “Discrete abstractions of hybrid systems,” in Proceedings of the IEEE, 2000, pp. 88(2):971–984.

- [9] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi, “Computational techniques for the verification and control of hybrid systems,” in Proceedings of the IEEE, Volume 91, Number 7, July 2003, pp. 986–1001.
- [10] G. D. Penna, B. Intrigila, I. Melatti, M. Minichino, E. Ciancamerla, A. Parisse, E. Tronci, and M. V. Zilli, “Automatic verification of a turbogas control system with the Mur ϕ verifier,” in HSCC, 2003, pp. 141–155.
- [11] D. Nesic and D. Liberzon, “A small-gain approach to stability analysis of hybrid systems,” in Proceedings of the 44th Conference on Decision and Control, Seville, Spain, Dec 2005.
- [12] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, Feedback control theory. New York: MacMillan, 1992. [Online]. Available: citeseer.ist.psu.edu/doyle90feedback.html
- [13] L. H. Keel and S. P. Bhattacharyya, “Robust, fragile, or optimal,” in IEEE Transactions on Automatic Control, vol. 42, no. 8, 1997, pp. 1098–1105.
- [14] J. F. Whidborne, R. S. H. Istepanian, and J. Wu, “Reduction of controller fragility by pole sensitivity minimization,” in IEEE Transactions on Automatic Control, vol. 46, no. 2, 2001, pp. 320–325.
- [15] J. Wu, S. Chen, R. H. I. G. Li, and J. Chu, “An improved closed-loop stability related measure for finite-precision digital controller realizations,” in IEEE Transactions on Automatic Control, vol. 46, no. 7, 2001, pp. 1162–1166.
- [16] M. Roozbehani, E. Feron, and A. Megretski, “Modeling, optimization and computation for software verification.” in HSCC, 2005, pp. 606–622.
- [17] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. SIAM, 2000.

- [18] N. A. Bruinsma and M. Steinbuch, "A fast algorithm to computer the h infinity norm of a transfer function matrix," in Syst. Control Lett, vol. 14, no. 4. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V., 1990, pp. 287–293.
- [19] G. Linxin, A. Xue, and S. Youxian, "On an explicit expression to compute h infinity norm of symmetric systems," in Proceedings of the American Control Conference, vol. 5. Denver, CO, United States: Institute of Electrical and Electronics Engineers Inc., 2003, pp. 4207–4212.
- [20] B. C. Chang and S. S. Banda, "Optimal h infinity norm computation for multivariable systems with multiple zeros," in IEEE Transactions on Automatic Control, vol. 34, no. 5, 1989, pp. 553–557.