Nonlinear Systems Control

Using a Subset Stabilization Approach

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Adam T. Simmons

Certificate of Approval:

Jitendra K. Tugnait
Professor
Department of Electrical and Computer Engineering

A. Scottedward Hodel, Chair
Associate Professor
Department of Electrical and Computer Engineering

John Y. Hung
Associate Professor
Department of Electrical and Computer Engineering

Stephen L. McFarland
Acting Dean
Graduate School

Nonlinear Systems Control
Using a Subset Stabilization Approach

Adam T. Simmons

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
May 11, 2006

Nonlinear Systems Control

Using a Subset Stabilization Approach

Adam T. Simmons

Signature of Author

Date of Graduation

Vita

Adam Terry Simmons was born March 10, 1978, the second child of D. Terry and Beth Simmons. In his hometown, Rome, Georgia, he attended St. Mary's Catholic School through the third grade. From fourth grade until High School Graduation, he attended Pepperell Public Schools located in Lindale, Georgia. At Pepperell, Adam graduated with Honors and an Advanced Technical/Science degree.

After graduating from High School, he attended his grandfather's alma mater, Mercer University, a small Baptist University located in Macon Georgia. At Mercer he was a member of Phi Delta Theta, placed on the Freshman Advisory Board and given membership to several Honor Societies (i.e. Order of Omega, Phi Sigma Tau). After graduating from Mercer with a B.S. in Engineering, he worked in a small software firm for 6 months, after which he decided to attend Auburn University to continue his education.

Dissertation Abstract

Nonlinear Systems Control Using a Subset Stabilization Approach

Adam T. Simmons

Doctor of Philosophy, May 11, 2006
(M.S., Auburn University, 2003)
(B.S., Mercer University, 2001)

95 Typed Pages

Directed by A. Scottedward Hodel

Numerous problems in nonlinear control systems are difficult or impossible to solve with commonly proposed methods of design. For such systems, we propose a two-loop control strategy that we call the subset stabilization approach (SSA), that combines the strengths of nonlinear control design with the flexibility of robust control design. We present several control system examples for which standard control system techniques either fail or perform poorly with respect to regions of convergence. We then apply our proposed method to these systems and compare results. These simulation examples establish that the SSA method provides a relatively simple but powerful method for the design of nonlinear control systems. Further, these observations are bolstered by a theoretical analysis that establishes the principles upon with the success of the SSA is based.

Acknowledgments

I must first give my thanks to Jesus Christ for all that He has done and will continue to do in my life.

My undying gratitude and love go to my parents, Terry and Beth Simmons for their financial and emotional support throughout my life. Without their religious guidance and continuous love, I would not be the person I am today.

I would like to thank my committee. They have been patient and helpful. I especially would like to thank Dr. A. Scottedward Hodel, my graduate advisor and mentor; his vast experience and knowledge, along with a truly devout religious character were instrumental in my completion of this research. His understanding over the past 5 years has been limitless and he has my profound gratitude. Dr. John Y. Hung also deserves my thanks due to his constant push for me to approach problems in unique ways, and this research would not have existed without the persistence of such a teacher, researcher and mentor.

The completion of my degree would have been infinitely more difficult without the prayers, support and advice of my sister, Tracy, and her husband, Chris Small.

Finally, good friends provided much needed prayers, encouragement and accountability, like Rob Daily, Matthew Hollimon, Kim Jackson, Joe Jacobs, Karen Miller, Jill Moore, Will & Kim Travis and others.

This work was funded by a NASA Graduate Student Research Fellowship from NASA-NGT8-52904.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliograpy follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `auphd.sty`.

Table of Contents

List of Figures x

1 Introduction 1

2 Nonlinear Control 2
- 2.1 Lyapunov Control Design . . . 3
- 2.2 Input-Output Linearization . . . 5
- 2.3 Input-State Linearization . . . 11
- 2.4 Integrator Backstepping . . . 16

3 Ball On Beam Example 20
- 3.1 Review of the Linear control . . . 22
- 3.2 Lyapunov Control Design Method . . . 23
- 3.3 Input-Output Linearization Method . . . 24
- 3.4 Input-State Linearization Method . . . 25
- 3.5 Underlying Issue: Poorly Defined Relative degree . . . 26
- 3.6 Dynamic Analysis of the Ball On Beam System . . . 27
- 3.7 Concluding Remarks . . . 31

4 New Subset Stabilization Approach (SSA) Control Design 33
- 4.1 Subset Stabilization Approach . . . 33
- 4.2 Five Case Studies . . . 35
  - 4.2.1 System 1 . . . 35
  - 4.2.2 System 2 . . . 39
  - 4.2.3 System 3 . . . 41
  - 4.2.4 System 4 . . . 42
  - 4.2.5 System 5 . . . 44
- 4.3 Theoretical analysis . . . 48
- 4.4 Conclusion . . . 54

5 Conclusions 55
- 5.1 Design Advantages . . . 55
- 5.2 SSA for Nonlinear Control . . . 56

6 Future Work 57
- 6.1 Experimental Validation of the Ball-on-beam Control Effort . . . 57
  - 6.1.1 Subset Stabilization Approach to [21] . . . 59
  - 6.1.2 Comparison of State Equations and Control Solution . . . 61
- 6.2 Analytical Stability . . . 63

Bibliography 64

Appendices 66

A Five Systems Examples,Code: 67

B Code for Five System Phase Portrait 76

C Ball On Beam Example Code/Diagrams 79
C.1 Block Diagram For Ball-on-beam Model 79
C.2 S-Function for Nonlinear Plant 80
C.3 Feedback M File 83

List of Figures

2.1 Inner loop of input-output linearizing control, and equivalent model 8

2.2 Input-output relationship 10

2.3 Inner loop of input-state linearizing control, and equivalent model 11

2.4 Integrator Backstepping Example Block Diagram 17

2.5 Integrator Backstepping Example Block Diagram, Part 2 18

3.1 Ball on Beam Diagram 21

3.2 Comparison of a Modified System with the Original System 28

3.3 Block Diagram of Simplified ball-on-beam System 28

3.4 Stability of x_3 and x_4 confirmed 29

3.5 x_1 Response with nonlinear control, u_{nl} 30

3.6 x_3 and x_4 Responses with nonlinear control, u_{nl} 31

3.7 x_1 Response with u_{nl} and $u = -Kx$ control inputs 32

3.8 x_1 Response with u_{nl} and Hauser's control inputs 32

4.1 Example System 1: State responses to LQR feedback and SSA control 38

4.2 Example System 2: State responses to LQR feedback and SSA control 40

4.3 Example System 3: State responses to LQR feedback and SSA control 43

4.4 Example System 4: State responses to LQR feedback and SSA control 45

4.5 Example 5 (ball-on-beam control): State responses to LQR feedback and SSA control 47

4.6 Subset Stabilization Approach Block Diagram 49

4.7 Subspace stabilized system phase portrait. The integer parameter n in equation (4.15) is varied from 0 (a) to 1 (b) to 2 (c). Observe that with increased values of n the phase portrait tends to more closely approach the null subset manifold $x_2 = 0$. 52

4.8 Region of Convergence, Comparison of SSA-based control and LQR control efforts for System 2 53

6.1 Uran and Jezernik Ball on Beam Apparatus 58

6.2 Uran and Jezernik System Simulation with SSA control Solution 61

Chapter 1

Introduction

All engineering disciplines model the dynamics of complex systems through mathematical equations. These models represent a system through a set of either linear or nonlinear differential equations. Linear models approximate simple dynamic characteristics and are widely used in undergraduate engineering classes. Conversely, nonlinear models approximate complex dynamic characteristics. In either representation, using the most accurate model of a system will provide the most complete understanding of that system; system understanding is the cornerstone of control systems engineering.

Control of nonlinear systems has found increased usage within industrial electronics applications such as robotics, electric motors, and power electronics. Some of the most well-known techniques include Lyapunov stability design [13], input-output linearization [11], input-state linearization [2], and integrator backstepping [1]. For some nonlinear models these techniques are either excessively difficult to apply or result in an infinite control effort at specific operating points, i.e. singularities. These techniques are further limited by structural model requirements.

In this dissertation we present a brief overview of nonlinear control techniques, several example control systems, and a new approach to nonlinear systems control. In Chapter 2 four traditional nonlinear control techniques are reviewed: Lyapunov control design, input-output linearization, input-state Linearization, and integrator backstepping. A well-known example, the ball-on-beam system, is presented in Chapter 3. In Chapter 4, five other example systems are used to present a new two-loop control strategy that we call the subset stabilization approach (SSA). Finally, conclusions and future work are presented in Chapters 5 and 6, respectively.

Chapter 2

Nonlinear Control

In this chapter we provide background related to nonlinear systems and nonlinear systems control. We present four common nonlinear control techniques: Lyapunov control design, input-output linearization, input-state linearization, and integrator backstepping. Nonlinear control techniques are typically used to achieve better closed loop system properties, e.g., region of stability, performance, robustness, etc., than that achievable through the use of linear control techniques. Linear control design techniques do not directly address the nonlinearities of a plant, and good performance is restricted to a region about the nominal design condition. Nonlinear techniques utilize more mathematical information about the modeled plant and, therefore, provide a control solution with a larger region of asymptotic stability.

Our analysis will require the following definitions.

Definition 1 We say that a point $\bar{x} \in \mathbb{R}^n$ is an *equilibrium point* of the system $\dot{x} = f(x)$ if $f(\bar{x}) = 0$. Without loss of generality, we assume the equilibrium point $\bar{x}$ is at the origin.

Definition 2 We say that $\bar{x}$ is a *stable* (*attractive*) equilibrium point for $\dot{x} = f(\bar{x})$ if, for each small value of $\epsilon > 0$ there is a $\delta = \delta(\epsilon) > 0$ such that $||\bar{x}|| < \delta \Rightarrow ||x(t)|| < \epsilon, \forall t \geq 0$. We can further say $\bar{x}$ is *asymptotically stable* if it is stable and the value of δ is chosen such that $||\bar{x}|| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0$. If $\bar{x}$ is not *stable*, then $\bar{x}$ is an *unstable* equilibrium point.

Definition 3 A *vector field* is a construction in vector calculus that associates a vector with every point in a Euclidean space.

Definition 4 Consider functions $h\colon D \to \mathbb{R}$ and $f\colon D \to \mathbb{R}^n$. The *Lie Derivative* of h with respect to f at a point $x \in D$ is defined by

$$L_f h(x) = \nabla_x h^T f(x)$$

The Lie derivative is analogous to the derivative of h along the trajectories of the system $\dot{x} = f(x)$.

2.1 Lyapunov Control Design

We here consider the application of Lyapunov's stability theory in the design of control laws for nonlinear systems. Recently, Lyapunov control design methods have been used to explore stabilizing control efforts for system with constrained inputs [7, 12] and time delayed systems [13]. Lyapunov control methods have also been applied to a variety of complex mechanical systems [14], systems with discontinuous dynamics [20], and aerospace applications [15].

Lyapunov's stability theory is summarized concisely below.

Theorem 1 *Let $\bar{x} = 0$ be an equilibrium point of a system described by $\dot{x} = f(x)$ in a domain $D \subset \mathbb{R}^n$ containing the origin. Let $V\colon D \to \mathbb{R}$ be a continuously differentiable function such that*

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in } D - \{0\}$$

and

$$\dot{V}(x) \leq 0 \text{ in } D$$

Then $\bar{x} = 0$ is a stable equilibrium point of the system $\dot{x} = f(x)$. Moreover, if

$$\dot{V}(x) < 0 \text{ in } D - \{0\}$$

then $\bar{x} = 0$ is an asymptotically stable equilibrium point of $\dot{x} = f(x)$.

A proof of Theorem 1 is given in [6].

Consider the system $\dot{x} = f(x, u)$ where x represents the states and u is a control input and suppose it is desired to regulate the state x to a target value $\bar{x}$. Lyapunov stability theory can be applied to the design of a control $u = g(x)$ as follows. We select a candidate Lyapunov function,i.e., $V(x)$ to be a scalar-valued positive definite function of the state vector x. A common choice is the total energy of the system state, e.g. $V(x) = x^T x$. The derivative $\dot{V}(x)$ of $V(x(t))$ with respect to t is given by

$$\dot{V}(x) = (\nabla_x V) f(x, u) \tag{2.1}$$

or represented as a Lie Derivative

$$\dot{V}(x) = L_f V$$

Lyapunov-based control design involves the selection of a feedback control $u = g(x)$ that will guarantee a negative definite time derivative (2.1) in a region containing the desired equilibrium point $\bar{x}$.

Example 1 *Consider the system*

$$\begin{aligned} \dot{x}_1 &= x_1 x_2 + x_2^2 \\ \dot{x}_2 &= x_1^2 + u \end{aligned} \tag{2.2}$$

We select the Lyapunov function $V(x_1, x_2) = \frac{1}{2}\left(x_1^2 + x_2^2\right)$ *which has the corresponding derivative with respect to time*

$$\dot{V}(x_1, x_2, \dot{x}_1, \dot{x}_2) = x_1 \dot{x}_1 + x_2 \dot{x}_2$$

where $\dot{x}_2$ *is a function of the control input* u *and the states* x_1 *and* x_2*.*

$$\dot{V}(x_1, x_2, u) = 2x_1^2 x_2 + x_1 x_2^2 + x_2 u$$

We select the control input u to guarantee $\dot{V}(x_1, x_2)$ is negative definite, hence stable using Lyapunov's definitions. One possible choice is

$$u = -2x_1^2 - x_1x_2 - x_1^2x_2 - x_2$$

resulting in negative definite function $\dot{V}(x_1, x_2) = -x_1^2x_2^2 - x_2^2$. This gives a closed loop stable system using Lyapunov's definition of stability, in the introduction of this chapter.

2.2 Input-Output Linearization

Input-output linearization is a feedback control method where the result is a direct and simple relation between a system's output and control input. Input-output linearization is used for time-delayed single input single output nonlinear system's control [11]. Singh *et al* used input-output linearization to develop adaptive flight controls [17, 16] and more recently, input-output linearization is being used to develop state observers for nonlinear systems [5, 22].

Consider a nonlinear system

$$\dot{x} = f(x, u) \tag{2.3a}$$

$$y = h(x) \tag{2.3b}$$

Input-Output linearization attempts to linearize the input-output relationship 2.3a at the expense of producing nonlinear output functions. For example, consider a system described by

$$\begin{aligned} \dot{x}_1 &= x_2^2 \\ \dot{x}_2 &= -x_1 + u \\ y &= x_1 \end{aligned}$$

with an output $y = x_1$. A change of variables

$$z_1 = x_1, \quad z_2 = x_2^2$$

and the selection of a feedback control $u = x_1 - \frac{v}{2x_2}$ yields closed loop dynamics

$$\begin{aligned} \dot{z}_1 &= z_1 && (2.4a) \\ \dot{z}_2 &= v && (2.4b) \\ y &= \sqrt{z_2} && (2.4c) \end{aligned}$$

The state transformation from x to z produces linear state equations but the output measurement (2.4c) is now a nonlinear function of the state. As a result, traditional linear control approaches cannot be applied to this revised system.

A generalized input-output linearization for single-input-single-output (SISO) systems is described below. Consider the SISO plant

$$\begin{aligned} \dot{x} &= f(x) + g(x)u && (2.5) \\ y &= h(x) && (2.6) \end{aligned}$$

with state vector $x \in \mathrm{IR}^n$, scalar input u, and scalar plant output y. Input-output linearization decomposes the input u into a nonlinear inner loop control $u(x, v) = \alpha(x) + \beta(x)v$ that creates a linear dynamic relationship between the output y and a user defined input v [6, 19]. Performance of the input-output linearized system is controlled by linear feedback system design methods applied to the new input v. The new linear model state variables are the output y of (2.6) and its derivatives $\frac{d^i y}{dt^i}$, $i = 1, \cdots, n-1$. Because of this structure, input-output linearization is well suited for tracking control problems.

A general form of input-output linearization follows below. Differentiate the system output function (2.6) until the plant input u appears explicitly in the final differentiation. The general form for the i^{th} time derivative of the output can be expressed using Lie derivative notation

$$y^{(i)} = L_f^{(i)} h + (L_g L_f^{(i-1)} h) u \tag{2.7}$$

The i^{th} time derivative of the output y does not have any explicit dependence on the plant input u for a system with relative degree r, where $i < r$. An explicit dependence on the input u appears in the r^{th} derivative of the output. That is,

$$L_g L_f^r h \neq 0$$

The input $u(t)$ is expressed as a gain scheduled linear affine function of an external input v. Selecting the plant input as

$$\begin{aligned} u(x,v) &= \alpha(x) + \beta(x) v \\ &= \frac{1}{L_g L_f^r h}[-L_f^{r-1} h + v] \end{aligned} \tag{2.8}$$

a linear dynamic relationship between the input variable v and the system output y is achieved. The dynamic relationship between v and y is then described by the linear model

$$y^{(r)} = \frac{d^r y}{dt^r} = v \tag{2.9}$$

The inner loop for input-output linearizing control is illustrated by the block diagram in Fig. 2.1. The equivalent system is shown in the dashed block.

The input-output linear system (2.9) is stabilized by designing an outer loop control law for the new input v by any linear control design method. The applicability of linear design methods is one reason that input-output linearization is very attractive [10]. Additionally,

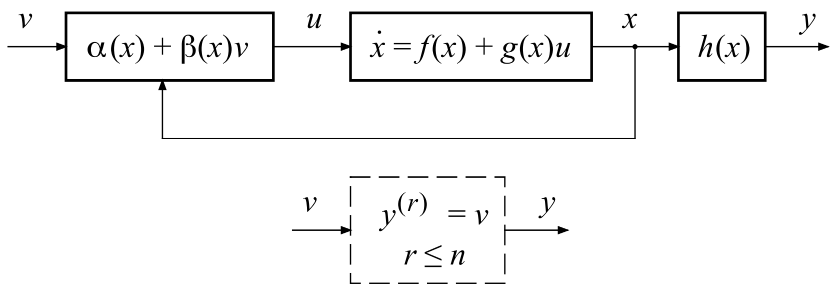


Figure 2.1: Inner loop of input-output linearizing control, and equivalent model

the straightforward computation of the time derivative functions (2.7) makes input-output linearization attractive.

There are difficulties related to input-output linearization. Canceling nonlinearities by mathematical means may produce singularities in any control solution. More precisely,

Definition 5 A system has a *singularity* at x_0 if an input-output linearization feedback law $u(x, v)$ satisfies $\lim_{x\to x_0}$u(x,v)$= \infty$. The feedback law $u(x, v)$ is also considered *singular* at x_0.

Example 2 *The control law $u(x, v) = \frac{1}{x} + v$ is singular at $x = 0$.*

Singularities appear in the design of input-output linearization control laws (2.8) at state values where $L_g L_f^r h$ is zero.

A second issue that affects input-output linearization control law design is associated with the relative degree r of a system. The relative degree is defined below.

Definition 6 Consider the nonlinear state equations

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

where $f : D \to \mathbb{R}^n$,$g : D \to \mathbb{R}^n$,and $h : D \to \mathbb{R}$ are sufficiently smooth on a domain $D \subset \mathbb{R}^n$, the state equations are said to have *relative degree* r, $1 \leq r \leq n$, in a region $D_0 \subset D$ if [6]

$$\frac{\partial \psi_i}{\partial x} g(x) = 0, \quad i = 1, 2, \cdots, r-1; \quad \frac{\partial \psi_r}{\partial x} g(x) \neq 0$$

for all $x \in D_0$ where

$$\psi_1(x) = h(x) \text{ and } \psi_{i+1}(x) = \frac{\partial \psi_i}{\partial x} f(x), \quad i = 1, 2, \cdots, r-1$$

If the relative degree r of a system is a function of the operating point x or is not well defined over the state space, then singularities may occur in input-output linearization. That is, the control solution $u(x, v) = \alpha(x) + \beta(x)v$ will have a singularity at points x where the relative degree of (2.5)-(2.6) change. Therefore, input-output linearization is not a well defined solution for systems with poorly defined relative degree. Pseudo-linearization techniques may be able to circumvent the usual input-output linearization techniques difficulties [4, 8, 9, 10].

If the relative degree r of a system is equal to the order of the system,i.e., n then input-output linearization provides a linearized system from v to y. If the relative degree is less than number of states, i.e., $r < n$, then the system has a set of dynamics, called *internal dynamics* with an order $n - r$, that are not considered during input-output linearization. The system also has *zero dynamics*, which are defined as the internal dynamics of the system where the system output y is kept at zero by the input u [19, page 219]. The stability of the input-output linearized system (2.5)-(2.6) is only guaranteed if the internal and zero dynamics are stable. In general, the determination of the stability of the internal dynamics is very difficult due to nonlinearities and the fact they are coupled to the other dynamics.

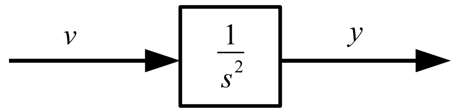


Figure 2.2: Input-output relationship

Example 3 *Consider a system*

$$\begin{aligned} \dot{x}_1 &= x_1 x_2 + x_2^2 \\ \dot{x}_2 &= x_1^2 + u \\ y &= x_1 \end{aligned} \tag{2.10}$$

We differentiate the output y until u appears explicitly in the result:

$$\begin{aligned} y &= x_1 \\ \dot{y} &= \dot{x}_1 = x_1 x_2 + x_2^2 \\ \ddot{y} &= x_2 \dot{x_1} + x_1 \dot{x_2} + 2 x_2 \dot{x_2} \\ &= x_1 x_2^2 + x_2^3 + x_1^3 + x_1 u + 2 x_2 x_1^2 + 2 x_2 u \end{aligned}$$

We incorporate an auxiliary (user-defined) input v into our control law and we select the input u as

$$u = \frac{-x_1 x_2^2 - x_2^3 - x_1^3 - 2 x_2 x_1^2 + v}{x_1 + 2 x_2}$$

to eliminate the plant nonlinearities. This process yields the input-output linearized form

$$\ddot{y} = v$$

as shown in Figure 2.2. This revised system is easily controlled by linear techniques. The final

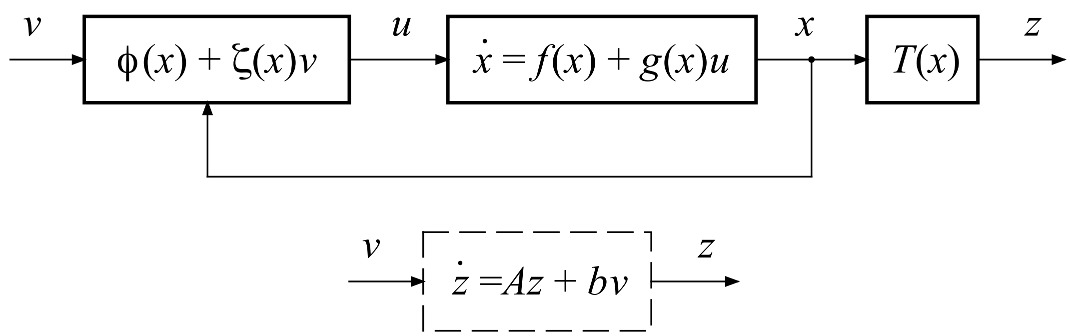


Figure 2.3: Inner loop of input-state linearizing control, and equivalent model

solution in this example can take the form

$$\begin{aligned} v &= -K \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ u &= \frac{-x_1 x_2^2 - x_2^3 - x_1^3 - 2x_2 x_1^2 + v}{x_1 + 2x_2} \end{aligned}$$

where K is a set of stability state feedback gains associated with Figure 2.2.

2.3 Input-State Linearization

Recall the single input single output (SISO) plant (2.5). Input-State linearization is achieved by the combination of a smooth, invertible state transformation $z = T(x)$ and a nonlinear inner loop control $u = \phi(x) + \zeta(x)v$, where v is a user defined input. The control u yields a linear relationship between the new input v and the new state z, as illustrated in Fig. 2.3. The equivalent linear input-state system is shown in the dashed block. The block labeled $T(x)$ represents the smooth (differentiable), invertible nonlinear transformation of the state x; the block labeled $\phi(x) + \zeta(x)v$ represents a nonlinear mapping from the new input v to the original input u. The result of input-state linearization is a model

$$\dot{z} = Az + bv \tag{2.11}$$

where the pair (A, b) is in controllable canonic form. An outer-loop control law for v is designed with the controllable canonic state variable model.

Two definitions will aid in the following discussion.

Definition 7 A linearly independent set of vector fields $\{f_1, f_2, \cdots, f_m\}$ is *involutive* if there are scalar functions $\alpha_{ijk}\colon \mathbb{R}^n \to \mathbb{R}$ such that

$$[f_i, f_j](x) = \sum_{k=1}^{m} \alpha_{ijk}(x) f_k(x), \quad \forall\ i, j$$

Definition 8 An *Ad field* is

$$ad_f g = [f, g] = \nabla_g f - \nabla_f g$$

There are two conditions that must be satisfied in order to apply input-state linearization. They are (1) the set of vector fields $\left\{g(x) \ \ ad_f g \cdots ad_f^{n-2} g\right\}$ (Lie Brackets) must satisfy

$$rank\left\{\left[g(x) \ \ ad_f g \ \ ad_f^2 g \cdots ad_f^{n-1} g\right]\right\} = n$$

and must be involutive over the field of scalar nonlinear functions in x. A positive test for involutivity exists if the Lie bracket of any two elements of a set of vector fields is spanned by the set. Both conditions, (1) and (2), must be satisfied for the existence of a linearizing transformation $T(x)$.

If a linearizing transformation $T(x)$ exists for the system $\dot{x} = f(x) + g(x)u$, then in fact an infinite number of such transformations exist. This property can be exploited to reduce the complexity of choosing a transformation. Let the transformation take the form

$$T(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ \vdots \\ T_{n-1}(x) \\ T_n(x) \end{bmatrix}$$

Recall that the pair (A, b) in (2.11) is assumed to be in controllable canonic form

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \\ \vdots & & & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Thus, $\dot{z}_i = z_{i+1}$ for $i = 1, \cdots, n-2$ and $\dot{z}_n$ will be a nonlinear function containing the input u. The transformation $T(x)$ from x to z is written recursively as

$$\begin{aligned} \dot{z}_1 &= T_1(x) = z_2 \\ &= (\nabla_{\bar{x}} T_1)\, \dot{x} = (\nabla_{\bar{x}} T_1)\, [f(x) + g(x)u] \\ \dot{z}_2 &= (\nabla_{\bar{x}} T_2)\, [f(x) + g(x)u] \\ &= (\nabla_{\bar{x}}\, (\nabla_{\bar{x}} T_1)\, [f(x) + g(x)u])\, [f(x) + g(x)u] \end{aligned}$$

All of the transformation elements $T_1 \cdots T_n$ are expressed recursively in terms of T_1. This is a direct result of the form of A. Due to the structure of b, $(\nabla_x T_j)\, g(x) = 0$ for all $j = 1, 2, \cdots, n-1$ and $(\nabla_x T_n)\, g(x) \neq 0$. Thus, the calculation of $T(x)$ is reduced to the calculation of $T_1(x)$ in

$$\nabla_x T_i g(x) = 0, \quad i = 1, 2, \cdots, n-1; \quad \nabla_x g(x) \neq 0$$

where

$$T_{i+1}(x) = \nabla_x T_i f(x), \quad i = 1, 2, \cdots, n-1$$

As with input-output linearization, input-state linearization has potential difficulties that increase the complexity of is control design or prevent it from providing a satisfactory control solution. The computation of the Lie brackets is extremely labor intensive for all but the simplest of plants. Symbolic processing by computers can be attempted in principle, but the rank conditions are often difficult to verify for all state space values. The development of the nonlinear state transformation $T(x)$, which requires the solution of n partial differential equations, is very labor intensive. The input-state linearizaiton control solution may have singularities within the operating region of the state space.

Example 4 *The system*

$$\begin{aligned} \dot{x}_1 &= a\sin(x_2) \\ \dot{x}_2 &= -x_1^2 + u \\ y &= x_1 \end{aligned}$$

is in the form required of input-state linearization, Figure 2.3

$$f(x) = \begin{bmatrix} a\sin(x_2) \\ -x_1^2 \end{bmatrix} \quad g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

We first determine the conditions under which the system is a valid input-state linearizable system. The vector field set $\{g(x), ad_f g(x)\}$ is examined for linear independence. The $ad_f g(x)$ component is given

$$ad_f g(x) = \nabla_x g(x) f(x) - \nabla_x f(x) g(x) = \begin{bmatrix} a\cos(x_2) \\ 0 \end{bmatrix}$$

and the field matrix

$$\{g(x), ad_f g(x)\} = \left\{ \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} , \begin{bmatrix} a\cos(x_2) \\ \\ 0 \end{bmatrix} \right\}$$

is only full rank when $x_2 \neq \pm\pi/2$. *Input-state linearization will only be valid for operating regions where* $|x_2| = \pi/2$.

We apply the above observations to compute recursively the state transformation. Recall that we must solve

$$\begin{aligned} (\nabla_x T_1)\, g(x) &= 0 \\ (\nabla_x T_1)\, ad_f g(x) &\neq 0 \end{aligned}$$

or

$$\begin{aligned} \frac{\partial T_1}{\partial x_2} &= 0 \\ -\frac{\partial T_1}{\partial x_1} a\cos(x_2) &\neq 0 \end{aligned}$$

for $T_1(x)$. *One possible solution is* $T_1(x) = x_1^2$. *We select this function for* $T_1(x)$ *and we obtain* $T_2(x)$ *as*

$$T_2(x) = (\nabla_x T_1(x))\, f(x) = \begin{bmatrix} 2x_1 & 0 \end{bmatrix} \begin{bmatrix} a\sin(x_2) \\ \\ -x_2 \end{bmatrix} = 2ax_1 \sin(x_2)$$

With the above definition of $T(x)$ *the input-state linearized model becomes*

$$\begin{aligned} \dot{z}_1 &= 2x_1\dot{x}_1 = 2ax_1\sin(x_2) = z_2 \\ \dot{z}_2 &= 2ax_1\cos(x_2)u + 2a(a - a\cos(x_2)^2 - x_1^3\cos(x_2)) \end{aligned}$$

We choose the modified system input to cancel the nonlinearities and introduce a new control v.

$$u = \frac{-2a^2 + 2a^2 \cos(x_2)^2 + 2ax_1^3 \cos(x_2) + v}{2ax_1 \cos(x_2)}$$

which yields the linear input-state relationship from v to z

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v \end{aligned}$$

2.4 Integrator Backstepping

The technique of integrator backstepping is applied to strict feedback systems with the following definition.

Definition 9 We say that a system is in *strict feedback form* or that a system is a *strict feedback system* if it can written as a cascaded set of state equations

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \qquad (2.12) \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \dot{x}_3 &= f_2(x_1, x_2, x_3) + g_2(x_1, x_2, x_3)x_4 \\ &\vdots \\ \dot{x}_n &= f_n(x) + g_n(x)u \end{aligned}$$

A block diagram representation of (2.12) appears as a set of nested loops.

Integrator backstepping is a recursive approach to control system design. A stabilizing control law is constructed for an inner state variable (e.g. x_1), treating the state variable from the next level (e.g. x_2) as a pseudo-input. The stabilizing control law for the inner variable is propagated to the input of the next state variable by differentiation. This process is called *backstepping*. Due to this process of repeated differentiation in the design of backstepping

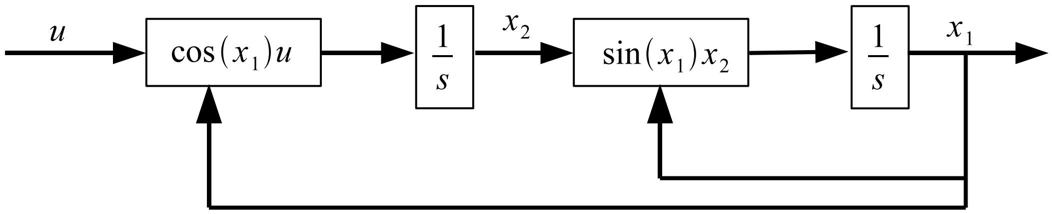


Figure 2.4: Integrator Backstepping Example Block Diagram

control laws, there is a tendency for the control laws to "blow up" or to contain singularities within a selected operating region of the state space.

Backstepping has advantages over the linearization techniques discussed in the previous sections. Unlike the previous two methods, the linearization of system dynamics is not an inherent result or requirement of the backstepping methodology. Thus, backstepping is arguably a more flexible design approach than input-output or input-state linearization. Nonlinearities that are not inherently harmful to a system's stability do not have to be mathematically canceled. The backstepping approach also satisfies Lyapunov stability theory at every iteration and, in the final result, provides an explicit Lyapunov function for an entire closed loop system.

Example 5 *Consider the strict feedback system*

$$\dot{x}_1 = \sin(x_1)x_2 \tag{2.13a}$$

$$\dot{x}_2 = \cos(x_1)u \tag{2.13b}$$

A block diagram for the system is shown in Figure 2.4. We choose the variable x_2 as the input to the subsystem (2.13a) *and the Lyapunov function*

$$V_1(x_1) = \frac{1}{2}x_1^2$$

The derivative of V_1 with respect to time is

$$\dot{V}_1(x_1, x_2) = x_1 \sin(x_1)x_2$$

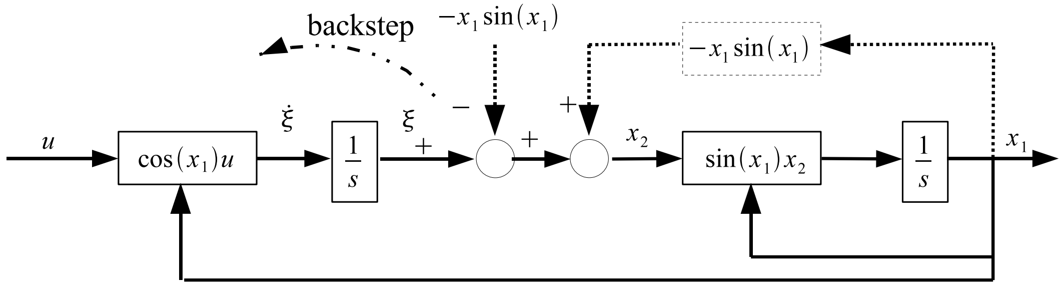


Figure 2.5: Integrator Backstepping Example Block Diagram, Part 2

We therefore select the desired state $x_2(t)$ function to be

$$x_2 = -x_1 \sin(x_1)$$

and add this to the input x_2, the same input was then subtracted from the input, Figure 2.5. A new variable ξ was defined by the user.

$$\xi = x_1 \sin(x_1) + x_2$$

and the desired control x_2 was backstepped through the integrator, Figure 2.5.

$$\dot{\xi} = \frac{d}{dt}\left(x_1 \sin(x_1)\right) + \cos(x_1)u \tag{2.14}$$

Let

$$V_2(x_1, \xi) = V_1(x_1) + \frac{1}{2}\xi^2$$

and the derivative of V_2 with respect to time is

$$\dot{V}_2(x_1, \xi) = \dot{V}_1(x_1) + \xi\dot{\xi}$$

The term $\dot{V}_1(x_1)$ is negative definite so we select $\dot{\xi} = -\xi$. We thus solve

$$-\xi = \frac{d}{dt}\left(x_1 \sin(x_1)\right) + \cos(x_1)u$$

for the external input u.

$$\begin{aligned} u &= \frac{1}{\cos(x_1)}\left[-\xi - \frac{d}{dt}\left(x_1 \sin(x_1)\right)\right] \\ &= \frac{1}{\cos(x_1)}\left[-\xi - \dot{x}_1\left(\sin(x_1) + x_1\cos(x_1)\right)\right] \\ &= \frac{1}{\cos(x_1)}\left[-x_1\sin(x_1) - x_2 - \sin(x_1)x_2\left(\sin(x_1) + x_1\cos(x_1)\right)\right] \end{aligned}$$

Observe that the final control solution is only valid for $|x_1| \leq \pi/2$.

Chapter 3

Ball On Beam Example

In this chapter a classical nonlinear control problem, the ball-on-beam problem, is used to illustrate weaknesses in the four nonlinear control design techniques presented in Chapter 2 and to demonstrate the utility of a newly proposed hierarchical control system design technique, the subspace stabilization approach (SSA). The SSA involves the selection of a subset of system states for an inner-loop nonlinear control law that is used to aid an outer loop robust control law. The ball-on-beam problem involves several design challenges that are not met by the design approaches of Chapter 2 but that are easily addressed by the SSA. In particular, the relative degree in the ball-on-beam problems is poorly defined, which renders infeasible such common nonlinear control methods as input-state linearization or input-output linearization.

Linear feedback control designed around a model linearization of the ball-on-beam often performs well near nominal operating conditions. Since LQR control design is an established technique with known robustness properties, we shall henceforth use LQR linear state feedback design for the outer loop control law in our treatment of the SSA. The reader should notice that this choice is made for convenience in presentation and that the SSA is not restricted to the use of LQR control for the outer loop robust control law design.

A diagram of the ball-on-beam system is shown in Figure 3.1. One well known mathemat-

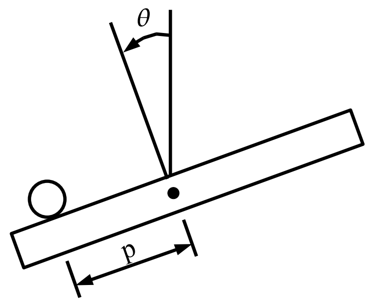


Figure 3.1: Ball on Beam Diagram

ical representation of this system is given by

$$\begin{aligned}
\dot{x_1} &= x_2 \\
\dot{x_2} &= \frac{m x_1 x_4^2}{\left(m + \frac{I_{ball}}{r^2}\right)} + \frac{mg\sin(x_3)}{\left(m + \frac{I_{ball}}{r^2}\right)} \\
\dot{x_3} &= x_4 \\
\dot{x_4} &= \frac{\tau - C_d x_4 + g m x_1 \cos(x_3)}{\left(I_{beam} + I_{ball} + m x_1^2\right)}
\end{aligned}$$

where x_1 is the position p of the ball on the beam (m), x_2 is the velocity v of the ball measured along the beam (m/s), x_3 is the angle θ of the beam relative to horizontal (rad), x_4 is the angular velocity of the beam (rad/s),τ is a moment applied to the beam by a controlling motor (N-m), r is the radius of the ball (m), g is a gravitational constant (m/s^2), m is the mass of the ball, I_{beam} is the inertia of the beam (kgm^2), I_{beam} is the inertia of the beam (kgm^2), and C_d is a friction term associated with the beams pivot (N). For clarity in exposition, we simplify these equations to be

$$\dot{x_1} = x_2 \tag{3.1a}$$

$$\dot{x_2} = x_1 x_4^2 + \sin(x_3) \tag{3.1b}$$

$$\dot{x_3} = x_4 \tag{3.1c}$$

$$\dot{x_4} = u \tag{3.1d}$$

where u is the control input. In spite of numerous studies in the literature, a control law that will stabilize the system (3.1a)-(3.1d) remains an active topic of research discussion [3, 4, 18].

The remainder of this chapter is organized as follows. We first review LQR control design techniques in Section 3.1. Following this, we apply four nonlinear control design techniques of Chapter 2 to the ball-on-beam problem in Sections 3.2–3.4, where we establish that *none* of these techniques provide a satisfactory control law design. The ball-on-beam system is not in strict feedback form, therefore integrator backstepping is not a valid design technique. In Section 3.5 we illustrate that one issue that causes these other methods to fail is the poorly defined relative degree of the ball-on-beam system. Following this, we illustrate our SSA method in Section 3.6, in which we demonstrate not only a satisfactory control law, but also a relatively simple design procedure by which this control law may be obtained. We defer a theoretical analysis of the SSA method to Chapter 4.

3.1 Review of the Linear control

We linearize (3.1a)-(3.1d) about the origin to obtain

$$\dot{x_1} = x_2 \tag{3.2a}$$

$$\dot{x_2} = x_3 \tag{3.2b}$$

$$\dot{x_3} = x_4 \tag{3.2c}$$

$$\dot{x_4} = u \tag{3.2d}$$

or

$$\dot{x} = Ax + Bu$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A state-feedback law $u = -Kx$ is found using the above two state matrices and a linear-quadratic regulator (LQR) design. LQR calculates the optimal gain matrix K such that the state-feedback law minimizes a cost function J with user defined weights, i.e. $Q = I_{4x4}, R = 10$.

$$\begin{aligned} \min \quad J &= \int x^T Q x \; dt \\ \text{subject to} \quad \dot{x} &= Ax + Bu \end{aligned}$$

This state-feedback control law is able to control the nonlinear plant (Equations 3.1a - 3.1d) for initial position values close to the origin ($x(0) = \mathbf{0}$).

3.2 Lyapunov Control Design Method

In the Lyapunov control design method the designer selects a positive definite Lyapunov function $V(x)$ The equations (3.1a) thru (3.1d) show that most positive definite energy-like functions $V(x)$ defined for all states will cause the resulting derivative $\dot{V}(x)$ to have the form

$$V(x) \quad = \quad f_2(x_1, x_2, x_3, x_4)$$

where

$$\dot{V}(x) = (\nabla_x f_2)^T \; \dot{x}$$

Due to the structure of the system (3.1a)-(3.1d) , Lyapunov-theory based control laws u will typically include terms of the form $1/x_4$, which creates a singularity in the control at $x_4 = 0$

(zero angular velocity). For example, consider the Lyapunov function $V(x) = \frac{1}{2}(x_1^2+x_2^2+x_3^2+x_4^2)$ whose derivative with respect to time is

$$\dot{V}(x) \;=\; x_1\dot{x_1} + x_2\dot{x_2} + x_3\dot{x_3} + x_4\dot{x_4}$$

We therefore select a control input u that is decomposed as

$$u = u_1(x) + u_2(x)$$

where $u_1(x)$ cancels the positive nonlinear terms and $u_2(x)$ is a negative definite function.

That is, we select the control solution

$$\begin{aligned} u_1(x) &= -\frac{x_1 x_2}{x_4} - \frac{x_2 \sin(x_3)}{x_4} - x_3 - x_4 x_2 x_1 \\ u_2(x) &= -x3^2 x_4 - x_1^2 x_4 - x2^2 x_4 - x_4 \end{aligned}$$

The control u_1 has a singular point at $x_4 = 0$. That is, the law fails at the desired operating point. This difficulty arises in every energy-based Lyapunov design we tested.

3.3 Input-Output Linearization Method

We apply input-output linearization to the ball-on-beam example as follows. The ball-on-beam output is $y = x_1$ and the input is found in Equation (3.1d) as u. The output is differentiated until the input explicitly appears, to obtain

$$
\begin{aligned}
y &= x_1 \\
\dot{y} &= \dot{x}_1 = x_2 \\
\ddot{y} &= \dot{x}_2 = x_1 x_4^2 + \sin(x_3) \\
\dddot{y} &= \dot{x}_1 x_4^2 + \dot{x}_4(2x_1x_4) + \dot{x}_3\cos(x_3) \\
&= x_2 x_4^2 + 2x_1x_4u + x_4\cos(x_3)
\end{aligned}
$$

The function $\dddot{y}$ is written more conveniently as $\dddot{y} = f_1(x) + ug_1(x)$. Solution $\dddot{y}$ for the control input u gives the control solution for input-output linearization, as

$$u = \frac{-f_1(x)}{g_1(x)} \tag{3.3}$$

If the denominator $g_1(x)$ of u is non-zero in the region of operation, then this control is valid within the region of operation. If the origin is within the region of operation, such as in the ball-on-beam system ($x = 0$ and $g_1(0) = 0$), then the control contains a singularity. For the ball-on-beam example, the origin is within the region of operation, making input-output linearization an infeasible control design method. Approximate input-output linearization is extensively discussed for the ball-on-beam system by Hauser et al [4]. Hauser et al neglect terms, i.e. $x_1x_4^2$, in the dynamics that produce singularities within the input-output linearization control solution.

3.4 Input-State Linearization Method

We apply input-state linearization to the ball-on-beam example as follows. As discussed in Section 2.3, a transformation $T(x)$ must exist for input-state linearization to be a valid method. One of two necessary and sufficient conditions for a transformation to exist requires the following set of vector fields (Lie brackets) to have full rank [19]. The vector field matrix

$$\text{Vec. Field} \;=\; \begin{bmatrix} 0 & 0 & 2x_1x_4 & v_1(x) \\ 0 & -2x_1x_4 & v_2(x) & v_3(x) \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} v_1(x) &= 4x_4x_2 - \cos(x_3) \\ v_2(x) &= -2x_4x_2 + \cos(x_3) \\ v_3(x) &= -2x_1x_4^3 - 2x_4\sin(x_3)\cdots \\ &\quad - \sin(x_3)x_4 - 2x_4^3x_1 \end{aligned}$$

the $rank \neq n$ when $x_3 = 90^\circ$ and $x_4 = 0$. The ball-on-beam system does not satisfy at least one of the necessary and sufficient vector field rank condition, so input-state linearization will not work for the this system. Hauser *et al.* show that the matrix $\left[g(x) \;\; ad_f g \;\; ad_f^2 g \cdots ad_f^{n-2} g\right]$ for the ball-on-beam equations is not involutive [4], ans so this system fails the second condition for the existence of $T(x)$.

3.5 Underlying Issue: Poorly Defined Relative degree

A major issue in all of these failed methods is the relation between the state x_4 and the relative degree of the system (3.1a)-(3.1d). The ball on beam system has a state dependent relative degree of 3, as seen from the result $g_1(x)$ in Equation (3.3) and the relative degree is not well defined at the origin (i.e. $x = 0$).

For the ball-on-beam equations (3.1a-3.1d), the linear control design methods produces the only "successful" control input u around the origin. We conclude that nonlinearities that are neglected in a linearized system model must be related to the poorly defined relative degree of the system. Examination of the system equations (3.1a)-(3.1d) and the linearized equations (3.2a)-(3.2d) reveals that the centrifugal acceleration term $x_1x_4^2$ is entirely neglected in the linearized equations. The observation is also related to results in Hauser et al [4].

3.6 Dynamic Analysis of the Ball On Beam System

The nonlinear design techniques attempt to compensate for the centrifugal acceleration term $x_1x_4^2$ located in the dynamics of $\dot{x}_2$. Hauser *et al* neglect this term in their approximate input-to-output linearization [4]. We evaluate the effects of this term in a simulation study. A simulation was run with simple LQR gains applied to two different systems: the original system (3.1a)-(3.1d) and a modified system with the $x_1x_4^2$ neglected, $\dot{x_2} = \sin(x_3)$. By neglecting the centrifugal acceleration term $x_1x_4^2$ the modified system is able to return much higher initial values $x_1(0)$ to the origin.

The position response x_1 for both systems with the same parameters is seen in Figure 3.2. The centrifugal acceleration term produces an "anti-damping" term in the dynamics $\dot{x}_2$. The states associated with this centrifugal acceleration term are specifically addressed in the following control design. Theory associated with this concept will be discussed in Chapter 4.

The state dynamics of x_3 and x_4 in the equations (3.1a) thru (3.1d) are separated from the state dynamics of x_1 and x_2, as shown in Figure 3.3.

We chose the Lyapunov

$$\dot{V}(x_3, x_4) = x_3\dot{x}_3 + x_4\dot{x}_4 \quad (3.4a)$$

$$\dot{V}(x_3, x_4) = x_3x_4 + x_4u \quad (3.4b)$$

which implies stability of x_3 and x_4 with

$$u = -x_4 - x_4x_3^2 - x_3 \quad (3.5)$$

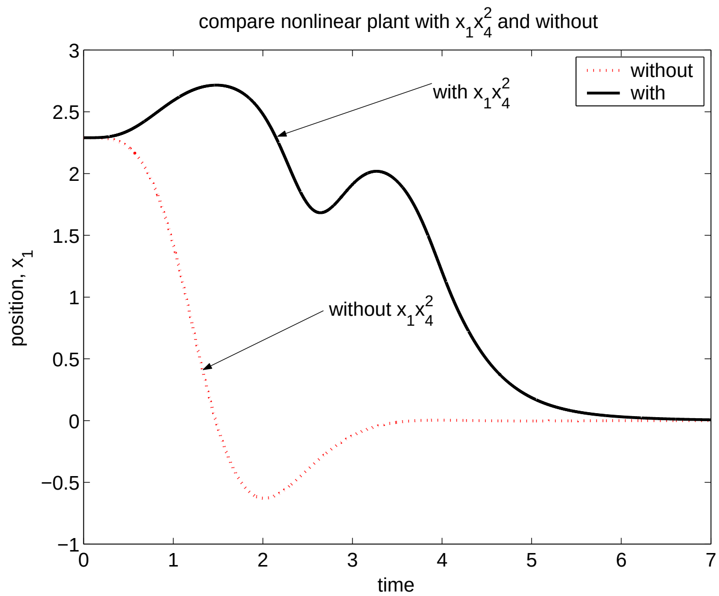


Figure 3.2: Comparison of a Modified System with the Original System

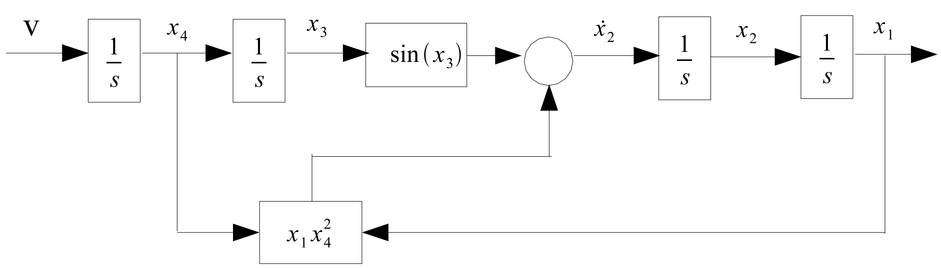


Figure 3.3: Block Diagram of Simplified ball-on-beam System

To verify states $x_3 = \theta$ and $x_4 = \dot{\theta}$ are stabilized, the system (3.1a)–(3.1a) was simulated with an initial condition of $x_3(0) = \pi/10$. State x_3 and x_4 were returned to the origin, as seen in their response shown in Figure 3.4.

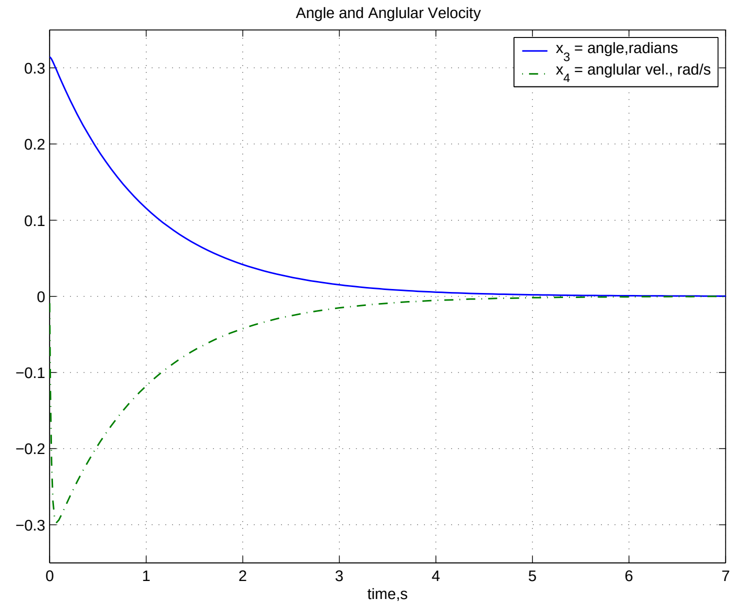


Figure 3.4: Stability of x_3 and x_4 confirmed

The Lyapunov designed control (3.5) and a new user defined control input v were feedback into the original ball-on-beam system.

$$\dot{x_1} = x_2 \tag{3.6a}$$

$$\dot{x_2} = x_1 x_4^2 + \sin(x_3) \tag{3.6b}$$

$$\dot{x_3} = x_4 \tag{3.6c}$$

$$\dot{x_4} = -x_4 - x_4 x_3^2 - x_3 + v \tag{3.6d}$$

The new dynamics (3.6a)–(3.6d) were linearized about the origin, and an LQR optimized set of gains K were chosen to stabilize the origin. The following control was chosen.

$$u_{nl} = -x_4 - x_4 x_3^2 - x_3 - K \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

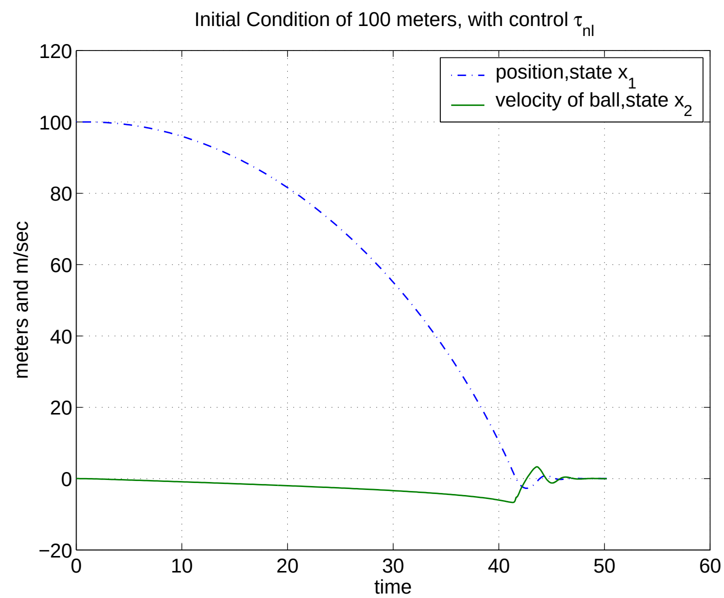


Figure 3.5: x_1 Response with nonlinear control, u_{nl}

A plot of the position of the ball, velocity of the ball, angle of the beam and angular rate of the beam are seen in Figure 3.5 and 3.6. Numerous initial conditions were applied to the system, and the system was stable for initial conditions up to 150 meters (i.e. over time, all states would return to zero $x = 0$).

A comparison of the LQR control input from Section 3.1 and the above control u_{nl}, is seen in Figure 3.7. The control input u_{nl} produces a similar settling time for similar initial conditions, and although damping in the linear control law is higher, u_{nl} maintains stability for initial conditions *far* outside of the linear LQR state-feedback's ($u = -Kx$) stability range.

Hauser et al.'s [4] approximate input-output linearization mentioned in Section 3.3 is compared to the above control u_{nl} in Figure 3.8. Both position responses in Figure 3.8 are simulated with the same mechanical constants, i.e. ball mass and beam mass, and initial position $x_1(0) = 1.92\ m$. The control input u_{nl} produces a slightly better settling time than Hauser's control input and u_{nl} maintains stability for initial conditions *far* outside of Hauser's control input stability range, $x_1(0) > 1.92\ m$.

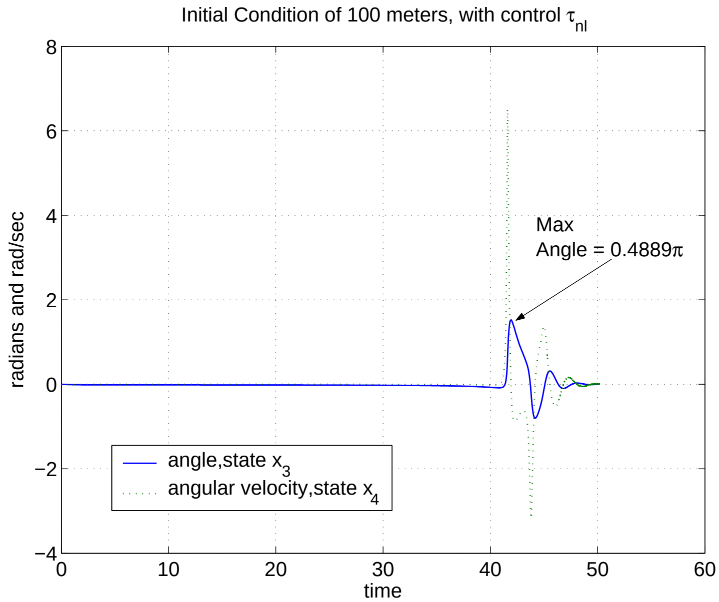


Figure 3.6: x_3 and x_4 Responses with nonlinear control, u_{nl}

3.7 Concluding Remarks

The ball-on-beam example system, discussed in this chapter, has a poorly defined relative degree. Well-known nonlinear methods fail when the relative degree is uncertain or changing. This problem was tackled by designing a nonlinear control component that directly addresses the state variables that affect the system's relative degree and with the relative degree addressed, an outer loop linear controller became significantly more effective. With these observations, a new approach to nonlinear control design has been outlined. In the following chapter, five more examples will be discussed and used as catalyst in outlining this new control design approach.

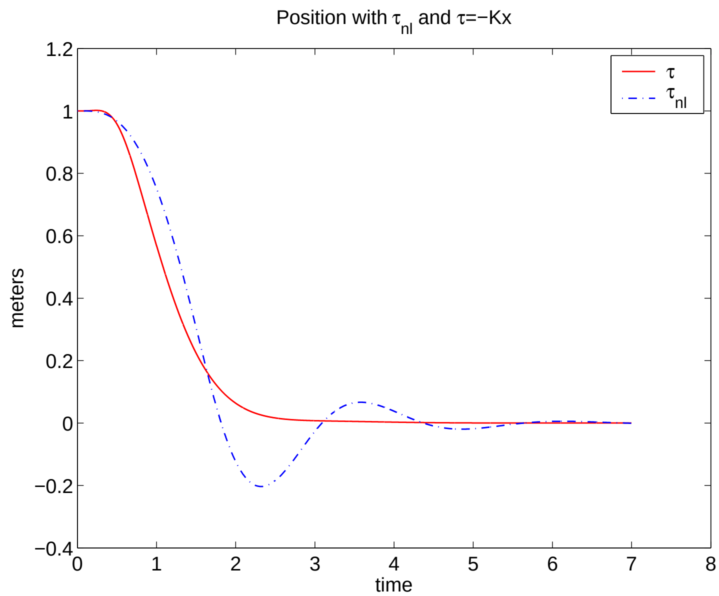


Figure 3.7: x_1 Response with u_{nl} and $u = -Kx$ control inputs

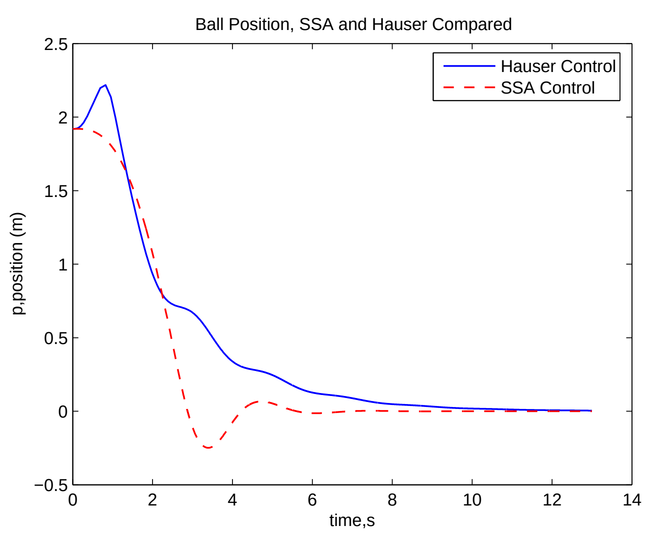


Figure 3.8: x_1 Response with u_{nl} and Hauser's control inputs

Chapter 4

New Subset Stabilization Approach (SSA) Control Design

In this chapter, five example systems will supplement the ball-on-beam example in the previous chapter. For the purposes of this dissertation, the approach used in the ball-on-beam example will be referred to as a Subset Stabilization Approach (SSA). This approach provided a control effort, designed through nonlinear and linear methods, that stabilized the ball-on-beam nonlinear plant well beyond standard linear techniques. Additionally, the new SSA control was able to provide a feasible control effort where traditional nonlinear control techniques could not. To take a more in depth look at SSA control within the aspects of nonlinear systems, we will examine five different nonlinear system examples. In each example, the traditional nonlinear approaches discussed in Chapter 2 fail to provide a feasible solution, or do not provide a stabilizing control effort. In each example system, a standard linear control technique will be applied and compared with the new SSA.

4.1 Subset Stabilization Approach

The proposed SSA control combines the benefits of larger stability region offered by nonlinear control design with the simplicity and design ease of the linear systems approach. The SSA was born out of two observations of many practical nonlinear problems:

- System nonlinearities are often a function of a subset of the full state vector. (Robotic systems are a notable exception to this observation, but rigid robot dynamics are well structured and can be controlled by all of the existing nonlinear methods.)
- In situations where existing nonlinear methods fail, a subset of the state variables contribute to structural problems that result in control singularity or loss of controllability.

The SSA nonlinear control approach reduces the effects of variables that contribute to either nonlinearities or structural problems. Steps in the SSA nonlinear design approach are:

1. Partition the state vector into two sets of variables

$$x = \begin{bmatrix} z \\ w \end{bmatrix}$$

   where the variables w are those contributing to the dominant nonlinearities of the system. In some systems certain variables trigger control singularity, or impact system structure; these variables would be included in the set of variables w. The system model can be expressed in the form

$$\dot{z} = A_1 z + A_2 w + B_1 u + f_1(z, w, u) \tag{4.1}$$

$$\dot{w} = A_3 z + A_4 w + B_2 u + f_2(z, w, u) \tag{4.2}$$

   Here $A_i,\ i = 1, ..., 4$ and $B_j,\ j = 1, 2$ are linear matrices of appropriate dimension, and functions $f_k(z, w, u),\ k = 1, 2|$ represent the nonlinearities of the system.

2. Choose a candidate Lyapunov function $V(w)$, a positive definite function of the subset of variables w.

3. Design an inner loop control u_{lyp} so that the time derivative $\dot{V}(w)$ is negative definite in w.

4. Apply the control $u = u_{lyp} + v$ to the system 4.1 to stabilize the w-dynamics, and to introduce a new input variable v for outer loop control. The partially stabilized system is less sensitive to the nonlinear terms $f_k(z, w, u)$ associated with w.

5. Design an outer loop control v using linear system methods to stabilizes either (1) the linearization of the uncompensated plant 4.1, or (2) a linearization of the plant with the

inner loop control u_{lyp}. Past experience has been successful by applying linear quadratic regulator (LQR) methods with either technique.

6. The composite SSA control is given by

$$\begin{aligned} u &= u_{lyp} + v \\ &= u_{lyp} + u_{LQR} \end{aligned} \tag{4.3}$$

Local asymptotic stability is guaranteed by standard Lyapunov stability techniques and Linear techniques around an operating point.

4.2 Five Case Studies

In this section, five example plants are studied to illustrate the challenges of standard nonlinear control methods and the utility of the SSA control design method. A comparison of the resulting state responses is presented for each example system.

4.2.1 System 1

Consider the system

$$\begin{aligned} \dot{x}_1 &= x_2 + \tanh(u) \\ \dot{x}_2 &= x_1 + x_2^2 + u \\ y &= x_1 \end{aligned} \tag{4.4}$$

A Lyapunov design based on a Lyapunov function

$$V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) \tag{4.5}$$

is examined first. The derivative of V is given by

$$\dot{V}(x_1, x_2, u) = 2x_1x_2 + x_2^3 + x_1 \tanh(u) + x_2u.$$

Notice that the expression is not an affine function in u. Therefore, an input that yields a negative definite $\dot{V}$ is not determined easily and a stabilizing feedback control cannot be found in a straightforward manner by Lyapunov's design method and using the Lyapunov function (4.5). Though tuning with trail and error using different Lyapunov functions could possibly improve these results. But, for this Chapter, Lyapunov's design method will be standardized to include the energy like function (4.5).

Next, consider the input-output linearization approach. The system 4.4 has relative degree $r = 1$, so the input u appears explicitly in the first derivative of the output y:

$$\dot{y} = \dot{x}_1 = x_2 + \tanh(u)$$

Applying the nonlinear input

$$u = \operatorname{arctanh}(v - x_2)$$

yields a first-order linear input-output dynamic model

$$\dot{y} = v$$

to which a linear design is easily determined and stabilizes the result. The nonlinear control u approaches infinity, however, near the condition $|v - x_2| = 1$, due to the domain of the hyperbolic arctangent, $|v - x_2| < 1$. Therefore, the nonlinear solution is localized around the $v - x_2 = 0$.

For input-state linearization, a transformation must exist $T(x)$ that will allow a linear relationship between the input and the states. For $T(x)$ to exist, the ad field for (4.4) must be full rank. Do to structure of the system,non-affine structure, input-state linearization is not a feasible solution for control. The system's structure also prohibits the use of integrator backstepping due because it is not in strict feedback form.

To design a SSA controller, the model (4.4) is first examined for nonlinear terms. The dominant state nonlinearity is the quadratic term x_2^2 in second state equation. The presence of this term overwhelms any linear control law as x_2 deviates from zero. Therefore, a nonlinear control can be designed to stabilize x_2, and consequently reduce the effect of the nonlinear term x_2^2. Heuristically, the plant with nonlinear inner loop control will behave more like a linearized model that can be used to design a stabilizing linear control law.

Choose the Lyapunov function $V(x_2) = \frac{1}{2}x_2^2$. Then

$$\begin{aligned} \dot{V}(x_2) &= x_2\dot{x}_2 \\ &= x_1x_2 + x_2^3 + x_2u \end{aligned}$$

Choose the inner loop control

$$u_{lyp} = -x_1 - x_2^2 - x_2$$

to guarantee stability of the variable x_2. Apply the control $u_{lyp} + v$ to the original plant (4.4). The first term is the inner loop control, and the second term v is a new input for outer loop control. The resulting model is

$$\begin{aligned} \dot{x}_1 &= x_2 + \tanh(u) \\ \dot{x}_2 &= -x_2 + v \end{aligned}$$

Where the outer loop control v is designed using LQR methods; the gains used are $K = \begin{bmatrix} 0.3162 & 0.3162 \end{bmatrix}$[1]. The SSA control is given by the following equations.

$$u = u_{lyp} - Kx.$$

A comparison of the state responses for LQR control and the SSA control is seen in Fig. 4.1. This response is seen on the edge of stability for the LQR gains. If the initial condition of

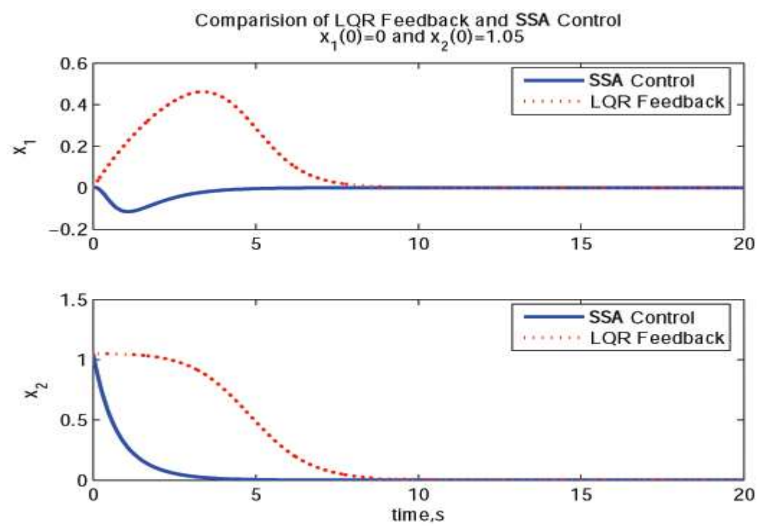


Figure 4.1: Example System 1: State responses to LQR feedback and SSA control

$x_2(0)$ is increased slightly, then the LQR gains do not provide a stable feedback control law. However, SSA control provides stability well beyond the initial condition, $x_2(0) \geq 1.05$.

[1] Operating point $\mathbf{x} = \mathbf{0}$ and weights $Q = I_{2\times 2}, R = 10$

4.2.2 System 2

The second case study examines the second-order plant modeled by

$$\begin{aligned}\dot{x}_1 &= x_1 + x_2 + x_2^2 \\ \dot{x}_2 &= x_1 + x_2^2 + u \\ y &= x_1\end{aligned} \tag{4.6}$$

Again, Lyapunov control design based on the Lyapunov function $V(x) = \frac{1}{2}(x_1^2 + x_2^2)$ yields a undesirable control law.

$$u = -x_1^2 x_2^{-1} - 2x_1 - x_1 x_2 - x_2^2$$

The control law has a singularity at $x_2 = 0$, so the plant input u goes to infinity as the state approaches the equilibrium point $\mathbf{x} = \mathbf{0}$.

This case study has a relative degree of $n = 2$, implying the input will explicitly appear in the second derivative of the output equation. The input-output linearizing control designed around System 2 yields the control law

$$u = -\frac{2x_1 + x_2 + 2x_2^2 + 2x_1 x_2 + 2x_2^3}{2x_2 + 1}$$

which has a singularity at $x_2 = -0.5$, and is only locally valid around the operating point $\mathbf{x} = \mathbf{0}$.

For input-state linearization, the system is written as

$$\dot{\mathbf{x}} = f(x) + g(x)u, \text{ where } g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ and } f(x) = \begin{bmatrix} x_1 + x_2 + x_2^2 \\ x_1 + x_2^2 \end{bmatrix}$$

and the ad field

$$[g(x) \;\; ad_f g] = \begin{bmatrix} 0 & -1 - 2x_2 \\ 1 & -2\,x_2 \end{bmatrix} \tag{4.7}$$

must have full rank for all values of $x_{1,2}$, or $rank = 2 \colon \forall \mathbf{x} \in \mathrm{I\!R}^2$. Clearly, from (4.7), a state pair along the line $(x_1, x_2) = (\mathrm{I\!R}, -0.5)$ produces a low rank ad field ($rank < 2$). Therefore, system 2 does not satisfy a necessary and sufficient condition for the existence of a transformation matrix $T(x)$ to exist. Finally, similar to System 1, System 2 does not have the correct structure to perform integrator backstepping control design. This leads to the use of a SSA control design.

The SSA control design method will focus on the nonlinear function, x_2^2. Following the example design process for System 1 yields the inner loop control

$$u_{lyp} = -x_1 - x_2^2 - x_2.$$

The LQR gains are $K = \begin{bmatrix} 4.1669 & 2.0714 \end{bmatrix}$, and the SSA control is given by

$$\begin{aligned} u &= u_{lyp} + u_{LQR} \\ &= -x_1 - x_2^2 - x_2 - Kx \end{aligned} \tag{4.8}$$

A comparison of the state responses for simple LQR control and the SSA control is seen in Fig. 4.2. As in System 1, Figure 4.1, the system's state response is plotted on the edge of a

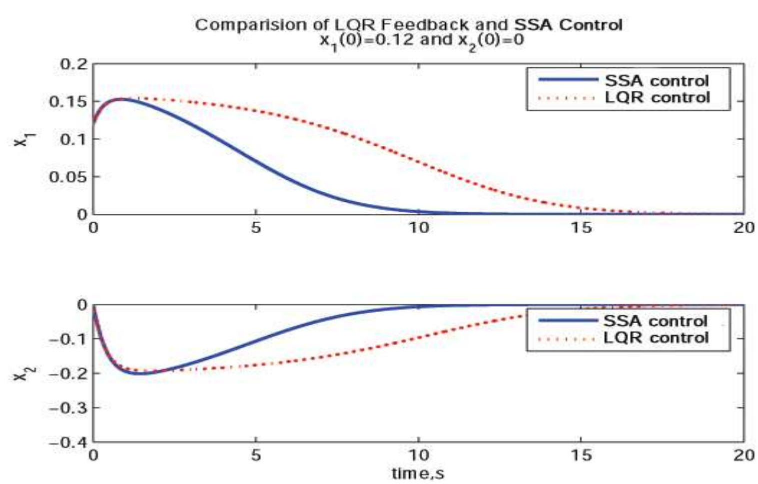


Figure 4.2: Example System 2: State responses to LQR feedback and SSA control

stable state region $(x_1(0), x_2(0)) = (0.125, 0)$. If the LQR designed control were to be simulated past the shown initial condition, then it would become unstable, and the states would go to ∞. The SSA control stabilizes System 2 beyond this initial condition and provides a faster settling time at the current initial condition $(x_1(0), x_2(0))$.

4.2.3 System 3

Consider the second-order plant model

$$
\begin{aligned}
\dot{x}_1 &= \sin(x_2) \\
\dot{x}_2 &= u \\
y &= x_1
\end{aligned}
\tag{4.9}
$$

which has an equilibrium point at the origin $x_1(0), x_2(0) = (0, 0)$. Lyapunov-based design based on the Lyapunov function $V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$ yields the control

$$
u = -\frac{x_1 \sin(x_2)}{x_2}
$$

with a singularity along the state space line $(x_1, x_2) = (\mathbb{R}, 0)$ which contains the origin. This control is not a valid solution for control around the origin.

Input-output linearization control design produces the control

$$
u = \frac{v}{\cos(x_2)}
$$

As with the Lyapunov control solution, this solution u has a singularity at the origin.

For input-state linearization, the ad field for the plant (4.9) is given as

$$
\begin{bmatrix} g(x) & ad_f g \end{bmatrix} = \begin{bmatrix} 0 & -\cos(x2) \\ 1 & 0 \end{bmatrix}
\tag{4.10}
$$

The ad field (4.10) loses full rank when x_2 satisfies $\cos(x_2) = 0$. As for backstepping, the plant model is in strict feedback form, so backstepping is a feasible approach, but produces a control with singularity under the condition $1 - x_1^2 = 0$, or $(x_1, x_2) = (1, \mathbb{R})$.

The SSA approach is now considered as a control solution design method. The plant nonlinearity is only a function of the variable x_2, so the nonlinear aspect of the SSA control is designed to stabilize the state x_2. The Lyapunov function $V(x_2) = \frac{1}{2}x_2^2$ is chosen and yields a control

$$u_{lyp} = -x_2$$

The linear portion of SSA control is designed by LQR methods to yield state feedback gain $K = \begin{bmatrix} 0.3162 & 0.3162 \end{bmatrix}$. The complete SSA control is totally linear in this case example:

$$\begin{aligned} u &= u_{lyp} + u_{LQR} \\ &= -x_2 - Kx \end{aligned}$$

A comparison of state responses is in Fig. 4.3. The linear control fails to compensate for nonlinear effects of x_2, as seen in a comparison of the SSA and linear responses in Figure 4.3.

4.2.4 System 4

Consider a third-order plant modeled by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_2 + \sin(x_3) \\ \dot{x}_3 &= u \\ y &= x_1 \end{aligned} \tag{4.11}$$

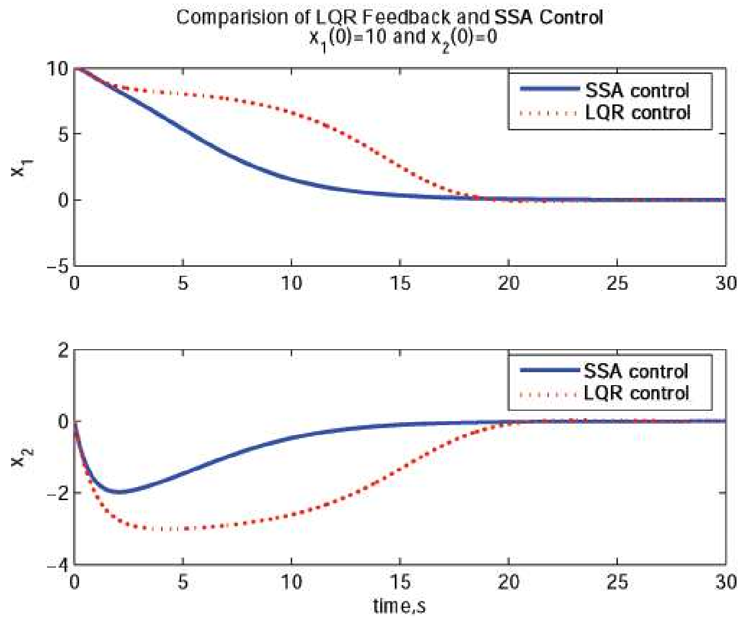


Figure 4.3: Example System 3: State responses to LQR feedback and SSA control

Using the same Lyapunov energy function $V(\mathbf{x}) = \frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right)$, the Lyapunov-based design produces the control

$$u = -\frac{x_1 x_2 + x_2^2 + x_2 \sin(x_3)}{x_3}$$

The Lyapunov control is similar to the Lyapunov control in the previous three systems; the control produces a singularity along the plane $(x_1, x_2, x_3) = (\mathbb{R}, \mathbb{R}, 0)$.

System 3 has a relative degree of $r = 3$ and the input-output linearizing control design yields

$$u = -\frac{x_2 + \sin(x_3)}{\cos(x_3)}$$

Again, the input-output linearization standard design technique produce a control law with singularities along the plane $(x_1, x_2, x_3) = (\mathbb{R}, \mathbb{R}, 0)$. Therefore, the equilibrium point is not stabilized with either Lyapunov or input-output linearization design. Similar to the earlier example systems, the ad field of the System 4 (4.11) is not full rank for specific conditions ($\sin(x_3) = 0$ which includes the origin).

For this example, the SSA control design method stabilizes the variable x_3, included in most of the nonlinear aspects of System 4. An inner loop control that stabilizes x_3 by Lyapunov's control design method is given as

$$u_{lyp} = -x_3$$

and applying u_{lyp} to the plant yields the new model.

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_2 + \sin(x_3) \\ \dot{x}_3 &= -x_3 + v\end{aligned}$$

The optimized LQR gains that stabilize the response about the origin and minimize the LQR cost function with Q, R weights are given by $K = \begin{bmatrix} 0.3162 & 4.9487 & 2.3162 \end{bmatrix}$. The SSA control's final form is given by

$$u = -x_3 - Kx$$

The responses of the three state variables are shown in Fig. 4.4 with the initial conditions $(x_1, x_2, x_3)_o = (3.75, 0, 0)$. The SSA control approach is able to compensate for nonlinearities related to x_3, as seen in Figure 4.4. The linear control K_{LQR} allows for must higher oscillations, and will eventually go unstable at higher initial conditions $x_1(0) > 3.75$.

4.2.5 System 5

The ball-on-beam example was given in the previous chapter, Chapter 3. To better understand the improvements of the SSA control, a simplified version of the ball-on-beam dynamics

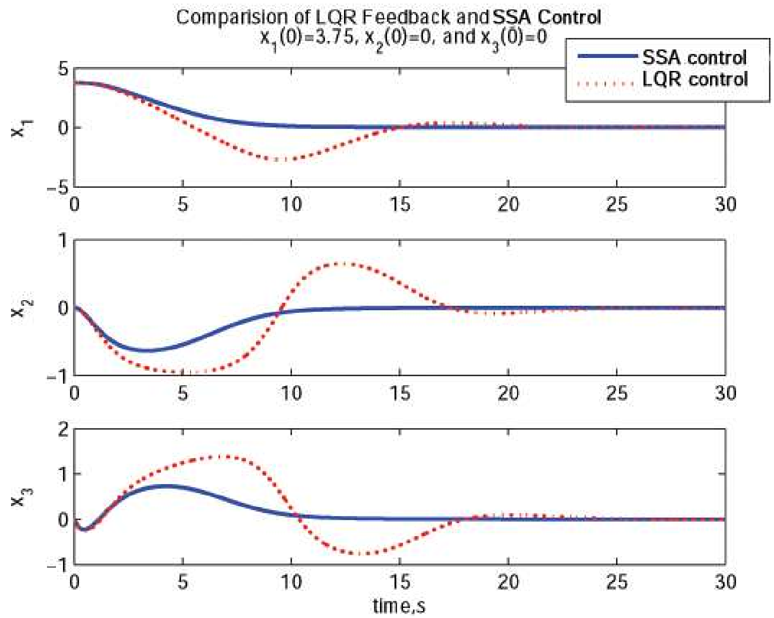


Figure 4.4: Example System 4: State responses to LQR feedback and SSA control

are modeled by the following fourth order system

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_1 x_4^2 + \sin(x_3) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= u \\
y &= x_1
\end{aligned}
\tag{4.12}
$$

As in the previous chapter, the plant output y is the ball position along the beam and the beam angle is the state variable x_3. Standard Lyapunov design based on $V(x) = \frac{1}{2}x^T Px$, with $P = I_{4\times 4}$, yields the control law

$$
u = -\frac{x_1 x_2 + x_1 x_2 x_4^2 + x_2 \sin(x_3) + x_3 x_4}{x_4}
$$

with a singularity along the state space region $(x_1, x_2, x_3, x_4) = (\mathbb{R}, \mathbb{R}, \mathbb{R}, 0)$, including the origin.

Input-output linearization yields the control law

$$u = -\frac{x_4 x_2 + \cos(x_3)}{x_1}$$

and, as in the Lyapunov designed control, the control u blow up at the origin, $x_4 = 0$. Hauser et al. show the ball-on-beam problem cannot be controlled by input-state linearization because the necessary conditions for state transformation are not satisfied [4], and is verified in Section 3.4. Integrator backstepping is not a feasible control design approach because the model (4.12) is not in strict feedback form.

It is obvious, $\sin(x_3)$ and the x_4^2 are the main nonlinear terms. Therefore, in SSA control, a nonlinear inner loop is designed to stabilize the variables x_3 and x_4. The Lyapunov function $V(x_3, x_4) = \frac{1}{2}(x_3^2 + x_4^2)$ is used to design the SSA's inner loop control of x_3 and x_4. The resulting nonlinear inner loop control is given by

$$u_{lyp} = -x_4 - x_4 x_3^2 - x_3$$

Applying the inner loop control law $u = u_{lyp} + v$ to the plant yields

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_1 x_4^2 + \sin(x_3) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -x_4 - x_4 x_3^2 - x_3 + v
\end{aligned}$$

and LQR gains are designed for the outer loop control variable v, with the equilibrium point as the origin. The final form of the SSA control is given by

$$\begin{aligned} u &= u_{lyp} + v \\ &= -x_4 - x_4 x_3^2 - x_3 - Kx \end{aligned}$$

A plot of the state variable responses is shown in Fig. 4.5. and the nonlinearities compensated

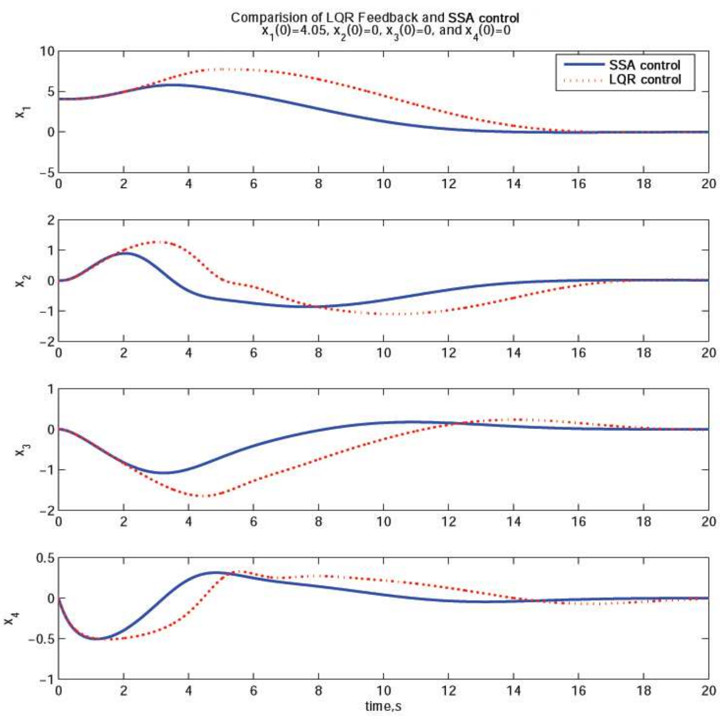


Figure 4.5: Example 5 (ball-on-beam control): State responses to LQR feedback and SSA control

for with SSA control are easily identified in the LQR state variable responses. As in Chapter 3, the SSA control approach allows for a larger stability region for LQR control designed, after an inner loop stabilization routine is applied to x_3 and x_4.

4.3 Theoretical analysis

In this section we present an analysis of the subset stabilization approach for the control of a nonlinear system. This analysis builds upon observations made in Chapter 3 and Sections 4.2.1-4.2.5 in which the SSA compares very favorably to other non hierarchical control approaches in a number of examples. We here bolster these anecdotal examples with theoretical observations that justify the use of the SSA under specific conditions. In particular, we establish that the SSA, in a simplistic sense, acts to decrease the impact of the system nonlinearities so that a robust control law designed about a given operating point may in fact extend its region of convergence in closed loop operation. While any robust control law design approach could be used, we here present our analysis in terms of the SSA as a companion to LQR control design.

While the examples presented in the previous sections establish evidence of the utility of our SSA control design method, they do not address the question of *why* the SSA method occurs at all. We now present a theoretical analysis of the SSA method and provide another example to illustrate the method. Our analysis will use the following definitions.

Definition 10 Suppose that there exists a control input $u(z, w)$ such that, for the corresponding closed loop system

$$\begin{aligned} \dot{z} &= A_{11}z + A_{11}w + f_{11}(w) + f_{12}(z, w) + f_{13}(u(z, w)) \\ \dot{w} &= A_{11}z + A_{11}w + f_{21}(z) + f_{22}(z, w) + f_{23}(u(z, w)) \end{aligned}$$

there is a Lyapunov function on the state subset w; that is, there exists a function $V(w) > 0$ such that $\frac{d}{dt}V(w(t)) < 0$ for the system in closed loop with $u(z, w)$. Then we say that the open-loop system $(4.13a)$-$(4.13b)$ satisfies the *the subspace stabilizability condition.*

Definition 11 Consider a system S that satisfies the subset stabilizability condition. The *null subset manifold of $\mathcal{S}$* is the set of points $\begin{bmatrix} z \\ w \end{bmatrix} \in \mathrm{I\!R}^{n+m}$ where $w = 0$.

Now, consider a system in the general form

$$\dot{z} = A_{11}z + A_{12}w + f_{11}(w) + f_{12}(z,w) + f_{13}(u) \quad (4.13a)$$

$$\dot{w} = A_{21}z + A_{22}w + f_{21}(z) + f_{22}(z,w) + f_{23}(u) \quad (4.13b)$$

the state vectors are $z \in \mathbb{R}^n$ and $w \in \mathbb{R}^m$. The system is controlled by a SSA-based control design, as shown in Figure 4.6. Let

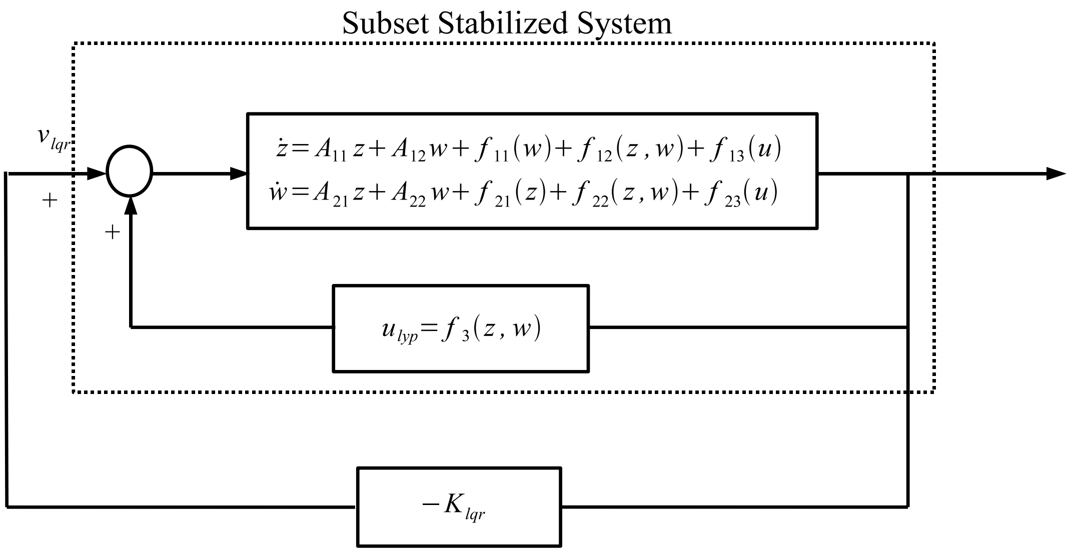


Figure 4.6: Subset Stabilization Approach Block Diagram

$$u_{lyp}(z,w) = \bar{u}_{lyp}(z,w) + u_2(w_1^n, ..., w_m^n)$$

be a Lyapunov designed control effort where $\bar{u}_{lyp}(z,w)$ drives the system to the null subset manifold $w = 0$, n is an integer, and u_2 is selected so that (1) an increase in n increases the rate of convergence of w to the null subset manifold, i.e., $\dot{V}(w, n+1) \leq \dot{V}(w,n)$ for all n, and (2) $\nabla_w u_2|_{w=0} = 0$. We interpret the action of the SSA inner loop control law u_{lyp} in a fashion similar to variable structure control, where state trajectories are often divided into a reaching phase and a manifold tracking phase. In the case of SSA control, the closed loop system may be considered to be in a reaching state when the system state is distant from the null space manifold, which can be characterized either in terms of $\|w\|$ or in terms of the relative magnitudes of the inner loop control command u_{lyp} and the outer loop control command u_{lqr}. Given an appropriate choice of system states in w, once the state trajectory is "near" to the null subset manifold, the effects of the system nonlinearities may be greatly reduced. This leads to two key observations regarding the success of SSA control in contrast to other control methods:

1. While effects of the nonlinearities in the underlying system may be reduced, they are not entirely canceled. This stands in contrast to both linear robust control, which describes linearities primarily in terms of norm bounds and the small gain theorem, and to nonlinear control, which uses a precise model of system nonlinearities in the selection of its control commands.

2. The use of an outer-loop robust control law is strengthened because a properly selected null subset manifold and SSA inner loop control law effectively limits the gain of undesirable nonlinearities that exist in the open-loop system.

As a result, we may consider the SSA closed loop system to be in a "reaching" phase while its states are outside of the region of convergence of the outer loop law, and consider the system to be in a "surface tracking phase" when its states are in the region of convergence of the outer loop law.

We illustrate these principles and the tuning of the integer parameter n in the auxiliary input u_2 in the following example

Example 6 *SSA Bounded Subset Example Consider System 2 from Section 4.2.2*

$$\dot{x}_1 = x_1 + x_2 + x_2^2 \tag{4.14a}$$

$$\dot{x}_2 = x_1 + x_2^2 + u \tag{4.14b}$$

A subset of the state space $x \in I\!R^2$ is chosen $x_2 \in I\!R^1$ and is bounded with a Lyapunov control designed input u_{lyp}. As in section 4.2.2, we use a Lyapunov function $V(x_2) = \frac{1}{2}x_2^2$ to design a *nonlinear feedback control for the dynamics of the state* x_2. *We define a set of parameterized subset stabilizing control inputs as*

$$u \;=\; -x_1 - x_2^2 - x_2 - x_2^{2n+1} \tag{4.15}$$

where n *is an integer that is used to adjust the "gain" of the subspace stabilization control law* $u(z, w)$ *for large values of* x_2.

Phase portraits based on simulation results of the subset stabilized system are shown in Figure 4.7(a). [2] *The null subset manifold* $x_2 = 0$ *is easily seen as* $x_2(t) \to 0$ *as* $t \to \infty$. *An increase of the exponent integer parameter* n *in* (4.15) *will increase the control effort* u *for large deviations* x_2 *from the desired null space manifold. By consequence, the impact of the nonlinearities involving* x_2 *are decreased once the state trajectories come near to the null subset manifold. As seen in Figure 4.7(b) and Figure 4.7(c) the corresponding state trajectories lead the subset stabilized phase portraits to a smaller deviation from the null subspace manifold.*

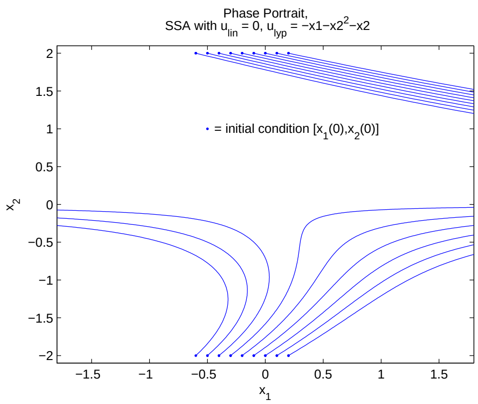
The phase portraits clearly illustrate that only the state x_2 *is regulated by the SSA controller* (4.15). *We now examine the effects of the SSA control law in closed loop with the outer loop LQR controller (see figure 4.6. The* region of convergence *(ROC) refers to the a region of state space, ROC* $\in I\!R^2$, *where the system* $\dot{x} = f(x)$ *converges to the origin as* $t \to \infty$. *An increase in the ROC area is a valid quantitative method to compare the SSA controller in figure 4.6 and a traditional linear LQR control solution.*

The traditional linear control solution uses an LQR optimization and the weights Q *and* R (4.16).

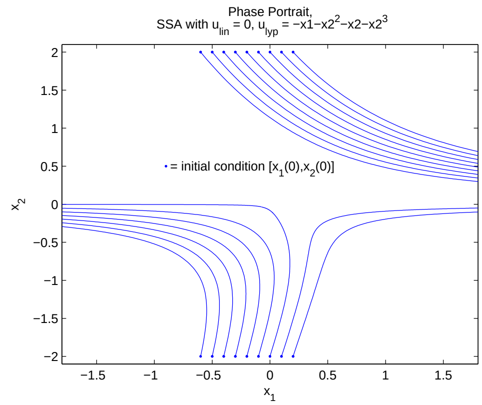
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad and \quad R = 10 \tag{4.16}$$

The required linear model for the LQR optimization is based on a linearization of the nonlinear model $(4.14a)$-$(4.14a)$ *evaluated at the origin. The resulting control effort for LQR linear control*

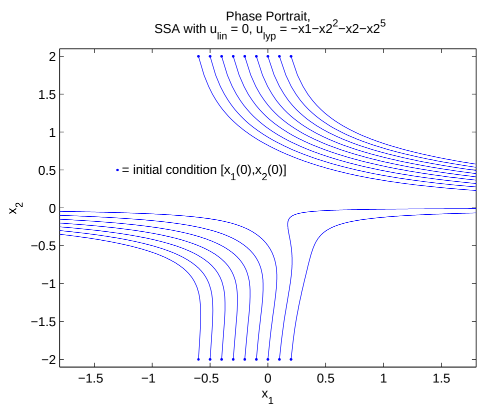
[2] *a plot of the trajectory of* $x_1(t)$ *with respect to the trajectory of* $x_2(t)$.



(a) $u = -x_1 - x_2^2 - x_2$



(b) $u = -x_1 - x_2^2 - x_2 - x_2^3$



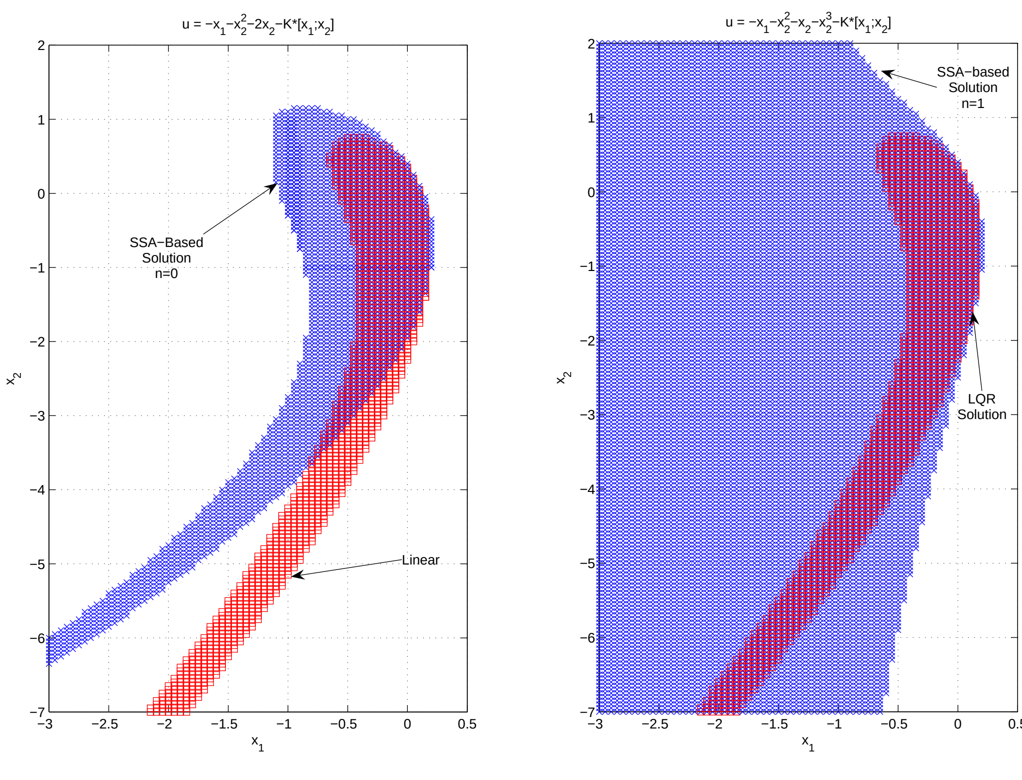
(c) $u = -x_1 - x_2^2 - x_2 - x_2^5$

Figure 4.7: Subspace stabilized system phase portrait. The integer parameter n in equation (4.15) is varied from 0 (a) to 1 (b) to 2 (c). Observe that with increased values of n the phase portrait tends to more closely approach the null subset manifold $x_2 = 0$.

is a negative linear feedback with gain K.

$$K = \begin{bmatrix} 5.3956 & 3.3002 \end{bmatrix}$$

The SSA controller was designed with the SSA control law 4.15 in closed loop with an outer loop LQR controller designed with the same matrices (4.16) *and a different linearization. Two versions of SSA control design are compared, $n = 0$ and $n = 2$. Observe in Figure 4.8(a) that the area of the region of convergence (blue) of the SSA-based solution is comparable to slightly greater than the area of the region of convergence corresponding to the simple linear LQR approach (red). Notice further in Figure 4.8(b) that the region of convergence significantly increases with the SSA control law parameter n.*



(a) $u_{SSA} = -x_1 - x_2^2 - 2x_2 + v$ (b) $u_{SSA} = -x_1 - x_2^2 - x_2 - x_2^3 + v$

Figure 4.8: Region of Convergence, Comparison of SSA-based control and LQR control efforts for System 2

4.4 Conclusion

Five case studies are presented in the previous chapter in which existing well-known nonlinear control design methods produced either control solutions with singularities or were unable to produce any feasible control solution. In engineering practice, standard nonlinear control design methods may not be mathematically tractable, or may result in control laws with singularities, or the design processes cannot be completed due to system structural deficiencies. A SSA nonlinear design proves effective in all of these specific system examples. An inner loop nonlinear control is first designed to stabilize a *subset* of the plant's state variables. Only those variables contributing to significant plant nonlinearities are stabilized by the inner loop control. The inner loop control law also introduces a new control variable for outer loop control. An outer loop control designed by usual linear system methods completes the SSA control law. The SSA control design approach combines the simplicity of linear design methods with the improved region of stability of nonlinear design methods, yet avoids the pitfalls of some standard nonlinear control design methods.

Chapter 5

Conclusions

In this chapter a description of conclusions drawn for the Subset Stabilization Approach (SSA) are discussed. The major advantages of SSA control design are re-iterated and major design considerations are discussed. A final argument is given for using SSA control design and in which situations SSA should be used.

5.1 Design Advantages

The Subset Stabilization Approach design approach has a direct application to a large group of systems and has proven itself usefulness for several specific nonlinear system examples, i.e. the ball-on-beam example in Chapter 3 and the five other example systems in Section 4.2. In each example, the traditional nonlinear design approaches failed to provide a valid control effort or were structurally inapplicable. The linear control method provided control efforts that were stable within small regions around the given operating point. The SSA-based control solutions were able to increase the region of convergence volume for the linear control effort and ,in the ball-on-beam example, provide a seemingly globally stable control.

SSA-based control solutions were not unique and provided tuning possibility, as discussed in Section 4.3. The system in Example 6

$$\begin{aligned}\dot{x}_1 &= x_1 + x_2 + x_2^2 \\ \dot{x}_2 &= x_1 + x_2^2 + u\end{aligned}$$

was simulated with multiple control solutions

$$u = -x_1 - x_2^2 - x_2 - x_2^{2n+1} - K \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where $n = 0$ and $n = 1$ were chosen. As n increased, the control input magnitude increased for large values of x_2 and the system responded faster. As with input-output linearization, input-state linearization, Integrator backstepping, and Lyapunov control design the response of the system may be changed by the choice of the Lyapunov function used for the design procedure,i.e. the energy function in example 6.

5.2 SSA for Nonlinear Control

The SSA successfully provided a control solution where traditional nonlinear control methods failed and improved the region of convergence for traditional linear control method solutions. In Section 4.3 theory for SSA was given in which a bounded subset of states for the entire state space provided advantages for traditional linear control solutions. The SSA control design technique gives a clear method for solving nonlinear control problems where traditional nonlinear methods fail, and improves linear design solutions for the same systems.

Chapter 6

Future Work

In this chapter a clear description of subsequent future work will be discussed. In previous chapters, a subset stabilization approach (SSA) control design was introduced. Questions related to the advantages of this approach and theoretical explanations where answered for specific examples. Although SSA clearly provides advantages, i.e. the ball-on-beam problem, these advantages are not fully explored through stability and realworld implementation. The SSA designed control effort found in Chapter 3 should be validated with a real world experiment and the stability requirements for a generalized SSA designed control should be analytically examined.

6.1 Experimental Validation of the Ball-on-beam Control Effort

In chapter 3 a SSA-based control design solution was tested on Matlab Simulink model/s-function. Dr. Uran and Dr. Jezernik in Slovenia have built a “ball”-on-beam system and implemented their approach for “ball”-on-beam control. Uran and Jezernik’s Ball and Beam like mechanism replaces the ball by a cart which slides ’frictionless’ on the beam, as in Figure 6.1 [21].

The Ball-on-beam mechanism above is under actuated, like the Ball-on-beam mechanism presented in Chapter 3. The first degree of freedom is present in the angle of a beam which is actuated by an electric direct drive motor. The second degree of freedom is represented by the cart sliding on the beam which is actuated by the movement of the beam. With states defined as $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \varphi$, and $x_4 = \dot{\varphi}$, the dynamics for the ball-on-beam system, in figure 6.1

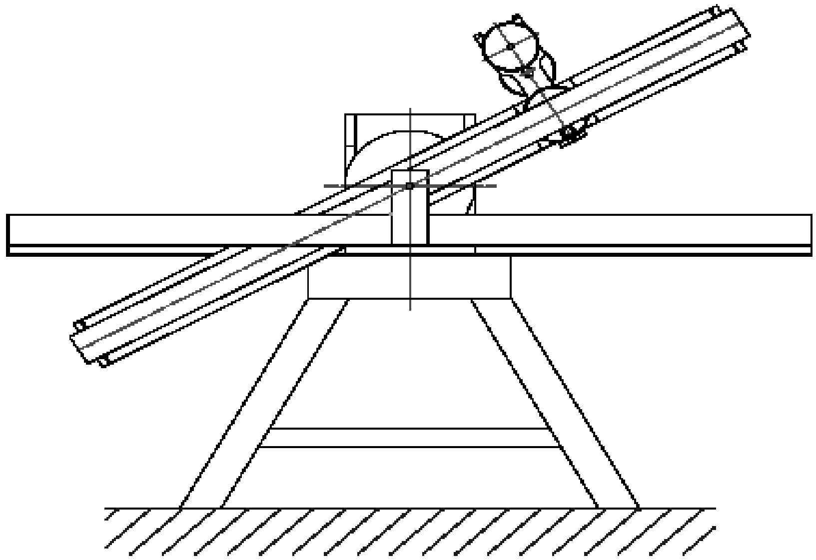


Figure 6.1: Uran and Jezernik Ball on Beam Apparatus

are given.

$$\begin{aligned}\left(m_v r^2 + I_{YYP}\right)\ddot{\varphi} + 2m_v r\dot{r}\dot{\varphi} + m_v gr\cos(\varphi) &= M_\varphi \\ m_v\ddot{r} - m_v r\dot{\varphi}^2 + m_v g\sin(\varphi) &= 0\end{aligned}$$

The variable r is the ball position, φ is the angle of the beam, m_v is a mass of the ball, I_{YYP} is inertia of the beam, g is the gravity constant and M_φ is an input torque. In Uran and Jezernik's ball-on-beam like mechanism the ball(cart) position r , the angle of the beam φ, as well as the ball(cart) position speed and the beam angle speed are measured through various sensors. Using the predefined states $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \varphi$, and $x_4 = \dot{\varphi}$. Uran and Jezernik's ball-on-beam dynamics equations become

$$\begin{aligned}\left(m_v x_1^2 + I_{YYP}\right)\dot{x_4} + 2m_v x_1 x_2 x_4 + m_v g x_1\cos(x_3) &= M_\varphi \\ m_v\dot{x_2} - m_v x_1 x_4^2 + m_v g\sin(x_3) &= 0\end{aligned}$$

and are written as a set of simple state equations using the nonlinear invertible transformation $M_\varphi = m_v g x_1 \cos(x_3) + 2m_v x_1 x_2 x_4 + v I_{YYP} + v m_v x_1^2$.

$$\begin{aligned}\dot{x_1} &= x_2 \\ \dot{x_2} &= x_1 x_4^2 - g\sin(x_3) \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= v\end{aligned}$$

We are communicating with Dr. Uran to implement and compare the subset stabilization approach with Uran and Jezernik's control solution presented in [21]. In the following sections, the SSA control solution is formulate and structurally compared with Uran and Jezernik's control solution.

6.1.1 Subset Stabilization Approach to [21]

The equations presented in [21] are used for the following design. The Lyapunov design is chosen around the angle of the beam x_3 and the angular rate of the beam x_4, $V(x_3, x_4) = \frac{1}{2}\left(x_3^2 + x_4^2\right)$. The derivative of $V(x_3, x_4)$ with respect to time is given below.

$$\dot{V} = x_3 x_4 + x_4 v$$

We chose the following control effort

$$u_{lyp} = -x_3 - x_4 - x_3^2 x_4$$

and supplemented it with a user defined input w.

$$\begin{aligned}\dot{x_1} &= x_2 \\ \dot{x_2} &= x_1 x_4^2 - g\sin(x_3) \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= -x_3 - x_4 - x_3^2 x_4 + w\end{aligned} \tag{6.1}$$

The system is linearized using the state matrices

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and the LQR command in Matlab. The following gains were used (`Q=eye(4)` and `R=10`).

$$K = \begin{bmatrix} 0.3162 & 0.7335 & 5.7938 & 2.5620 \end{bmatrix}$$

The invertible transformation used to simplify the dynamics of $\dot{x}_4$ were applied to the control effort, and the control input is shown below.

$$\begin{aligned} M_\varphi &= m_v g x_1 \cos(x_3) + 2m_v x_1 x_2 x_4 + vI_{YYP} + vm_v x_1^2 \\ & v = -x_3 - x_4 - x_3^2 x_4 + w \\ & w = -K \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{aligned}$$

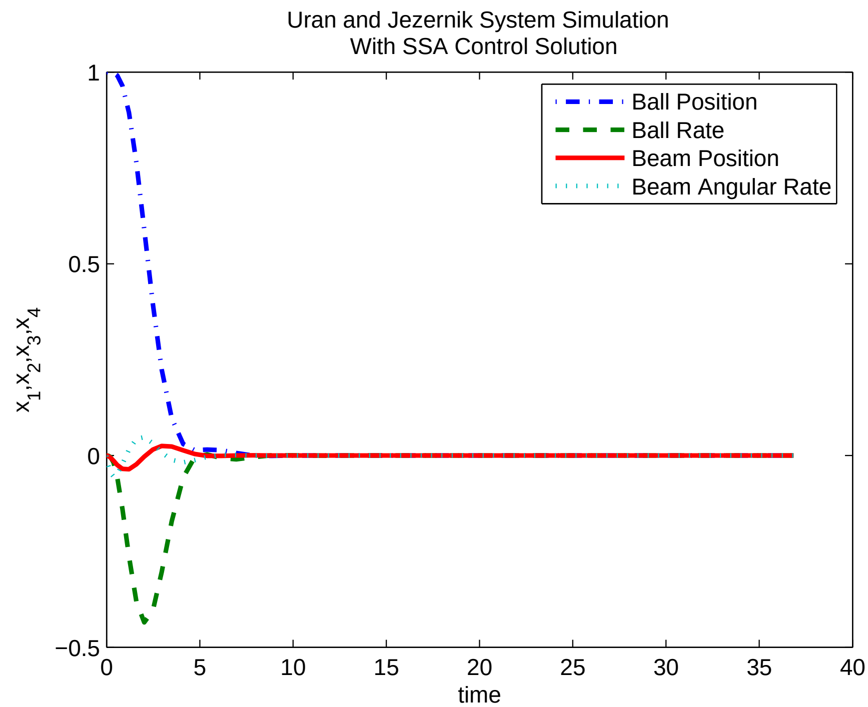


Figure 6.2: Uran and Jezernik System Simulation with SSA control Solution

Uran and Jezernik provided the following parametric values.

$$\begin{aligned}
\text{Beam Length} &= 0.8\ m \\
m_v &= 3.15\ kg \\
I_{YYP} &= 1.248\ kgm^2 \\
\text{Stiction of Beam} &= 1.3\ Nm \\
\text{Stiction of Cart} &= 4.86\ N \\
\text{Coulomb Friction} &\approx 30\%\ \text{of stiction}
\end{aligned}$$

A Matlab Simulink simulation provided the following state responses.

6.1.2 Comparison of State Equations and Control Solution

The state equations used for Uran and Jezernik's paper and this dissertation, chapter 3, are very similar. In fact, they are identical with $I_{ball} \approx 0$.

Uran/Jezernik Equations

$$\dot{x_1} = x_2$$

$$\dot{x_2} = x_1 x_4^2 - g\sin(x_3)$$

$$\dot{x_3} = x_4$$

$$\dot{x_4} = \frac{M_\varphi - 2m_v x_1 x_2 x_4 - m_v g x_1 \cos(x_3)}{\left(m_v x_1^2 + I_{YYP}\right)}$$

Chapter 3 Equations

$$\dot{x_1} = x_2$$

$$\dot{x_2} = C_r\left(m x_1 x_4^2 + mg\sin(x_3)\right)$$

$$\dot{x_3} = x_4$$

$$\dot{x_4} = \frac{\tau - C_d x_4 + g m x_1 \cos(x_3)}{\left(m x_1^2 + I_{beam} + I_{ball}\right)}$$

where $C_r = \left(m + \frac{I_{ball}}{r^2}\right)^{-1}$

Uran and Jezernik used two cascaded computed torque controllers $M_{\varphi,UJ}$ given by the following two equations.

$$\begin{aligned}
M_{\varphi,UJ} &= \left(m_v r^2 + I_{YYP}\right)\boxed{\ddot{\varphi}^z} + 2m_v r\dot{r}\dot{\varphi} + m_v g r\cos(\varphi) \\
&\ddot{\varphi}^z = K_{pP}\left(\varphi_z - \varphi\right) - K_{vP}\dot{\varphi} \\
&\varphi_z = \arcsin\left[\frac{-1}{m_v g}\left(m_v \ddot{r}_z - m_v r\dot{\varphi}^2\right)\right] \\
&\ddot{r}_z = \ddot{r} + K_{vV}\left(\dot{r}_z - \dot{r}\right) + K_{pV}\left(r_z - r\right)
\end{aligned}$$

The SSA control solution $M_{\varphi,SSA}$ uses a cascaded Lyapunov control u_{lyp}, LQR gain control $w = -K\mathbf{x}$, and an invertible transformation.

$$\begin{aligned}
M_{\varphi,SSA} &= \left(m_v r^2 + I_{YYP}\right)\boxed{u_{lyp}} + 2m_v r\dot{r}\dot{\varphi} + m_v g r\cos(\varphi) \\
&u_{lyp} = -\varphi - \dot{\varphi} - \varphi^2\dot{\varphi} + w \\
&w = -K\begin{bmatrix} r \\ \dot{r} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \\
&K = \begin{bmatrix} 0.3162 & 0.7335 & 5.7938 & 2.5620 \end{bmatrix}
\end{aligned}$$

The two control laws are very similar in their structure and require the same state feedback variables. The differential dynamics introduced in Uran and Jezernik's control are a major difference between the two control solutions.

We are working together with Dr. Uran and Dr. Jezernik to implement the SSA-based control solution in Slovenia. After the control is implemented, a comparison will be made with the SSA-based design approach and Uran and Jezernik's design approach. This work should provide great insight into the robustness of the SSA control solution for the ball-on-beam system.

6.2 Analytical Stability

The SSA-based nonlinear design proves effective in all the examples discussed in this dissertation. We presented theoretical analysis in section 4.3 and provided an example to illustrate the SSA method. These results still fail to answer analytical questions which are posed in the SSA-based control solution. Is there an optimal SSA-based design method which will guarantee the "reaching" phase of the control to converge to the "tracking" phase of the control, as discussed in section 4.3? SSA-based control acts to decrease the impact of the system nonlinearities so that a robust control law designed about a given operating point may extend its region of convergence in closed loop operation, but is this extension of the region of convergence guaranteed? Even with these questions unanswered, the SSA-based design approach provides a clear method for designing control solutions.

Bibliography

[1] Jay Farrell, Manu Sharma, and Marios Polycarpou. Backstepping-based flight control with adaptive function approximation. *Journal of Guidance, Control and Dynamics*, 28(6):1089–1101, November-December 2005.

[2] Moacir Rosado Filho and Celso Jose Munaro. A design methodology of tracking controllers for magnetic levitation systems. In IEEE, editor, *Proceedings of the 2001 IEEE International Conference on Control Applications*, pages 47–51, Mexico City, Mexico, September 2001. IEEE, IEEE.

[3] Yi Guo, David J. Hill, and Zhong-Ping Jiang. Global nonlinear control of the ball and beam system. In *Proceedings of the 35th Conference on Decision and Control*, number 3, pages 2818–2823, Kobe, Japan, December 1996. IEEE, IEEE.

[4] John Hauser, Shankar Sastry, and Peter Kokotovic. Nonlinear control via approximate input-output linearization: The ball and beam example. *IEEE Transactions on Automatic Control*, 37(3):392–398, March 1992.

[5] Nam H. Jo and Jin H. Seo. Input output linearization approach to state observer design for nonlinear system. *IEEE Transactions on Automatic Control*, 45(12):2388–2394, December 2000.

[6] Hassan K. Khalil. *Nonlinear Systems*, volume 1. Prentice Hall, Upper Saddle River, NJ 07458, 2nd edition, 1996.

[7] N. Kidane and H. Nakamura Y. Yamashita H. Nishitani. Stabilizability analysis and controller design for a nonlinear system with an input constraint. In *Preceedings of SICE 2003 Annual Conference*, volume 2, pages 1419–1422. IEEE, IEEE, August 2003.

[8] Douglas A. Lawrence. A general approach to input-output pseudolineairzation for nonlinear systems. *IEEE Transactions on Automatic Control*, 43(10):1497–1501, October 1998.

[9] Douglas A. Lawrence. Input-output pseudolinearization on controlled invariant submanifolds. In *Proceedings of the American Control Conference*, pages 555–560, Anchorage, AK, May 2002. IEEE, IEEE.

[10] Douglas A. Lawrence and Wilson J. Rugh. Input-output pseudolinearization. *IEEE Transactions on Automatic Control*, 39(11):2207–2218, November 1994.

[11] Luis A. Marquez Martinez and Claude H. Moog. Input output feedback linearization of nonlinear time-delay systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 4557–4563, Hawaii, December 2003. IEEE, IEEE.

[12] F. Mazenc, R.E. Mahony, and R. Lozano. Forwarding control of scale model autonomous helicopter: a lyapunov control design. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3960–3966, Hawaii, December 2003. IEEE, IEEE.

[13] K.D. Mease, S. Bharadwaj, and S. Iravanchy. Timescale analysis for nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 26(2):318–330, March-April 2003.

[14] Toshiyuki Ohtsuka and Hironori A. Fujii. Design of stabilizing control laws for mechanical systems based on lyapunov's method. *Journal of Guidance, Control, and Dynamics*, 19(1):172–180, January - February 1996.

[15] Shyh-Pyng Shue, M. E. Sawan, and Kamran Rokhsaz. Optimal feedback control of a nonlinear system - wing rock example. *Journal of Guidance, Control, and Dynamics*, 19(1):166–171, January-February 1996.

[16] Sahjendra N. Singh and Marc Steinberg. Adaptive control of feedback linearizable nonlinear systems with application to flight control. *Journal of Guidance, Control and Dynamics*, 19(4):871–877, July-August 1996.

[17] Sahjendra N. Singh, Marc Stienberg, and Robert D. DiGirolamo. Nonlinear predictive control feedback linearizable systems and flight control system design. *Journal of Guidance, Control, and Dynamics*, 18(5):1023 – 1028, September - October 1995.

[18] Hebertt Sira-Ramirez. On the control of the "ball and beam" system: A trajectory planning approach. In *Proceedings of 39th IEEE conference on Decision and Control*, number 4, pages 4042–4047, Sydney, Australia, December 2000. IEEE, IEEE.

[19] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*, volume 1. Prentice Hall, Upper Saddle River, New Jersey, 1991.

[20] T. Urakubo. Discontinuous feedback stabilization of a class of nonholonomic systems based on lyapunov control. In *Proceedings of the Fifth International Workshop on Robot Motion and Control, 2005*, pages 91–96, Kobe Univ., Japan, June 2005. IEEE.

[21] Suzana Uran and Karel Jezernik. Computed torque method for a ball and beam. In *11th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD*, pages 347–352, Balatonfüred, Hungary, June 2002. RAAD.

[22] X. Xia, C.H. Moog, and R. Pothin. Extended output injection and output feedback input output linearization. In *Electronic Letters*, volume 38, pages 200–202. IEEE, February 2002.

Appendices

Appendix A

Five Systems Examples,Code:

```
function  [tspan,xL,xH] = SimAllSys1_5(sys,x1i,x2i);

%  Batch Calculation for all the System's in Chapter 3

% [] = SimAllSys1_5(); %

%sys = 2; x1i = -2.5; x2i = 2;


plotit = 0;

warning off

tspan = [0:0.01:30]';

switch(sys)

    case 1

        %x1i = 1;x2i = 0;

        [tL1,xL1] = ode45(@sys1L,[tspan],[x1i x2i]);

        [tH1,xH1] = ode45(@sys1H,[tspan],[x1i x2i]);

        if(plotit),figure(1)

            for ii=1:max(size(xH1,2),size(xL1,2))

                subplot(max(size(xH1,2),size(xL1,2)),1,ii)

                plot(tspan(1:length(xH1)),xH1(:,ii),...

                tspan(1:length(xL1)),xL1(:,ii))

                if(ii==1),title('system 1'),end

                legend('hybrid','linear')

                ylabel(sprintf('x_%1d',ii))
```

```
        end,pause(0.1)
    [qq,gg] = find(xH1<=0.02*x1i);
    [hh,gg] = find(xL1<=0.02*x1i);
    if(~(length(qq)<1))
    fprintf('Hybrid settling time = %1.2f sec\n',tH1(qq(1)))
    fprintf('Linear settling time = %1.2f sec\n',tL1(hh(1)))
    end
    end
    xL = xL1;
    xH = xH1;
case 2
    %x1i = 0.124;x2i = 0;
    [tL2,xL2] = ode23tb(@sys2L,[tspan],[x1i x2i]);
    [tH2,xH2] = ode23tb(@sys2H2,[tspan],[x1i x2i]);
    if(plotit),figure(2)
        for ii=1:max(size(xH2,2),size(xL2,2))
            subplot(max(size(xH2,2),size(xL2,2)),1,ii)
            plot(tspan(1:length(xH2)),xH2(:,ii),...
            tspan(1:length(xL2)),xL2(:,ii))
            if(ii==1),title('system 2'),end
            legend('hybrid','linear')
            ylabel(sprintf('x_%1d',ii))
        end,pause(0.1)

    [qq,gg] = find(xH2<=0.02*x1i);
    [hh,gg] = find(xL2<=0.02*x1i);
    if(~(length(qq)<1))
```

```
    fprintf('Hybrid settling time = %1.2f sec\n',tH2(qq(1)))
    fprintf('Linear settling time = %1.2f sec\n',tL2(hh(1)))
    end
    end
    xL = xL2;
    xH = xH2;
case 3
    %x1i = 0.1;x2i = 0;
    [tL3,xL3] = ode45(@sys3L,[tspan],[x1i x2i]);
    [tH3,xH3] = ode45(@sys3H,[tspan],[x1i x2i]);
    if(plotit),figure(3)
        for ii=1:max(size(xH3,2),size(xL3,2))
            subplot(max(size(xH3,2),size(xL3,2)),1,ii)
            plot(tspan(1:length(xH3)),xH3(:,ii),...
            tspan(1:length(xL3)),xL3(:,ii))
            if(ii==1),title('system 3'),end
            legend('hybrid','linear')
            ylabel(sprintf('x_%1d',ii))
        end,pause(0.1)
    [qq,gg] = find(xH3<=0.02*x1i);
    [hh,gg] = find(xL3<=0.02*x1i);
    fprintf('Hybrid settling time = %1.2f sec\n',tH3(qq(1)))
    fprintf('Linear settling time = %1.2f sec\n',tL3(hh(1)))
    end
    xL = xL3;
    xH = xH3;
case 4
```

```
    %x1i = 5;x2i = 0;
    x3i = 0;
    [tL4,xL4] = ode45(@sys4L,[tspan],[x1i x2i x3i]);
    [tH4,xH4] = ode45(@sys4H,[tspan],[x1i x2i x3i]);
    if(plotit),figure(4)
        for ii=1:max(size(xH4,2),size(xL4,2))
            subplot(max(size(xH4,2),size(xL4,2)),1,ii)
            plot(tspan(1:length(xH4)),xH4(:,ii),...
            tspan(1:length(xL4)),xL4(:,ii))
            if(ii==1),title('system 4'),end
            legend('hybrid','linear')
            ylabel(sprintf('x_%1d',ii))
        end,pause(0.1)
    [qq,gg] = find(xH4<=0.02*x1i);
    [hh,gg] = find(xL4<=0.02*x1i);
    fprintf('Hybrid settling time = %1.2f sec\n',tH4(qq(1)))
    fprintf('Linear settling time = %1.2f sec\n',tL4(hh(1)))
    end
    xL = xL4;
    xH = xH4;
case 5
    %x1i = 6;x2i = 0;
    x3i = 0;x4i = 0;
    [tL5,xL5] = ode45(@sys5L,[tspan],[x1i x2i x3i x4i]);
    [tH5,xH5] = ode45(@sys5H,[tspan],[x1i x2i x3i x4i]);
    if(plotit),figure(5)
        for ii=1:max(size(xH5,2),size(xL5,2))
```

```
                subplot(max(size(xH5,2),size(xL5,2)),1,ii)
                plot(tspan(1:length(xH5)),xH5(:,ii),...
                tspan(1:length(xL5)),xL5(:,ii))
                if(ii==1),title('system 5'),end
                legend('hybrid','linear')
                ylabel(sprintf('x_%1d',ii))
            end,pause(0.1)
        [qq,gg] = find(xH5<=0.02*x1i);
        [hh,gg] = find(xL5<=0.02*x1i);
        fprintf('Hybrid settling time = %1.2f sec\n',tH5(qq(1)))
        fprintf('Linear settling time = %1.2f sec\n',tL5(hh(1)))
        end
        xL = xL5;
        xH = xH5;
end

%figure(1),hold on,plot(xH2(:,1),xH2(:,2),'b')
figure(1),hold on,plot(xL2(:,1),xL2(:,2),'g')



%************************   SYSTEMS
function dx = sys1H(t,x)
x1 = x(1); x2 = x(2);
Kh = [0.3162    0.2538];
u = -x1 - x2^2 - x2 - Kh*[x1;x2];
dx(1,1) = x2 + tanh(u);
dx(2,1) = x1 + x2^2 + u;
```

```
function dx = sys1L(t,x)

x1 = x(1); x2 = x(2);

K = [1.0477    1.0477];

u = - K*[x1;x2];

dx(1,1) = x2 + tanh(u);

dx(2,1) = x1 + x2^2 + u;


%*************************             2

function dx = sys2H(t,x)

x1 = x(1); x2 = x(2);

Kh = [4.1669    2.0714];

u = -x1-x2^2-x2;%-Kh*[x1;x2]; %-x1^2*x2 extra

dx(1,1) = x1 + x2 +x2^2;

dx(2,1) = x1 + x2^2 + u;


function dx = sys2H2(t,x)

x1 = x(1); x2 = x(2);

Kh = [4.1669    2.0714];

u = -x1-x2^2-x2-x2^5;%-Kh*[x1;x2]; %-x1^2*x2 extra

dx(1,1) = x1 + x2 +x2^2;

dx(2,1) = x1 + x2^2 + u;


function dx = sys2L(t,x)

x1 = x(1); x2 = x(2);

K = [5.3956    3.3002];

u = - K*[x1;x2];
```

```
dx(1,1) = x1 + x2 +x2^2;

dx(2,1) = x1 + x2^2 + u;


%************************              3

function dx = sys3H(t,x)

x1 = x(1); x2 = x(2);

Kh = [0.3162    0.3162];

u = -x2 - Kh*[x1;x2];

dx(1,1) = sin(x2);

dx(2,1) = u;


function dx = sys3L(t,x)

x1 = x(1); x2 = x(2);

K = [0.3162    0.8558];

u = - K*[x1;x2];

dx(1,1) = sin(x2);

dx(2,1) = u;


%************************              4

function dx = sys4H(t,x)

x1 = x(1); x2 = x(2); x3 = x(3);

Kh = [0.3162    4.9487    2.3162];

u = - x3 - Kh*[x1;x2;x3];

dx(1,1) = x2;

dx(2,1) = x2 + sin(x3);

dx(3,1) = u;
```

```
function dx = sys4L(t,x)

x1 = x(1); x2 = x(2); x3 = x(3);

K = [0.3162    4.0279    2.8558];

u = - K*[x1;x2;x3];

dx(1,1) = x2;

dx(2,1) = x2 + sin(x3);

dx(3,1) = u;


%**************************             5

function dx = sys5H(t,x)

x1 = x(1); x2 = x(2);

x3 = x(3); x4 = x(4);

Kh = [0.3162    1.2440    1.2889    0.9178];

u = -x4 - x4*x3^2 - x3 - Kh*[x1;x2;x3;x4];

dx(1,1) = x2;

dx(2,1) = x2*x4^2 + sin(x3);

dx(3,1) = x4;

dx(4,1) = u;


function dx = sys5L(t,x)

x1 = x(1); x2 = x(2);

x3 = x(3); x4 = x(4);

K = [0.3162    1.1969    2.1070    2.0770];

u = - K*[x1;x2;x3;x4];

dx(1,1) = x2;

dx(2,1) = x2*x4^2 + sin(x3);

dx(3,1) = x4;
```

```
dx(4,1) = u;
```

Appendix B

Code for Five System Phase Portrait

```
%  State Space Portrait for 5 different systems

%

%

if(~exist('LinearVal')&~exist('HybridVal')),LinearVal = 0;

HybridVal = 0;end

figurenum = 20;

x2HvecROC = [];x1HvecROC = [];

x2LvecROC = [];x1HvecROC = [];

plotit = 0;

plotPP = 1; % plot phase portrait

skipitH=0;skipitL=0;

ii = 0; jjH=1;jjL=1;

PPL = 0.1; %for system 2, keep below 0.1 pp box, others =0.5

for sys = [2];

    figurenum = figurenum+1;

    for x1i = [-0.6:0.1:0.2];

        for x2i = [-2,2]; %-0.5:0.1:0.5

            [tspan,xL,xH] = SimAllSys1_5(sys,x1i,x2i);

            if(size(xH,1)<2|size(xH,2)<2),skipitH=1;xH=zeros(2,2);

            end

            if(size(xL,1)<2|size(xL,2)<2),skipitL=1;xL=zeros(2,2);
```

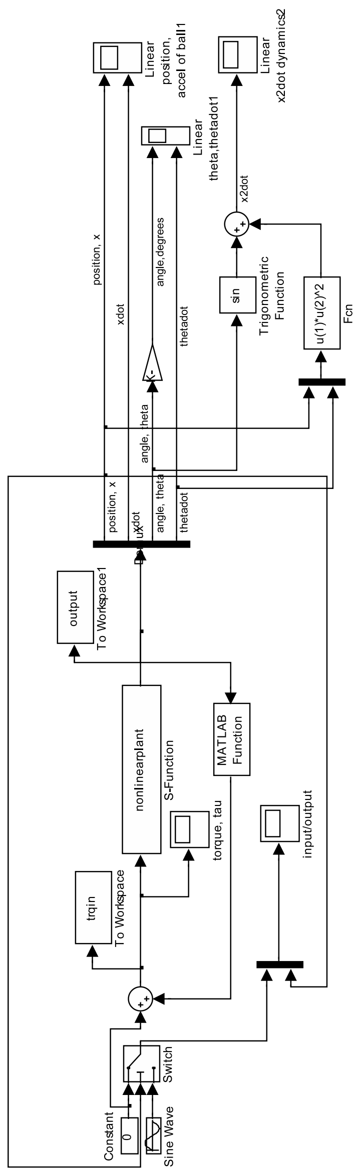
```
end
ii = ii+1;
x1vector(ii) = x1i;
x2vector(ii) = x2i;
v1vectorH(ii) = xH(2,1)-xH(1,1);
v2vectorH(ii) = xH(2,2)-xH(1,2);
v1vectorL(ii) = xL(2,1)-xL(1,1);
v2vectorL(ii) = xL(2,2)-xL(1,2);
if(plotPP),figure(100+figurenum),plot(x1i,x2i,'b.')
    plot(x1i,x2i,'b.'), hold on,
    plot(xH(:,1),xH(:,2),'b'),title('hybrid'),
    hold on, figure(200+figurenum)
    plot(x1i,x2i,'b.'), hold on,
    plot(xL(:,1),xL(:,2),'r:'),
    title('linear'), hold on, pause(0.01)
end
if(abs(xH(size(xH,1),1))<=PPL & ..
 abs(xH(size(xH,1),2))<=PPL & ~skipitH),
  HybridVal = HybridVal + 1; end
if(abs(xL(size(xL,1),1))<=PPL & ..
abs(xL(size(xL,1),2))<=PPL & ~skipitL),
 LinearVal = LinearVal + 1; end
if(abs(xH(size(xH,1),1))<=PPL & ..
abs(xH(size(xH,1),2))<=PPL & plotit & ~skipitH)
    figure(figurenum)
    plot(x1i,x2i,'bx'), hold on, pause(0.01)
    x1HvecROC(jjH) = x1i; x2HvecROC(jjH) = x2i;
```

```
            jjH = jjH+1;
        end
        if(abs(xL(size(xL,1),1))<=PPL & ..
        abs(xL(size(xL,1),2))<=PPL & plotit & ~skipitL)
            figure(figurenum)
            plot(x1i,x2i,'ko'), hold on, pause(0.01)
            x1LvecROC(jjL) = x1i; x2LvecROC(jjL) = x2i;
            jjL=jjL+1;
        end
    end
    clc
    fprintf('Hybrid Value = %f',HybridVal)
    fprintf('\nLinear Value = %f',LinearVal)
    skipitH=0;skipitL=0;
  end
end
quiver(x1vector,x2vector,v1vectorH,v2vectorH,'b'), hold on
quiver(x1vector,x2vector,v1vectorL,v2vectorL,'r')
```

Appendix C

Ball On Beam Example Code/Diagrams

C.1 Block Diagram For Ball-on-beam Model



C.2 S-Function for Nonlinear Plant

```
function [sys,x0,str,ts] = nonlinearplant(t,x,u,flag,a,b)

% s-function [sys,x0,str,ts] = nonlinearplant(t,x,u,flag, ..

%  Makes the input memory

%

%


switch flag,

  case 0, [sys,x0,str,ts]=mdlInitializeSizes(a,b);

  case 1, sys=mdlDerivatives(t,x,u);

  case 2, sys=mdlUpdate(t,x,u);

  case 3, sys=mdlOutputs(t,x,u);

  case 4, sys=mdlGetTimeOfNextVarHit(t,x,u);

  case 9, sys=mdlTerminate(t,x,u);

  otherwise error(['Unhandled flag = ',num2str(flag)]);

end


function [sys,x0,str,ts]=mdlInitializeSizes(a,b);

sizes = simsizes;

sizes.NumContStates  = 4;

sizes.NumDiscStates  = 0; %+ny; % weights of NN are the

% states plus DeltaWstates

% (nx*nx = p)

sizes.NumOutputs     = 4; %+ny;

sizes.NumInputs      = 1; %+ny;

sizes.DirFeedthrough = 1;
```

```
sizes.NumSampleTimes = 1;   % at least one sample time is needed


sys = simsizes(sizes);

x0  = [a b 0 0]; % [zeros(mdem*nx+ny,1)];   % initial conditions

str = [];         % str is always an empty matrix

ts  = [0];  % initialize the array of sample times

return


function sys=mdlDerivatives(t,x,u);

L     = 1;              % Length of Beam

gg    = 9.81;           % gravitational constant

mass  = 0.067;          % mass of the ball

rr    = 0.0127;         % radius of the ball

Iball = 0.4*mass*rr^2;  % moment of inertia of the ball

Ibeam = (1/12)*2*mass*L^2;       % moment of inertia of the beam

Cd    = 10;              % friction coefficient at the pivot




sys(1,1) = x(2);

sys(2,1) = (mass + Iball/rr^2)^(-1) * (mass*x(1)*x(4)*x(4) + ...

mass*gg*sin(x(3)));

sys(3,1) = x(4);

sys(4,1) = (Ibeam + Iball + mass*x(1)*x(1))^(-1) * ...

(u - Cd*x(4) + mass*gg*x(1)*cos(x(3)));


return
```

```
function sys=mdlUpdate(t,x,u);

  sys = [];

return


function y=mdlOutputs(t,x,u);

  % output neural network  output vector and weights

  C = eye(4);

  y = C*x;

return


%

%==================================================

% mdlGetTimeOfNextVarHit

%===================================================

%

function sys=mdlGetTimeOfNextVarHit(t,x,u)


sampleTime = 1;    %  Example, set the next

sys = t + sampleTime;


% end mdlGetTimeOfNextVarHit


%

%======================================================

% mdlTerminate

% Perform any end of simulation tasks.

%======================================================
```

```
%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate
```

C.3 Feedback M File

```
function u = feedback(input,method)
global K

K = [1.0741    0.6168    1.6836    0.3164];
Klin = [1.3867    1.8617   16.1673    0.0230]; % Linear Gain
Khyb = [0.3162    0.7815    4.6581    2.2274]; % Hybrid Gain
x1 = input(1);
x2 = input(2);
x3 = input(3);
x4 = input(4);
g = 9.81;
c1 = (5/7);
c2 = (g)*c1;
c3 = (22/5)*0.0127^2;
m = 0.05;
k1 = (180/pi);
k2 = (180/pi);
```

```
% Control Efforts

switch lower(method)

    case {'linear'}

        u = -Klin*[x1;x2;x3;x4];

    case {'hybrid'}

        y = -Khyb*[x1;x2;x3;x4];

        v = -x4-x4*x3^2-x3 + y;

        u = 10.*x4-.6573*x1*cos(x3)+.1117e-1*v+.6700e-1*v*x1^2;

    case {'inputoutput'}

        v = -1/2*(x4*x2+g*cos(x3))*(sign(x1))*(1/max(x1,1));

        u = - (9.81)*x1*cos(x3)*m+k1*(v)*m*c3 + k2*v*m*x1^2;

    case {'lyapunov'}

        v = max(min(-x4*x2*c1*x1-1/x4*x2*g*c1*sin(x3)...

        -x3,100),-100);

        %max(min(-x2*(x1+c1*x1*x4^2+g*c1*sin(x3))/x4,100),-100);

        u = - (9.81)*x1*cos(x3)*m+k1*(v)*m*c3 + k2*v*m*x1^2;

    otherwise

error('Specify a case: linear,hybrid,inputoutput, or hybridks')

end
```