An Engineering Method for Safety Region Development

Danbing Seto
Lui Sha

August 1999

TECHNICAL REPORT
CMU/SEI-99-TR-018
ESC-TR-99-018

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

An Engineering Method for Safety Region Development

CMU/SEI-99-TR-018
ESC-TR-99-018

Danbing Seto
Lui Sha

August 1999

Dependable System Upgrade

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Norton L. Compton, Lt Col., USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

Abstract

In this report, we study tolerance of semantic faults, one of the crucial issues in the Simplex™ architecture. In particular, we examine semantic faults that cause the controlled device to be unsafe (i.e., unable to carry out its normal operation) and eventually cause the device to become damaged. We also consider fault detection as a safety check. For the class of control systems operating around an equilibrium, the objective of maintaining the safety of the controlled device is formulated as a stabilization problem, and the safety of the controlled device is tested against the stability region of the device under the safety control. To establish the stability region, we apply the Lyapunov stability theory and linear matrix inequality (LMI) methodologies. It is shown that the stability region for a given safety controller as well as a safety control law can be systematically derived by LMI-based approaches. We conclude the report with a summary of the procedure for deriving the safety check and safety controller for a given application.

™ Simplex is a trademark of Carnegie Mellon University.

1 Introduction

With the rapid advancement of computing technology, the real-time control of physical devices has shifted from the analog domain to the digital domain, and control implementations have become an issue of software development. The so-called real-time computer-controlled systems have been seen in all practices[na1], ranging from simple motion-control systems to large-scale, complex systems. Figure 1 shows a typical real-time computer-controlled system.

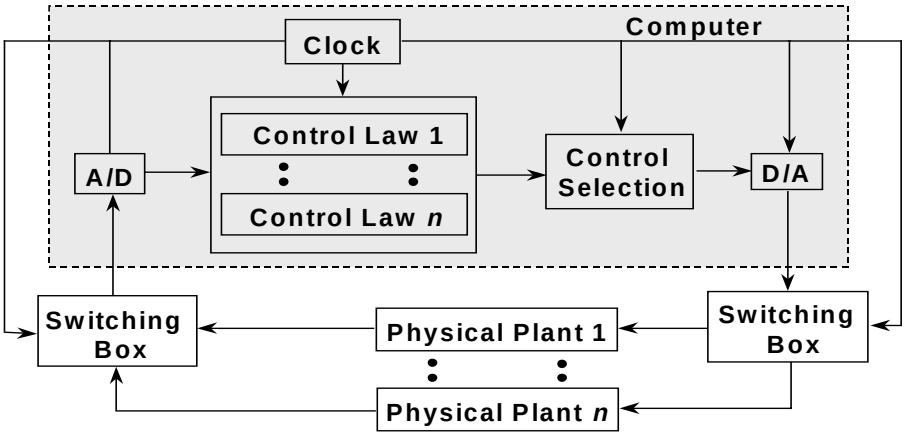

Figure 1: A Typical Real-Time Computer-Controlled System

In this system, the control unit is a computer, which consists of an analog to digital (A/D) converter, computing processes to generate control commands, a real-time clock, and a digital to analog (D/A) converter. The control unit controls a group of physical plants. The real-time clock governs periodic sampling of the physical plants and control update. For each sample, measurements of the physical plants are fed to the control unit and are converted to digital signals through the A/D converter. Based on these measurements, control commands are computed, converted to analog signals through the D/A converter, and sent to the physical plants. Such a control update cycle is repeated at a prescribed sampling rate. For ease of exposition, we have defined a controller as the software implementation of a control law, a physical plant (or plant) as the physical device to be controlled, and the overall system as the complete computer-controlled system. It is worthwhile to emphasize that Figure 1 presents only the basic configuration of a computer-controlled system. In a large-scale, complex system, the system shown in Figure 1 could be a subsystem, which is often referred to as an embedded system.

To take full advantage of advanced computing technology, most users want to be able to upgrade and evolve computer-controlled systems (especially large-scale, complex systems). In most cases, the ability to upgrade and evolve the overall system depends on the system's ability to adopt software changes. Since many systems are life critical, reliability and availability are the essential requirements for these systems. To achieve high reliability and availability when the system is upgrading or evolving, it is better to introduce software change in a safe and reliable fashion while the system is running. The Simplex™ architecture is designed for this purpose. By facilitating replacement units and analytically redundant controllers, the Simplex architecture allows an upgraded controller to be introduced to the system online to control the plant under the protection of the so-called safety controller. The upgraded controller will continue to control the plant unless it contains faults that will cause the plant to malfunction. If this is the case, the safety controller will take over when the fault is detected. In this report, we study the issue of fault detection related to controller design and implementation, and the control switching logic for fault tolerance. In particular, we focus on establishing the safety region (to be defined precisely in subsequent sections) and propose a systematic approach for deriving the safety region and designing the safety controller.

This report is organized as follows. In Section 2, we briefly review the Simplex architecture and formally define the notion of safety region. In Section 3, we establish the relation between the safety region and the stability region for a class of control systems, and we define the safety control objective as stabilization of the plant. The stability analysis is carried out based on the Lyapunov stabilization theory. In Section 4, we formulate the stabilization control as a linear matrix inequality (LMI)[1] problem and solve the problem by using the existing approaches in LMI literature. In particular, we first derive the stability region for the closed-loop system under a given linear state feedback control, and then design a state feedback control and derive the corresponding stability region. Furthermore, we discuss the design of the state feedback control with certain prescribed performance requirements. In Section 5, we conclude the report with a summary of what has been done and the lessons learned.

™ Simplex is a trademark of Carnegie Mellon University.

[1] A linear matrix inequality (LMI) is an inequality of a linear combination of matrix variables. For example, if A is an $n \times n$ constant matrix and Q is an $n \times n$ matrix variable, then the inequality $QA^T + AQ < 0$ is an LMI.

2 The Simplex Architecture and the Safety Region

The Simplex architecture is a software technology that supports safe, reliable, online software upgrade. A detailed description of the technology is given in the *Simplex Architecture Tutorial*.[2] Applications of the Simplex architecture in control systems are discussed in [Seto 98] and [Sha 97]. In this report, we will concentrate on the core functionality of the Simplex architecture—fault tolerance.

The fault tolerance in the Simplex architecture is based on the concept of analytic redundancy. The analytically redundant controllers are designed to take into account the upgrade of control algorithms. In particular, a highly reliable controller, the safety controller, is designed to work with the upgraded controller, the implementation of the upgraded control algorithm. When the upgraded controller is introduced to the system, it will take control of the physical plant, and the dynamic behavior of the plant will be monitored. The upgraded controller will continue to control the physical plant if the behavior of the plant is satisfactory with respect to some prescribed criteria. If the plant does not behave in a desired way, the upgraded controller may contain bugs. As a result, the upgraded controller will be disabled, and the safety controller will take over control to maintain the operation of the overall system. Then the upgraded controller will be taken offline to be investigated and repaired. After it is fixed, the upgraded controller will be reinserted into the system and will take back control of the physical plant. Such a cycle will be repeated until the reliability of the upgraded controller is the same as the reliability of the safety controller. In this way, we will have a highly reliable controller with the upgraded feature.

The fault tolerance in the Simplex architecture consists of two parts: fault detection and fault recovery. As mentioned earlier, fault detection is related to the switching criteria used when the control of the physical plant is switched from the upgraded controller to the safety controller, while fault recovery concerns the safety control, which prevents the plant from failing. Apparently, different faults may involve different detection mechanisms. In the *Simplex Architecture Tutorial*, Peter Feiler (of the Software Engineering Institute) summarizes the types of faults that the Simplex architecture can handle (namely, timing faults, semantic faults, and resource-sharing faults). In this report, we will focus on the semantic faults, which are faults caused by incorrect design and implementation of the upgraded control algorithm. This type

[2] The *Simplex Architecture Turorial* is available from Peter Feiler of the Software Engineering Institute, Pittsburgh, Pennsylvania.

of fault will cause malfunctioning in the physical plant and cause the plant to enter an unsafe state from which no control will be able to bring the plant back to normal operation. Eventually such a state will lead to physical damage. Therefore, the detection of semantic faults can be defined as the point where the upgrade controller is about to drive the physical plant into an unsafe state. In this sense, semantic fault detection becomes a safety check of the physical plant. Given that the safety controller will carry out the recovery once a semantic fault is detected, the safety check will depend on the control capability of the safety controller. In other words, for a given safety controller, the upgrade controller may contain a semantic fault if it is driving the physical plant to a state from which the safety controller can not bring the plant to normal operation. The safety of a physical plant with respect to the safety controller is defined precisely in [Seto 98], and we will review it in the remainder of this section.

A formal description of plant safety is based on a mathematical model of the plant. Let $x \in R^n$ be the *n*-dimensional state of the physical plant, and $u \in R^m$ be the *m*-dimensional control input to the plant. The class of physical plants that we are interested in can be described by the following state equations:

$$\dot{x} = f(x, u(x,t)) \qquad \text{with} \tag{1}$$

$$\text{state constraints: } q_1(x) \le 0, ..., q_l(x) \le 0\,, \tag{2}$$

$$\text{control constraints: } p_1(u) \le 0, ..., p_r(u) \le 0. \tag{3}$$

Definition 2.1: Given the plant in Equation (1) with the constraints in Equations (2) and (3),

1. A state *x* is admissible if it satisfies the constraints in Equation (2). The set of admissible states *F* is defined as $F = \{x : q_1(x) \le 0, ..., q_l(x) \le 0\}$.
2. A control input *u* is admissible if it satisfies the constraints in Equation (3). The set of admissible controls *G* is defined as $G = \{u : p_1(u) \le 0, ..., p_r(u) \le 0\}$.

The control law *u* can be either open loop or state feedback. The state and control constraints together give the physical constraints to the physical system, which are usually treated as hard constraints. The physical constraints reflect operating limits for physical devices or other considerations such as lack of sufficient knowledge to operate the physical system outside of these boundaries. The safety of the system is concerned with the operation of the physical system without violating the physical constraints. Soft constraints may also exist, reflecting regions within which certain desired control performance can be maintained. Violations of these performance-related limits do not necessarily threaten the safety or viability of the physical system, however. In this report, we focus on the class of systems in Equations (1)–(3) with hard physical constraints.

Example 2.1: Consider a simple mechanical system as shown in Figure 2.

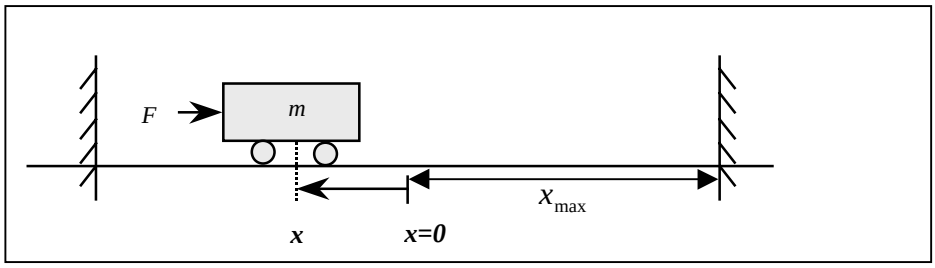

Figure 2: A Simple Mechanical System

Let $x_1 = x, x_2 = \dot{x}, u = F/m$. Then the equations of motion are given by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases} \text{ subject to } \begin{cases} |x_1| \le x_{max} \\ |u| \le F_{max}/m \end{cases}$$

The safety of the physical plant can be described with respect to a region in the state space where the safety controller can control the plant without violating the physical constraints. To characterize this region, we first define the operational region of a controller.

Definition 2.2: Consider the plant in Equations (1)–(3). An *operational region* (OR) for a given control law *u*, which takes values from *G*, is defined as a subset $O_u \subseteq F$ such that under the control of *u*, the trajectory of the plant, starting from any state in O_u, will remain in O_u and satisfy the control objective of *u*.

Since the safety controller is designed with the control objective of keeping the physical system from violating the physical constraints, the operational region of the safety controller can serve as a characterization of the plant safety. For instance, we could say that the plant is safe if its state is inside the OR of the safety controller; otherwise, it is unsafe. However, such a characterization can not be used as the switching criterion for the safety controller to take over. By the definition of the OR, it is clear that the control objective of a control law *u* may not be achieved if the physical plant starts from any state outside of the OR of *u*. Thus it would be too late for the safety controller to keep the plant from violating the physical constraints once the state of the physical system is out of its OR. To prevent this, we define a *restricted operational region* (ROR) as follows:

Definition 2.3: Given a plant in Equations (1)–(3), let *T* be the sampling period of the overall system and $\phi_v(t_0, x_0, t)$ be the solution of Equation (1) at $t \ge t_0$ with *v* the control input taking values from *G* and (t_0, x_0) the initial condition. A restricted operational region of the control law *u* is defined as a subset $R_u \subseteq O_u$,
$R_u = \{x : x \in O_u, \phi_v(t_0, x, t_0 + T) \in O_u, \forall t_0 \ge 0, \forall v \in G\}$.

Clearly, the restricted operational region contains all the states from which the state of the plant at the next sample will still be a point inside the corresponding operational region, no

matter what control is applied to the plant. Based on the definition of a restricted operational region, we define the notion of a safety region to characterize the safety of the plant.

Definition 2.4: Consider a plant given in Equations (1)–(3).

1. A *safety region* of a safety control law that takes values from *G* is defined as a restricted operational region of u_s (i.e., R_{u_s}). In addition, if all the trajectories of the plant can be driven to a subset $S \subseteq R_{u_s}$ by u_s, the safety region is said to be *recoverable* to *S*.

2. A given state of the physical system is *safe* with respect to a safety control u_s if it is inside R_{u_s}. Otherwise it is *unsafe*.

The safety region defined above may still not be conservative enough when there is one period delay in control implementation, in which case the control command computed based on the state at time *t* is sent to the physical plant at time *t*+*T*. To see this, we suppose that the state of the physical plant is detected to be out of the safety region at *t*. Although the safety controller will then be chosen to control the plant, its control command will not affect the physical plant until time *t*+*T*. At time *t*+*T*, however, the physical plant may have already evolved to a state out of the OR of the safety controller. Therefore, when the system involves one period delay in control implementation, the safety region of the safety controller u_s is further restricted as $R_{u_s} = \{x : x \in O_{u_s}, \phi_v(t_0, x, t_0 + 2T) \in O_{u_s}, \forall t_0 \geq 0, \forall v \in G\}$.

3 Lyapunov Stability Theory in Safety Control

When the physical plant involves equilibria or steady state, the safety of the plant can be characterized by the stability of the plant. In this case, the safety controller can be designed to maintain the stability of the physical plant, and the safety region can be defined as the stability region of the plant under the safety control. In this section, we first briefly review the Lyapunov stability theory, then formulate the safety-related issues as a stabilization problem. Most of the results in this section are well established in system and control literature, and we will simply state the results without proof. For details, readers can refer to a number of control texts (e.g., [Luenberger 79]).

3.1 Lyapunov Stability Theory

Before getting into the details of the Lyapunov stability theory, we will first give some definitions related to the stability of a dynamic system.[3] Here we consider a class of continuous-time autonomous dynamic systems described by the following equation:

$$\dot{x} = f(x(t)),\ x \in R^n \tag{4}$$

Definition 3.1: An equilibrium of the system in Equation (4) is a state x_e satisfying $f(x_e) = 0$.

Definition 3.2: Suppose x_e is an equilibrium state of a system in Equation (4). Then,

1. x_e is *stable* if for any $\varepsilon > 0$, there exists a δ, $0 < \delta < \varepsilon$, such that for all $x(t_0)$ satisfying $|x(t_0) - x_e| < \delta$, we have $|x(t) - x_e| < \varepsilon,\ \forall t > t_0$.
2. x_e is *asymptotically stable* if it is stable and $\lim_{t \to \infty} x(t) = x_e$.
3. x_e is *unstable* if it is not stable.

Definition 3.3: A dynamic system with an equilibrium state x_e is said to be *(asymptotically) stable* if x_e is a (an asymptotically) stable equilibrium. A *stability region S* of the system is

[3] In this and subsequent sections, we will often use the word "system" to refer to a plant whose dynamics can be described by a set of differential equations. This should not be confused with the system that we defined previously with respect to the overall computer-controlled system.

defined as a region in the system state space from which the system trajectories will stay inside a bounded region $B \supseteq S$. Furthermore, if *B = S, S* is called a *restricted stability region*.

Definition 3.4: A function *V(x)* defined in a neighborhood *U* of an equilibrium x_e of a system in Equation (4) is a Lyapunov function if it satisfies the following conditions:

1. *V* is continuous and has continuous first order partial derivatives;
2. x_e is the unique minimum of *V(x)* with respect to all other states in *U*;
3. The time derivative $\dot{V}(x) \leq 0, \forall x \in U$.

The above definitions have clear physical implications. The definition of equilibrium state implies that, once the system is at an equilibrium, it will stay there forever. For a dynamic system with a stable equilibrium, if the system starts close to the equilibrium, it will remain close to the equilibrium for all future time. Furthermore, if the equilibrium is asymptotically stable, the trajectory of the system will tend to the equilibrium as time increases. The stability region clearly characterizes the states, starting from which the system will be maintained close to the equilibrium, or will converge to the equilibrium. In safety control, we are interested in the stability region with an asymptotically stable equilibrium. Finally, the definition of the Lyapunov function represents an analogy to the energy dissipation process with minimum energy at the equilibrium point. Figure 3 illustrates some of the definitions.

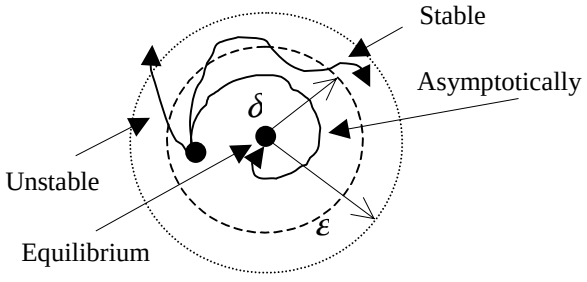

Figure3.a Illustration of stable, asymptotically stable, and unstable equilibrium.

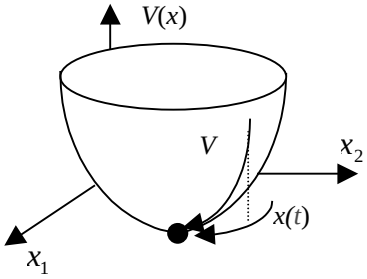

Figure 3.b Illustration of a Lyapunov function.

Figure 3: Illustrations of Stability-Related Definitions

Theorem 3.1 (Lyapunov Stability Theorem): For a dynamic system in Equation (4) with an equilibrium x_e , if there exists a Lyapunov function *V(x)* in a neighborhood *U* of x_e , then the equilibrium x_e is stable. Furthermore, if the time derivative $\dot{V}(x)$ is strictly negative everywhere in *U* except x_e , the equilibrium is asymptotically stable.

The Lyapunov stability theorem addresses two issues. First, for any given dynamic system with an equilibrium, if a Lyapunov function can be constructed with respect to the equilib-

rium, then a conclusion of the system stability (i.e., the system is stable or asymptotically stable about the equilibrium) can be made. However, finding a Lyapunov function is a sufficient condition for system stability. In other words, it can not be concluded that the system is unstable if no Lyapunov function has yet been found. Second, a stability region can be obtained from a Lyapunov function. Suppose there exists a Lyapunov function *V(x)* in a neighborhood *U* of an equilibrium of a given system. Then the Lyapunov function theorem implies that there exists a positive constant *c* such that the region defined by $S = \{x : V(x) \le c,\ x \in U\}$ is a stability region. It is worthwhile to note that the stability region defined in this way is not unique, and the set *S* with the largest *c* would give the largest stability region defined by this particular Lyapunov function. Since the time derivation of the Lyapunov function is always non-positive, the stability region defined by a Lyapunov function will be restricted. Thus, in the rest of this report, we will simply use stability region in the restricted sense when we derive the stability region from a Lyapunov function.

As a subclass of the systems in Equation (4), linear time-invariant (LTI) systems are of special interest. Numerous results related to this class of systems have been well established. In the next few paragraphs, we will show how the Lyapunov stability theorem is applied to this type of system. This class of system is given by the following equation:

$$\dot{x} = Ax,\ \ x \in R^n \tag{5}$$

Theorem 3.2: An LTI system in Equation (5) is asymptotically stable at the equilibrium $x = 0$ if and only if all the eigenvalues of matrix *A* are in the left half complex plane.

Definition 3.5: A system in Equation (5) is *quadratically stable* at the equilibrium $x = 0$ if there exists a positive definite matrix *P* such that the quadratic function $V(x) = x^T Px$ has negative derivatives along all the trajectories of Equation (5).

Theorem 3.3: A system in Equation (5) is asymptotically stable at the equilibrium $x = 0$ if and only if it is quadratically stable.

The equivalence of asymptotic stability and quadratic stability enables the systematic study of Lyapunov stability in LTI systems. Specifically, the construction of a Lyapunov function is narrowed to quadratic forms; however, such quadratic Lyapunov functions always exist as long as the LTI system is asymptotically stable. In other words, the existence of a quadratic Lyapunov function is a necessary and sufficient condition for the system to be asymptotically stable. To apply the Lyapunov stability theorem in an LTI system, we consider a quadratic function of state variables given by $V(x) = x^T Px$, where *P* is a positive definite matrix, denoted by $P > 0$.[4] We will show the conditions under which *V(x)* qualifies as a Lyapunov function, and therefore, the system is asymptotically stable. Apparently, any function *V(x)*

[4] Function *V(x)* is also called a *positive definite function* in the sense that $V(x) > 0,\ \forall x \neq 0$

just defined satisfies Conditions 1 and 2 in the definition of a Lyapunov function. To check the third condition in the definition, we differentiate *V(x)* along the system trajectory, and obtain the following:

$$\dot{V} = x^T (A^T P + PA)x$$

Then $\dot{V} < 0$ implies that the matrix $A^T P + PA < 0$. Hence we conclude that the system in Equation (5) is asymptotically stable if and only if there exist matrices *P* or $Q = P^{-1}$ such that

$$P > 0, \quad A^T P + PA < 0 \quad or \quad Q > 0, \ QA^T + AQ < 0 \tag{6}$$

This is also known as a feasible problem [na2]in the context of LMI. Namely, the LMIs in Equation (6) are feasible if there exist matrices *P* or $Q = P^{-1}$ satisfying Equation (6). Moreover, a system in Equation (5) is asymptotically stable if and only if LMIs in Equation (6) are feasible. This translates a stability problem to an LMI problem which can be solved by the interior-point methodology. We will discuss the solutions to this type of LMI problem in Section 4.

3.2 A Stabilization Problem

In the previous subsection, the Lyapunov stability theory was presented for a class of autonomous systems. In this subsection, we will apply the theory to control systems described in Equations (1)–(3). Specifically, we will concentrate on safety control [na3]since it is responsible for maintaining the safety of the physical plant, a crucial functionality in the Simplex architecture. As mentioned earlier, the safety of the physical plant can be characterized by the stability of the plant when there is an equilibrium in the set of admissible states. Namely, guaranteeing the safety of a plant is equivalent to maintaining stability of the plant when the plant is operating around an equilibrium; thus, a safety region can be defined as a stability region. In this sense, the safety controller can be designed to stabilize the plant around the equilibrium, and a corresponding stability region is derived as the safety region. A formal problem statement is given below. Again, consider a class of plants

$$\dot{x} = f(x,u) \text{ with } q_i(x) \le 0, \ i = 1,...,l, \text{ and } p_j(u) \le 0, \ j = 1,...,r \tag{7}$$

Suppose there is a unique equilibrium (x_e, u_e) defined by

$$f(x_e, u_e) = 0, \ \ q_i(x_e) \le 0, \ \ \forall i = 1,...,l, \text{ and } p_j(u_e) \le 0, \ \forall j = 1,...,r.$$

Then the control objective is to design a state feedback control law *u(x(t))* with $u(x(t)) \in G, \ \forall t > t_0$, such that the closed-loop system $\dot{x} = f(x, u(x))$ is asymptotically stable

at x_e. Furthermore, find the largest stability region of the closed-loop system contained in the set of admissible states.

The problem posed [na4]is a stabilization problem, and the solution can be obtained from the Lyapunov stability theory. With the control law designed in a state feedback form, we conclude that the closed-loop system is an autonomous system,[5] and the Lyapunov stability theory introduced in previous subsection can be applied directly. It is not trivial, however, to solve a nonlinear stability problem. Except for a small subclass of systems (for instance, systems that can be linearized by state feedback), most of the problems do not have known analytic solutions. Even though there are analytic solutions to some of the problems, they may not be generalized to other problems. To develop a systematic approach for control design and stability region derivation, we adopt the standard scheme to deal with nonlinear systems (namely, linearizing the nonlinear system at the equilibrium state), and then solve the problems with the linearized system. Specifically, let $\delta x = x - x_e$, and $\delta u = u - u_e$. Then expanding function *f(x,u)* by Taylor expansion and keeping only the first order terms, we get the following:

$$\dot{\delta x} = A\delta x + B\delta u \qquad \text{where } A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{\substack{x=x_e \\ u=u_e}} \text{ and } B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{\substack{x=x_e \\ u=u_e}}$$

are constant matrices. This transforms the nonlinear stabilization problem to a linear one. In the next section, we present several LMI-based approaches to solve the linear stability problem.

[5] It is not necessary for the control law to be state feedback, and it could be an open control loop (for instance, a big-bang control). If the control depends on time explicitly, the controlled system is no longer autonomous. Nevertheless, in this report, we will focus on the class of system in Equation (7) with state feedback control law *u(x)*.

4 Stability Analysis with LMI-Based Approaches

In the previous section, we defined the safety control as the control that stabilizes the plant at the equilibrium and characterized the safety region as a stability region of the plant under safety control. In this section, we present LMI-based approaches to solve the linear stabilization problem. In particular, we first formulate the problem in an LMI form, and then solve it for two different cases: (1) Derive the stability region for a given safety controller, and (2) design the safety controller and derive the corresponding stability region. Finally, we discuss further improvements of the presented LMI approaches. The fundamental concept and basic schemes used in this section are described in detail by Boyd et al in [Boyd 94].

As we discussed earlier, the stabilization problem will be solved for a class of linear time-invariant systems, which could be linearized approximations of the physical plants. Suppose this class of LTI systems is described as follows:

$$\dot{x} = Ax + Bu \text{ with constraints: } a_i^T x \le 1, \quad i = 1,...,l \text{ and } b_j^T u \le 1, \ j = 1,...,r \tag{8}$$

where $x \in R^n$ is a vector of state variables, $u \in R^m$ is a vector of control inputs, and $a_k \in R^n$ and $b_j \in R^m$ are constant vectors. Clearly, the equilibrium state *x=0* is a point in the set of admissible states. The control objective is to design a linear state feedback control in the form $u = Kx$ such that the closed-loop system is in an asymptotically stable state at the equilibrium. Moreover, the controlled system will evolve in a feasible region in the state space, where no constraints will be violated. This implies that the stability region of the closed-loop system will be restricted by the constraints. With the control law $u = Kx$, the closed-loop system is written as follows:

$$\dot{x} = \overline{A}x \text{ with constraints } \alpha_k^T x \le 1, \quad k = 1,..., p \tag{9}$$

where $\overline{A} = A + BK$, $\alpha_k = a_k$, $k = 1,..,l$, $\alpha_k^T = b_j^T K$, $j = 1,..,r, k = l + j$, $p = l + r$. According to the Lyapunov stability theory, the system in Equation (9) is asymptotically stable if and only if there exists a matrix P (or $Q = P^{-1}$) such that

$$P > 0, \quad \overline{A}^T P + P\overline{A} < 0 \quad or \quad Q > 0, \ Q\overline{A}^T + \overline{A}Q < 0 \tag{10}$$

Then a stability region *S* of Equation (9) can be defined as follows:

$$S = \{x : x^T P x \le 1\} \tag{11}$$

In addition, all the trajectories of the closed-loop system in Equation (9), starting from states in *S*, will satisfy the constraints if the stability region satisfies the constraints (i.e., $\alpha_k^T x \le 1 \ \forall x \in S,\ k = 1,..., p$). The following Lemma casts the constraints in an LMI form.

Lemma 4.1: Given an LTI system with the constraints in Equation (9), the stability region *S* defined in Equation (11) satisfies the constraints in Equation (9) if and only if $\alpha_k^T P^{-1} \alpha_k \le 1,\ k = 1,..., p$.

Proof: By definition, *S* satisfies the constraints if and only if $\alpha_k^T x \le 1 \ \forall x \in S,\ k = 1,..., p$. This is equivalent to $\max_{x \in S} \alpha_k^T x \le 1,\ k = 1,..., p$. Next we will show $\max_{x \in S} \alpha_k^T x = \sqrt{\alpha_k^T P^{-1} \alpha_k}$, $\forall k = 1,..., p$, which implies the Lemma. To this end, we solve the following nonlinear programming problem for each *k* = 1,...,*p*:

$$\text{maximize} \quad \alpha_k^T x \quad \text{subject to} \quad x^T P x \le 1$$

Let x^* be the optimal solution. Then x^* satisfies the Kuhn-Tucker conditions:

$$\alpha_k - 2\lambda P x^* = 0,\ \ \lambda(1 - x^{*T} P x^*) = 0,\ \ \lambda \ge 0$$

Apparently, there is a solution to x^* only if $\lambda > 0$. Solving the above equations, we obtain the following:

$$x^* = \left(P^{-1}\right)^T \alpha_k \Big/ \sqrt{\alpha_k^T P^{-1} a_k} \ \Rightarrow \ \max_{x \in S} \alpha_k^T x = \alpha_k^T x^* = \sqrt{\alpha_k^T P^{-1} a_k}$$

Then we conclude that $\max_{x \in S} \alpha_k^T x \le 1$ if and only if $\alpha_k^T P^{-1} \alpha_k \le 1$ for all *k*=1,...,*p*.

We now complete the transformation of a linear stabilization problem to a feasible problem with the following summary: The plant is stabilizable (i.e., it can be stabilized at the equilibrium without violating the constraints), if there exists a matrix *P* (or $Q = P^{-1}$) such that the following LMIs are satisfied:

$$\begin{cases} P > 0; \\ \overline{A}^T P + P\overline{A} < 0; \\ \alpha_k^T P^{-1} \alpha_k \le 1,\ k = 1,...p. \end{cases} \quad or \quad \begin{cases} Q > 0; \\ Q\overline{A}^T + \overline{A} Q < 0; \\ \alpha_k^T Q \alpha_k \le 1,\ k = 1,...p. \end{cases} \tag{12}$$

In the above feasible problem, the solutions to *K* and *P* (or *Q*) are not unique. In fact, there are an infinite number of *K* such that the control $u = Kx$ will stabilize the plant as long as all the eigenvalues of $\overline{A}$ are in the left half of the complex plane. In addition, for each *K*, there may be an infinite number of stability regions defined in Equation (11) satisfying the constraints. Given that a stability region is derived as a safety region, and the larger the safety region is, the more freedom an upgraded controller may have to explore new functionalities, we will be interested in the largest safety region. This leads to two different cases that will be investigated next: (1) Find the largest stability region with a given safety controller, and (2) design the safety controller such that the resulting stability region is maximized.

4.1 Stability Region with a Given Controller

In this case, we derive the safety region of the plant controlled by a given controller (i.e., $u = Kx$ with *K* given). This is the case when the safety control design and the safety region derivation are carried out separately. The safety control could be designed by some methods other than LMI, for instance, the linear quadratic regulation (LQR) technique or pole placement method, when some performance specifications need to be satisfied. It could also be the control algorithm that has been used in the past and has been proven reliable. Given that the stability region defined in Equation (11) is not unique, we are interested in deriving the largest *S* subject to the constraints. Since each stability region geometrically defines an ellipsoid in the state space of the plant, the size of a stability region is referred to as the volume of the ellipsoid. Hence the stability region in this case will be derived by solving an optimization problem: Maximize the volume of the ellipsoid subject to the constraints.

Since the control gain *K* is given, matrix $\overline{A}$ is completely determined, and the optimization problem is solved over all feasible matrices *Q* subject to LMI constraints in Equation (12). Since the volume of an ellipsoid given by $S = \{x: \ x^T Px \le 1\}$ is proportional to $\sqrt{\det P^{-1}}$, maximizing the volume is equivalent to minimizing the determinant $\det Q^{-1}$. Hence, a complete LMI problem for the optimization can be formulated as follows: For a dynamic plant $\dot{x} = \overline{A}x$ with constraints $\alpha_k^T x \le 1, \quad k = 1,..., p$, find the matrix *Q* that

$$
\begin{array}{ll}
\text{minimizes} & \log\det Q^{-1} \\
\text{subject to} & Q > 0; \\
 & Q\overline{A}^T + \overline{A}Q < 0; \\
 & \alpha_k^T Q \alpha_k \le 1, \quad k = 1,..., p.
\end{array}
$$

This problem is solved by Vandenberghe et al in [Vandenberghe 98], and a software implementation of the algorithm was developed by Wu and Boyd [Wu 97].[6] The following example illustrates the derivation of the stability region using the SDPSOL[7] software.

Example 4.1: Consider the simple mechanical plant given in Example 2.1. Suppose the physical parameters are given the following values:

$m = 1\,\text{kg},\ x_{\max} = 2\,\text{meters},\ \text{F}_{\max} = 1\,\text{Newton}.$

In addition, the safety controller is designed with the control gain K = [-2, -3].

Then the dynamics of the closed-loop plant is described by $\dot{x} = \bar{A}x$ with

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ \bar{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix},\ \text{and constraints}: \begin{cases} |x_1| \le 2 \\ |u| \le 1,\ or\ |-2x_1 - 3x_2| \le 1 \end{cases}$$

and the stability region is specified by $S = \{x:\ x^T Q^{-1} x \le 1\}$ with Q a 2×2 symmetric matrix to be determined. Then the LMI problem is formulated as

$$\begin{aligned} &\text{minimize} && \log\det Q^{-1} \\ &\text{subject to} && Q > 0; \\ & && Q\bar{A}^T + \bar{A}Q < 0; \\ & && \alpha_k^T Q \alpha_k \le 1,\ \ k = 1, \ldots 4, \end{aligned}$$

where $\alpha_1^T = [1/2,\ 0],\ \alpha_2^T = [-1/2,\ 0],\ \alpha_3^T = [-2,\ -3]$, and $\alpha_4^T = [2,\ 3]$. Solving this problem, we obtain the Q matrix as

$$Q = \begin{bmatrix} 4.0 & -2.6686 \\ -2.6686 & 1.8915 \end{bmatrix}$$

and the stability region displayed in Figure 4. (Note: The dashed lines in Figure 4 indicate the constraints.)

[6] The software can be downloaded via anonymous ftp <http://www.stanford.edu/~boyd/sdpsol/> and <http://www.stanford.edu/~boyd/maxdet/>.

[7] SDPSOL is a parser/solver for semidefinite programming and determinant maximization problems with matrix structure.

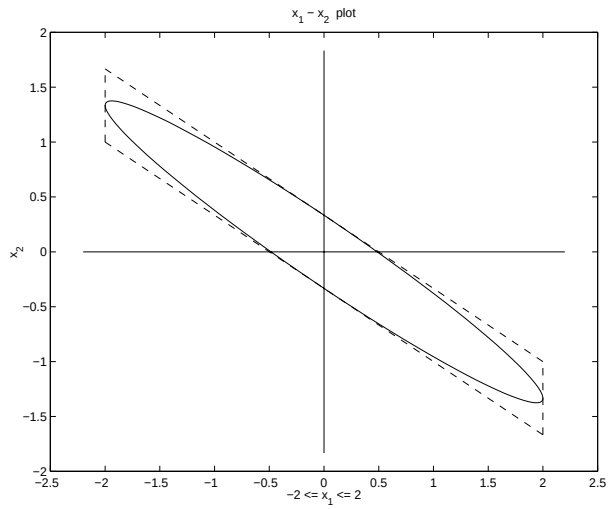

Figure 4: The Stability Region (Solid Line) of the Closed-Loop Plant in Example 4.1 with Control Law $u = -2x_1 - 3x_2$

4.2 Design of the Safety Controller

In this case, we design the safety controller and construct the corresponding stability region. This is the case when the control gain *K* and the matrix *P* (or *Q*) are determined jointly. We solve an optimization problem over all possible *K* and *P* (or *Q*) subject to the constraints such that the resulting closed-loop plant is asymptotically stable and the corresponding stability region is maximized. The stability region obtained in this case will be the largest one given by a quadratic Lyapunov function with respect to all possible *K*s that render asymptotic stability in the physical plant. Since the control gains are unknown in this case, and the choice of them will be restricted by the control constraints, we consider the dynamics systems given in Equation (8). Substituting $\overline{A} = A + BK$ in Equation (12), we obtain the following:

$$QA^T + AQ + QK^T B^T + BKQ < 0$$

By introducing the change of variable $Z = KQ$, the above condition becomes

$$QA^T + AQ + Z^T B^T + BZ < 0$$

and the constraints $b_j^T u \le 1 \Rightarrow b_j^T Kx \Rightarrow b_j^T KQK^T b_j \le 1 \Rightarrow b_j^T ZQ^{-1}Z^T b_j \le 1$, where the second step is the result of Lemma 4.1 and the third step is due to the change of variable. Using the Schur complements, we convert the last inequality to an LMI form as follows:

$$\begin{bmatrix} 1 & b_j^T Z \\ Z^T b_j & Q \end{bmatrix} \ge 0, \quad j = 1,...,r$$

Then the LMI problem can be formulated as follows: For the dynamic plant $\dot{x} = Ax + Bu$ with control law $u = Kx$, and the constraints $a_i^T x \le 1,\ \ i = 1,...,l$ and $b_j^T u \le 1,\ j = 1,...,r$, find Q and Z that

$$\begin{aligned}
&\text{minimizes} && \log\det Q^{-1} \\
&\text{subject to} && Q > 0; \\
&&& QA^T + AQ + Z^T B^T + BZ < 0; \\
&&& a_i^T Q a_i \le 1,\ \ i = 1,...,l; \\
&&& \begin{bmatrix} 1 & b_j^T Z \\ Z^T b_j & Q \end{bmatrix} \ge 0,\ \ j = 1,...,r.
\end{aligned}$$

Again, this problem can be solved by the approach developed in [Vandenberghe 98] and the SDPSOL software. Applying the change of variable, we obtain the control gain $K = ZQ^{-1}$.

In some plants, not only is the state constrained, but also the rates of change of state. Such constraints are often called *rate limits*. In this report, we consider the rate limits in the form $c_k^T \dot{x} \le 1,\ k \in \{1,...,n\}$, and translate them to an LMI as follows:

$$\begin{aligned}
&c_k^T \dot{x} \le 1 \Rightarrow (c_k^T A + c_k^T BK)x \le 1 \Rightarrow\ (c_k^T A + c_k^T BK)Q(c_k^T A + c_k^T BK)^T \le 1 \\
&\Rightarrow\ (c_k^T A + c_k^T BZQ^{-1})Q(c_k^T A + c_k^T BZQ^{-1})^T \le 1 \Rightarrow (c_k^T AQ + c_k^T BZ)Q^{-1}(c_k^T AQ + c_k^T BZ)^T \\
&\Rightarrow \begin{bmatrix} 1 & c_k^T AQ + c_k^T BZ \\ (c_k^T AQ + c_k^T BZ)^T & Q \end{bmatrix} \ge 0
\end{aligned}$$

Therefore, the LMI problem for optimization involving rate limits can be stated as follows: For the dynamic plant $\dot{x} = Ax + Bu$ with control law $u = Kx$ and the constraints $a_i^T x \le 1$, $i = 1,...,l$, $b_j^T u \le 1,\ j = 1,...,r$, and $c_k^T \dot{x} \le 1,\ k = 1,...,q$, find Q and Z that

$$\begin{aligned}
&\text{minimizes} && \log\det Q^{-1} \\
&\text{subject to} && Q > 0; \\
&&& QA^T + AQ + Z^T B^T + BZ < 0; \\
&&& a_i^T Q a_i \le 1,\ \ i = 1,...,l; \\
&&& \begin{bmatrix} 1 & b_j^T Z \\ Z^T b_j & Q \end{bmatrix} \ge 0,\ \ j = 1,...,r; \\
&&& \begin{bmatrix} 1 & c_k^T AQ + c_k^T BZ \\ (c_k^T AQ + c_k^T BZ)^T & Q \end{bmatrix} \ge 0,\ k = 1,...,q.
\end{aligned}$$

Example 4.2: To illustrate the design of the safety controller together with the derivation of the corresponding stability region, we consider Example 2.1 again. In this case, the dynamics of the plant are described by $\dot{x} = Ax + Bu$ with

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \text{and constraints}: \begin{cases} |x_1| \le 2 \\ |u| \le 1 \end{cases}$$

Let Q be a 2×2 symmetric matrix and Z be a 2×1 matrix. Then the LMI problem is formulated as follows: Find Q and Z that

$$\begin{array}{ll} \text{minimizes} & \log\det Q^{-1} \\ \text{subject to} & Q > 0; \\ & QA^T + AQ + Z^T B^T + BZ < 0; \\ & a_i^T Q a_i \le 1,\ \ i = 1,\ldots,l; \\ & \begin{bmatrix} 1 & b_j^T Z \\ Z^T b_j & Q \end{bmatrix} \ge 0,\ \ j = 1,\ldots,r. \end{array}$$

where $\alpha_1^T = [1/2,\ 0]$, $\alpha_2^T = [-1/2,\ 0]$, $b_1 = 1$, and $b_2 = -1$. Solving this problem using SDPSOL software, we obtain Q and Z as

$$Q = \begin{bmatrix} 4.0 & -1.2408 \\ -1.2408 & 3.4641 \end{bmatrix} \text{ and } Z = [-1.1547,\ -1.0746]$$

which determine the control gain: $K = ZQ^{-1} = [-0.433,\ -0.4653]$, and the corresponding stability region as depicted in Figure 5. (Note: The dashed lines in Figure 5 indicate the constraints of the plants.)

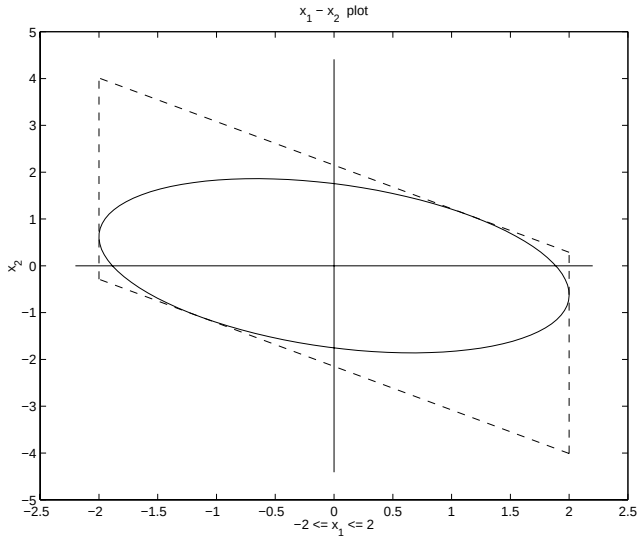

Figure 5: The Stability Region (Solid Line) for the Designed Safety Controller in Example 4.2

We now make a comparison of the controllers designed in this subsection (referred to as the *designed controller*) and the one given in the previous subsection (referred to as the *given controller*). As mentioned earlier, the designed controller results in the largest stability region of the closed-loop system with respect to all the possible control laws for stabilizing linear state feedback. Figure 6 shows that its corresponding stability region is indeed larger than the one obtained from the given controller. In addition, the performance of the physical plant under the two controllers is also different. The simulation results in Figure 6 show that, in terms of the convergence rate, the performance of the plant under the given controller is much better than when it is controlled by the designed controller, when the plant starts from the state $[x_1, x_2] = [1.0, -0.8]$ in both cases. The comparisons of the stability region and the closed-loop system performance reveal a general tradeoff for linear state feedback control laws; namely, the size of the stability region and the performance of the closed-loop system are inversely related. This is an important point in the concept of analytic redundancy with respect to the controller design in the Simplex architecture. Specifically, since the safety controller is responsible for providing protection, it should be designed so that the upgraded controller can explore new functionality in a large domain of the state space. Therefore, the primary goal in the safety controller design is to make its operational region as large as possible, and the secondary concern may be to increase the performance it yields. On the other hand, the baseline controller serves as the complement of the safety controller, so its performance should be the first priority, and its operational region becomes a minor issue. In summary, in the examples that we considered, the designed controller can serve as the safety controller, and the given controller can be used as the baseline controller. An extensive analysis of the tradeoff was given in the case study on the inverted pendulum control system; see [Seto 99a].

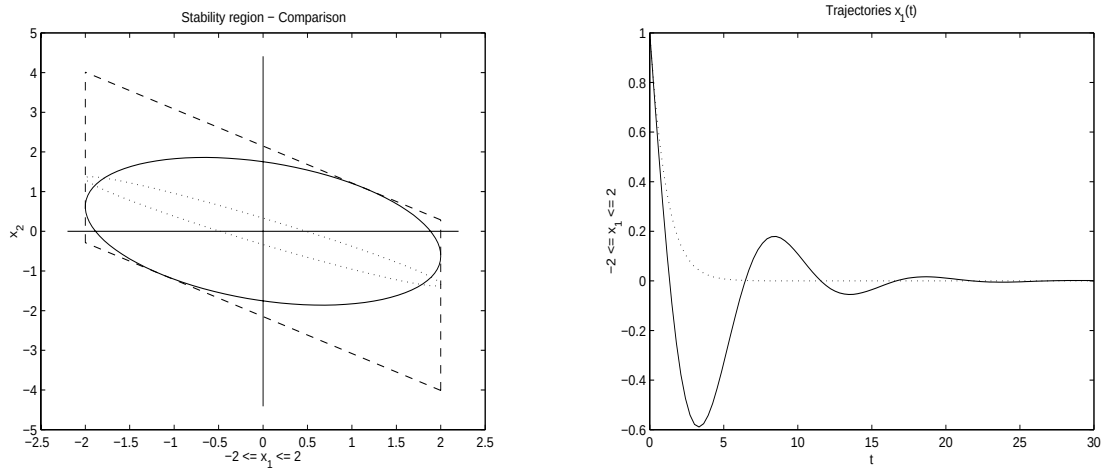

Figure 6: Comparisons of the Stability Region and Performance of the Plant Under the Designed Controller and the Given Controller[8]

The above comparisons also motivate a general design strategy for the safety controller and baseline controller from an existing control algorithm, which has been used in the past. In particular, starting from the existing control algorithm, by adjusting the parameters in the algorithm such that the operational region is enlarged, we may get a safety controller; by adjusting the parameters to improve the performance of the controlled system, we will obtain a baseline controller. If the existing control algorithm is linear state feedback, the adjustment of the control gains can be carried out systematically using the LMI approaches that we have proposed.

4.3 Further Improvements on Safety Control Design

Previously we have seen that the safety controller may result in low performance in the closed-loop system. Such a reduced performance may not be acceptable in some systems because the recovery by the safety controller may take too much time. In this subsection, we will show how to improve the performance with respect to some performance specifications. In addition to designing the safety controller to maximize the corresponding stability region subject to the constraints, we also require the closed-loop system to satisfy the given specifications. The specifications imposed on the performance should be moderate so that the corresponding stability region remains a reasonably large size.

The specification that we will consider in this subsection is the closed-loop pole location. Depending on how the specification is given, it can have various effects on performance (the decay rate, the natural frequencies, etc.). Not only will the performance of the closed-loop system be affected by the pole location, but the shape of the resulting stability region will change as well. We will present a general approach developed by Chilali and Gahinet

[8] In Figure 6, solid lines represent the result obtained from the designed controller, and dotted lines show the result generated by the given controller. Again, the dashed lines indicate the constraints.

[Chilali 96] to incorporate the specification into an LMI problem. Refer to [Seto 99b] for some examples in aircraft control.

Definition 4.1: An LMI region is defined as a subset *L* of the complex plane *C*, described by

$$L = \{z : z \in C,\ f_L(z) < 0\}$$

where $f_L(z) = \Phi + z\Psi + \bar{z}\Psi^T$, and $\Phi = \Phi^T \in R^{m\times m}, \Psi \in R^{m\times m}$.

Theorem 4.1: Given an LTI system in the form $\dot{x} = Ax$, the system is asymptotically stable with poles in an LMI region *L* if and only if there exists a symmetric matrix *Q* such that

$$M_L(A, Q) < 0, \quad Q > 0$$

where $M_L(A, Q) = \Phi \otimes Q + \Psi \otimes (AQ) + \Psi^T \otimes (AQ)^T$, and $\otimes$ denotes Kronecker product.

Corollary 4.1: Given an LTI control system $\dot{x} = Ax + Bu$ with control law $u = Kx$, the system is asymptotically stable with all the poles in an LMI region *L* if and only if there exist a symmetric matrix *Q* and a matrix *Z* with proper dimensions such that

$$M_L(A, Q, Z) < 0, \quad Q > 0$$

where $M_L(A, Q, L) = \Phi \otimes Q + \Psi \otimes (AQ + BZ) + \Psi^T \otimes (AQ + BZ)^T$. Moreover, the control gain is determined by $K = ZQ^{-1}$.

Theorem 4.1 and Corollary 4.1 give the LMI conditions for the system, with or without control, to be asymptotically stable with the specified pole location. When the system involves constraints, additional LMI constraints such as we presented in the previous subsection should be considered. Most of the often-used pole location specifications can be cast as LMI regions defined in Definition 4.1 and incorporated into the LMI conditions for stability. For example, suppose the poles of a completely controllable system $\dot{x} = Ax + Bu$ are required to be inside a disk of radius *r* and center (*-d*, 0), $d > 0$, in the complex plane. Let a complex pole be denoted by $z = x + jy$. Then the specified region in the complex plane is given by $(x + d)^2 + y^2 < r^2$, or $(z + d)(\bar{z} + d) < r^2$ because $x = (z + \bar{z})/2,\ x^2 + y^2 = z\bar{z}$. Applying Schur complements, we obtain the LMI region given as

$$f_L(z) = \begin{bmatrix} -r & z+d \\ \bar{z}+d & -r \end{bmatrix} = \begin{bmatrix} -r & d \\ d & -r \end{bmatrix} + z\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \bar{z}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \Phi + z\Psi + \bar{z}\Psi^T$$

and the LMI conditions for stability as

$$M_L(A,Q) = \begin{bmatrix} -rQ & dQ \\ dQ & -rQ \end{bmatrix} + \begin{bmatrix} 0 & AQ \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ QA^T & 0 \end{bmatrix} = \begin{bmatrix} -rQ & AQ + dQ \\ QA^T + dQ & -rQ \end{bmatrix} < 0,\ Q > 0$$

Another important specification is related to the decay rate, namely the rate of the trajectories of the closed-loop system converging to the equilibrium. Such a rate requirement can be translated to the pole location by making all the poles of the closed-loop system located at the left side of the vertical line $x = -d$, $d > 0$. Then all the trajectories of the closed-loop system will converge to the equilibrium at rates no less then d. The specification of the pole location in this case is given by $x < -d$, or $f_L(z) = 2d + z + \bar{z} < 0$. Then the LMI conditions in Theorem 4.1 are given by

$$M_L(A,Q) = 2dQ + AQ + QA^T < 0,\ \ Q > 0$$

Incorporating the constraints on pole location into the stabilization problem will improve the performance of the closed-loop system. This has been demonstrated in a case study on an aircraft auto-landing control system [Seto 99b]. An extensive study on pole placement in the context of LMI is also reported in [Chilali 96].

5 Conclusions

In this report, we addressed the semantic fault tolerance issue in the Simplex architecture. Fault detection and recovery were established with respect to the safety of the physical system under control. Specifically, faults are detected by checking the safety of the physical plant against a predefined safety region, and the recovery is guaranteed by the safety controller. When the physical plant is operated around an equilibrium, the safety controller is designed to stabilize the system at the equilibrium, and the safety region is defined as the stability region of the physical plant under the safety controller. By linearizing the plant at the equilibrium, a linear approximation of the plant is obtained. Based on this linear model of the plant, several LMI-based approaches are presented to (1) systematically derive the largest stability region of the plant under a given controller and (2) systematically design the safety controller and derive the corresponding safety region. Figure 7 shows a flow chart of this complete procedure for developing the semantic fault tolerance mechanism using LMI.

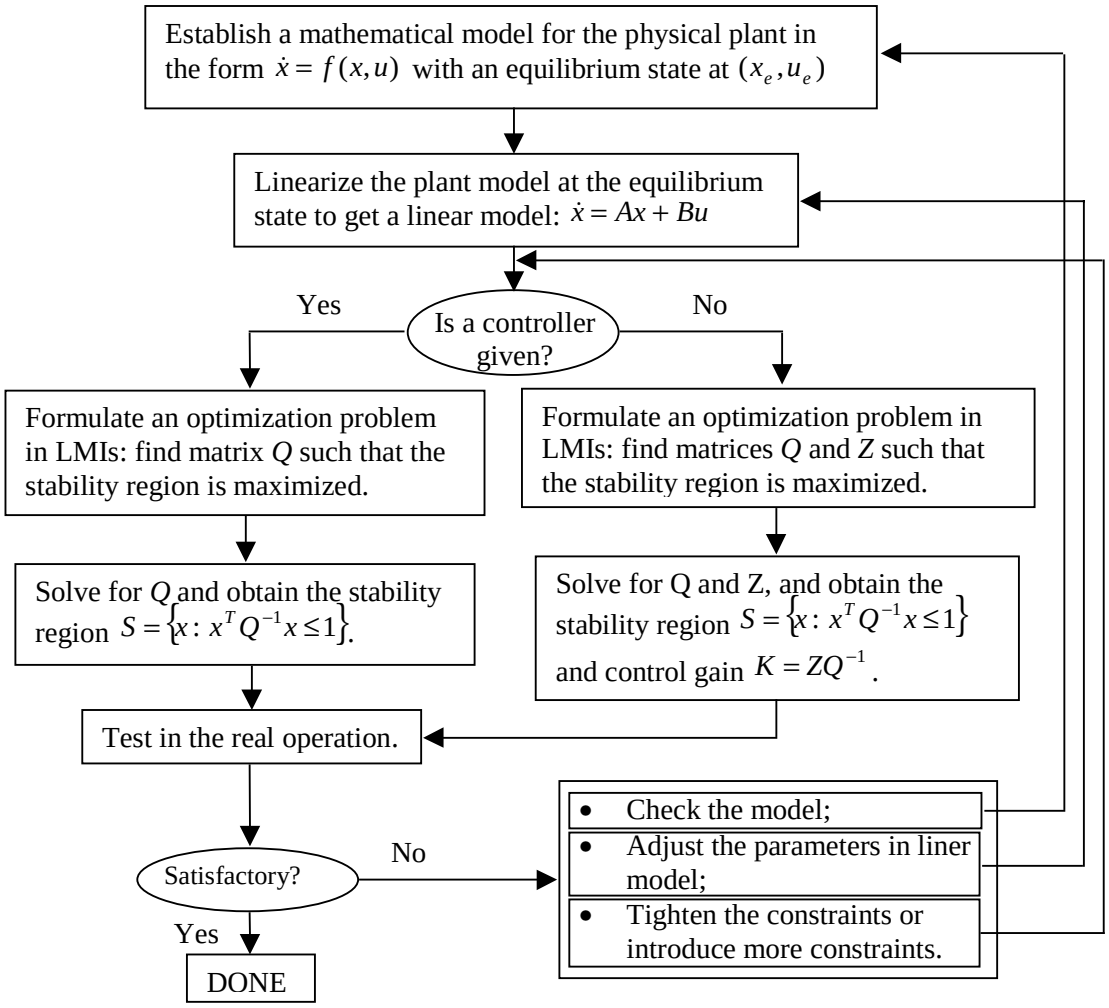

Figure 7: A Development Cycle for Semantic Fault Tolerance Mechanism Using LMI

References

[Boyd 94] Boyd, S.; El Ghaoui, L.; Feron, E.; and Balakrishnan, V. "Linear Matrix Inequalities in System and Control Theory." *SIAM Studies in Applied Mathematics*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1994.

[Chilali 96] Chilali, M. and Gahinet, P. "Design with Pole Placement Constraints: an LMI Approach." *IEEE Transactions of Automatic Control*, Vol. 41 (March 1996).

[Luenberger 89] Luenberger, D. G. *Introduction to Dynamic Systems, Theory, and Applications*. New York, NY: John Wiley & Sons, 1979.

[Seto 98] Seto, D.; Krogh, B. H.; Sha, L.; and Chutinan, A. "Dynamic Control System Upgrade Using the Simplex Architecture." *IEEE Control System Magazine* (August 1998).

[Seto 99a] Seto, D. and Sha, L. *A Case Study on Analytical Analysis of the Inverted Pendulum Real-Time Control System* (CMU/SEI-99-TR-023). Pittsburgh, PA: Software Engineering Institute, 1999.

[Seto 99b] Seto, D. and Ferreira, E. *A Case Study on Development of a Baseline Controller for Automatic Landing of an F-16 Aircraft Using LMIs* (CMU/SEI-99-TR-020). Pittsburgh, PA: Software Engineering Institute, 1999.

[Sha 97] Sha, L. "Sifting the Computation Paradigm of Real-Time Control Systems." (invited paper). *Proceedings of the Real-Time Computing Symposium, (SNART97)*, Lund, Sweden, August 1997.

[Vandenberghe 98] Vandenberghe, L.; Boyd, S.; and Wu, S.-P. "Determinant Maximization with Linear Matrix Inequality Constraints." *SIAM Journal on Matrix Analysis and Application* Vol. 19 (1998).

[Wu 97] Wu, S.-P. and Boyd, S. *SDPSOL: Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure - User's Guide, beta version*. Available WWW <URL: http://www.stanford.edu/~boyd/sdpsol/> (1999).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	**2.** REPORT DATE **August 1999**	**3.** REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE **An Engineering Method for Safety Region Development**		**5.** FUNDING NUMBERS C — F19628-95-C-0003
6. AUTHOR(S) **Danbing Seto, Lui Sha**		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		**8.** PERFORMING ORGANIZATION REPORT NUMBER **CMU/SEI-99-TR-018**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/DIB 5 Eglin Street Hanscom AFB, MA 01731-2116		**10.** SPONSORING/MONITORING AGENCY REPORT NUMBER **ESC-TR-99-018**
11. SUPPLEMENTARY NOTES		
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		**12.B** DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

In this report, we study tolerance of semantic faults, one of the crucial issues in the Simplex™ architecture. In particular, we examine semantic faults that cause the controlled device to be unsafe (i.e., unable to carry out its normal operation) and eventually cause the device to become damaged. We also consider fault detection as a safety check. For the class of control systems operating around an equilibrium, the objective of maintaining the safety of the controlled device is formulated as a stabilization problem, and the safety of the controlled device is tested against the stability region of the device under the safety control. To establish the stability region, we apply the Lyapunov stability theory and linear matrix inequality (LMI) methodologies. It is shown that the stability region for a given safety controller as well as a safety control law can be systematically derived by LMI-based approaches. We conclude the report with a summary of the procedure for deriving the safety check and safety controller for a given application.

14. SUBJECT TERMS **linear matrix inequality (LMI) methodology , Lyapunov stability theory, semantic faults, Simplex™ architecture, stability analysis**			**15.** NUMBER OF PAGES **28** **16.** PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18.** SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	**19.** SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	**20.** LIMITATION OF ABSTRACT UL