

AFIT/DS/ENG/98-09

**HIGH AMPLITUDE TRACKING CONTROL  
DISSERTATION**

**Joseph W. McNamee B.S.E.E, M.S.E.E.  
Major, USAF**

**AFIT/DS/ENG/98-09**

Approved for public release; distribution unlimited

19980629 033

## **Disclaimer**

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT/DS/ENG/98-09

# **HIGH AMPLITUDE TRACKING CONTROL**

## **DISSERTATION**

Presented to the Faculty of the Graduate School of Engineering of the Air Force Institute of  
Technology Air University In Partial Fulfillment for the Degree of

**Doctor of Philosophy**

**Joseph W. McNamee B.S.E.E, M.S.E.E.**

**Major, USAF**

Air Force Institute of Technology

Wright-Patterson AFB, Ohio

June, 1998

Sponsored in part by AFRL/VACS

Approved for public release; distribution unlimited

# HIGH AMPLITUDE TRACKING CONTROL

**Joseph W. McNamee B.S., M.S.**

**Major, USAF**

Approved:

M. Pachter

Dr. Meir Pachter  
Committee Chairman

3 June 98  
Date

Sharon A. Heise

Maj. Sharon Heise, Ph. D.  
Committee member

3 JUN 98  
Date

Mark Oxley

Dr. Mark Oxley  
Committee member

3 Jun 98  
Date

Bradley S. Liebst

Dr. Bradley S. Liebst  
Dean's Representative

3 JUN 98  
Date

Accepted:

Robert A. Calico, Jr.

Robert A. Calico, Jr.  
Dean, Graduate School of Engineering

## Acknowledgments

The research reported here would not have been possible without the foundation of knowledge gained through course work. Thus, I would like to thank my many excellent professors who have taught me so well. In particular, I would like to recognize Dr. Maybeck for his stochastic estimation and control sequence, Major Sharon Heise for her optimal control sequence, and Drs. Oxley and Lair for their mathematical analysis and optimization courses. I would also like to thank Major Heise and Dr. Oxley for their support as members of my research committee. Finally, I owe a great debt to my research committee chairman, Dr. Pachter. He possesses a tremendous depth and breadth of knowledge concerning control theory, which I benefited from greatly. Moreover, his knowledge in the area of computational geometry was critical for the completion of this research. Finally, Dr. Pachter's wisdom in everyday matters makes working with him a real pleasure. In closing I would like to express my appreciation to my family for their patience and support over these past several years. Thank you Cindy for accepting the extra work load and letting me work all hours of the day and night. Also, thank you, Chelsea, Mackenzie, and Ryan for being the wonderful children that you are.

# Table Of Contents

	Page
Acknowledgments .....	iv
Table Of Contents .....	v
List of Figures .....	ix
List of Tables .....	xiv
Abstract .....	xv
Chapter 1. INTRODUCTION .....	1
1.1 Overview .....	1
1.2 Previous Work .....	3
1.2.1 Performance Enhancement Methods .....	4
1.2.2 Constraint Avoidance Methods .....	5
1.2.3 Invariance Based Constraint Avoidance .....	8
1.2.4 Summary .....	14
1.3 Problem Statement .....	15
1.4 Approach .....	15
1.5 Scope and Assumptions .....	16
1.6 Organization .....	17
1.7 Summary .....	18
Chapter 2. STATIC ADMISSIBILITY AND INVARIANCE CONCEPTS .....	19

2.1	Overview .....	19
2.2	Problem formulation and Definitions .....	19
2.3	Maximal Output Admissible Sets .....	24
2.3.1	Discrete Time Reference Governor .....	27
2.3.1.1	Scalar Example .....	31
2.3.2	Output Admissible Sets Versus Statically Admissible Sets .....	34
2.4	Summary .....	37
Chapter 3.	PROJECTION OF POLYTOPES ONTO A SUBSPACE .....	38
3.1	Overview .....	38
3.2	Projection of the Vertices .....	39
3.3	Recursive Convex Hull Algorithm .....	40
3.3.1	Numerical Precision Considerations .....	51
3.4	Two Dimensional Problem .....	54
3.4.1	The Maximal Statically Admissible Set .....	60
3.5	Summary .....	68
Chapter 4.	CONSTRAINT AVOIDANCE METHODS .....	69
4.1	Overview .....	69
4.2	Constraint Avoidance Strategy .....	69
4.3	Determination of the Modified Reference Signal .....	71
4.3.1	Control Concepts .....	73

4.4	Simulation Results . . . . .	74
4.4.1	Example One . . . . .	76
4.4.2	Example Two . . . . .	85
4.4.3	Example Three . . . . .	93
4.5	Constraint Avoidance Using Subsets of $X_s^\varepsilon$ . . . . .	96
4.5.1	Constraint Avoidance Using Polyhedral Subsets . . . . .	100
4.5.1.1	Simulation Results . . . . .	103
4.5.2	Constraint Avoidance Using Ellipsoidal Subsets . . . . .	111
4.5.2.1	Simulation Results . . . . .	112
4.6	Summary . . . . .	112
Chapter 5.	FLIGHT CONTROL APPLICATION . . . . .	116
5.1	Overview . . . . .	116
5.2	Aircraft Model and Constraints . . . . .	116
5.3	Linear Inequality Constraints . . . . .	119
5.3.1	Validation . . . . .	120
5.3.2	Polyhedral Subset Comparisons . . . . .	123
5.4	Simulation Results . . . . .	126
5.5	Summary . . . . .	129
Chapter 6.	THE CONSTRAINED REGULATOR CONTROL PROBLEM . . . . .	133
6.1	Overview . . . . .	133



6.2	Reference Signal Generator .....	133
6.3	Second-Order Example .....	136
6.4	Simulation Results .....	139
6.5	Summary .....	140
Chapter 7.	CONCLUSION .....	146
7.1	Overview .....	146
7.2	Contributions and Accomplishments .....	146
7.3	Research Conclusions .....	151
7.3.1	Recommendations .....	153
Appendix A.	RECURSIVE CONVEX HULL ALGORITHM RESULTS .....	154
A.1	Recursive Convex Hull Algorithm and Brute Force Algorithm Results Comparison .....	154
A.2	Results for Selected Subsets of Vertices .....	158
Appendix B.	SELECTED SCRIPT FILES .....	162
B.1	Script Files for the Second Order Problem .....	162
B.2	Script Files for the Fourth Order Problem .....	170
Bibliography	.....	186
Vita	.....	188

## List of Figures

		Page
Figure 1.	Dual-loop tracking controller architecture . . . . .	5
Figure 2.	Cartesian product space, $V$ , with arrows indicating the sign of the derivative of $x$ . . . . .	11
Figure 3.	State response to a statically admissible pulse. . . . .	35
Figure 4.	Modified reference signal response to a statically admissible pulse. . . . .	35
Figure 5.	Control signal response to a statically admissible pulse. . . . .	35
Figure 6.	State response to a statically inadmissible pulse. . . . .	36
Figure 7.	Modified reference signal response to a statically inadmissible pulse. . . . .	36
Figure 8.	Control signal response to a statically inadmissible pulse. . . . .	36
Figure 9.	Initial tetrahedron for a problem in $E^3$ . . . . .	44
Figure 10.	Incorporation of $v_{k+1}$ for an $E^2$ arrangement. . . . .	46
Figure 11.	Invalidation of multiple hyperplanes for a problem in $E^3$ . . . . .	46
Figure 12.	Incorporation of $v_8$ for the $E^3$ example. . . . .	48
Figure 13.	A set of points that are not in general position in $E^3$ . . . . .	52
Figure 14.	A nonsimplicial polyhedron in $E^3$ . . . . .	52
Figure 15.	Statically admissible sets $X_s(0)$ and $X_s(0.9)$ for the 2D problem. . . . .	58
Figure 16.	Extremal trajectories for several values of $r_o$ for the 2D problem. . . . .	58
Figure 17.	Convex hulls of $X_s$ and $X_s(0)$ for the 2D problem. . . . .	59

	Page
Figure 18. Maximal positively invariant set for the 2D problem . . . . .	61
Figure 19. Vertices of $X_s^\varepsilon$ for the 2D problem. . . . .	63
Figure 20. Supporting hyperplanes for $X_s^\varepsilon$ for the 2D problem. . . . .	65
Figure 21. Supporting hyperplanes for $X_s^{\delta_1}$ for the 2D problem. . . . .	65
Figure 22. Supporting hyperplanes for $X_s^{\delta_2}$ for the 2D problem. . . . .	67
Figure 23. Nonlinear control system. . . . .	75
Figure 24. System response for Example 1a with no constraint avoidance measures. . . . .	79
Figure 25. Control signal response for Example 1a with no constraint avoidance measures. . . . .	79
Figure 26. MDRSG state trajectory for Example 1a. . . . .	80
Figure 27. MDRSG control signal response for Example 1a. . . . .	80
Figure 28. MDRSG reference signal constraints for Example 1a . . . . .	81
Figure 29. Modified reference signal response for Example 1a. . . . .	81
Figure 30. SRSR system control signal response for Example 1a. . . . .	82
Figure 31. Comparison of DTRG and MDRSG tracking performance for Example 1a. . . . .	82
Figure 32. DTRG control signal response for Example 1a. . . . .	83
Figure 33. MDRSG state trajectory for Example 1b. . . . .	86
Figure 34. MDRSG control signal response, statically inadmissible input for Example 1b. . . . .	86
Figure 35. MDRSG reference signal constraints, statically inadmissible input . . . . .	87

	Page
Figure 36. Modified reference signal response for Example 1b. . . . .	87
Figure 37. Expanded views of modified reference signal response for Example 1b. . . . .	88
Figure 38. Unconstrained system response to a unit step input for Example 2. . . . .	90
Figure 39. Supporting hyperplanes of $X_s^\epsilon$ for Example 2. . . . .	90
Figure 40. Comparison of $X_s^\epsilon$ obtained for Examples 1 and 2. . . . .	91
Figure 41. Small signal tracking performance comparison for Example 2. . . . .	91
Figure 42. SRSG control signal response to a 0.1 amplitude step input for Example 2. . . . .	92
Figure 43. Comparison of SRSG, DTRG, and LTI system tracking performances for Example 2. . . . .	94
Figure 44. System response to a slewing maneuver without constraint avoidance for Example 2. . . . .	94
Figure 45. Comparison of DTRG, MDRSG, and SRSG responses to a slewing maneuver for Example 2. . . . .	95
Figure 46. SRSG control signal response to a slewing maneuver for Example 2. . . . .	95
Figure 47. Comparison of the unconstrained system responses to a unit step input for Examples 2 and 3. . . . .	97
Figure 48. Supporting hyperplanes for $X_s^\epsilon$ for Example 3. . . . .	97
Figure 49. Comparison of $X_s^\epsilon$ for Examples 1, 2, and 3. . . . .	98
Figure 50. Small signal tracking performance comparison. . . . .	98
Figure 51. Slewing maneuver tracking performance comparison. . . . .	99
Figure 52. State trajectory for Example 4a. . . . .	105

	Page
Figure 53. Control signal response for Example 4a. . . . .	105
Figure 54. Reference signal constraints for Example 4a. . . . .	106
Figure 55. Modified reference signal response for Example 4a. . . . .	106
Figure 56. State trajectory for Example 4b. . . . .	107
Figure 57. Control signal response for Example 4b. . . . .	107
Figure 58. Reference signal constraints for Example 4b. . . . .	108
Figure 59. Modified reference signal response for Example 4b. . . . .	108
Figure 60. Inconsistent constraints for Example 4b. . . . .	109
Figure 61. State trajectory resulting from inconsistent constraints for Example 4b. . . . .	110
Figure 62. State trajectory for Example 5. . . . .	113
Figure 63. Control signal response for Example 5. . . . .	113
Figure 64. Reference signal constraints for Example 5. . . . .	114
Figure 65. Reference signal response for Example 5. . . . .	114
Figure 66. Unconstrained system responses to a 0.1 amplitude step. . . . .	127
Figure 67. Constrained system responses to a 0.1 amplitude step. . . . .	127
Figure 68. SRSR and DTRG system responses to a 0.1 amplitude step. . . . .	128
Figure 69. MDRSG tracking performance comparison for a 1.0 amplitude step. . . . .	130
Figure 70. Comparison of admissibility constraints. . . . .	130
Figure 71. Comparison of MDRSG and DTRG performance . . . . .	131

	Page
Figure 72. Nonlinear dual-loop controller architecture. . . . .	141
Figure 73. Constrained regulator control problem. . . . .	141
Figure 74. $X_r$ for the second-order regulator control system. . . . .	142
Figure 75. Nonlinear dual-loop controller architecture. . . . .	142
Figure 76. Supporting hyperplanes for $X_s^\epsilon$ . . . . .	143
Figure 77. Comparison of $X_r$ and $X_s$ . . . . .	143
Figure 78. Dual-loop control system response for $x(0) = [0.9, -0.5]^T$ . . . . .	144
Figure 79. Responses of $u$ and $r'$ for $x(0) = [0.9, -0.5]^T$ . . . . .	144
Figure 80. Regulator control system response for $x(0) = [0.9, -0.5]^T$ when saturation mitigation is not employed. . . . .	145
Figure 81. Fourteen element subset of vertices . . . . .	155
Figure 82. Recursive convex hull algorithm results, 14 element subset of vertices . . . . .	156
Figure 83. Brute force algorithm results, 14 element subset of vertices . . . . .	157
Figure 84. 18 element subset of vertices . . . . .	159
Figure 85. 20 element subset of vertices . . . . .	159
Figure 86. List of vertices / facet for the 18 element subset of vertices. . . . .	160
Figure 87. List of vertices / facet for the 20 element subset of vertices. . . . .	161

## List of Tables

	Page
Table 1. Numerical precision analysis . . . . .	55
Table 2. Recursive edging algorithm data . . . . .	122
Table 3. Results for several subsets of $V$ . . . . .	125

## Abstract

The problem of tracking control in the face of high amplitude dynamic reference signals, hard state and control constraints, and open-loop unstable plants is investigated. The stated problem is addressed using a nonlinear dual-loop controller architecture. The inner-loop contains the controlled process which consists of the bare plant and a predetermined linear control law that provides good small signal performance in the absence of saturation. The controlled process is augmented with an outer supervisory loop which contains a nonlinear reference signal governor. The purpose of the reference signal governor is to modify the exogenous reference signal so that the controlled process state and control constraints are not violated. The proposed methodology is based on a discrete time formulation. In contrast to current reference signal governor methods a BIBO stable system is obtained without restricting the modified reference signal to statically admissible values. This allows more aggressive control concepts which improve tracking performance. An arbitrarily close approximation to the controlled process' maximal statically admissible set is characterized with a finite set of linear inequalities. These linear inequalities are exploited by the on-line reference signal governor to restrict the controlled process state to a positively invariant set without restricting the modified reference signal to statically admissible values. The linear inequalities are generated from a set of vertices using a new computationally efficient recursive convex hull algorithm.

The nonlinear dual-loop controller architecture is also applied to the discrete time constrained regulator problem. In this case the outer-loop nonlinearity acts as a reference signal generator. By synthetically generating non-zero modified reference signals, when required, the volume of the controlled process' maximal statically admissible set is dramatically increased. This improves the robustness of the controlled process with respect to external disturbances, and alleviates the trade-off between small signal performance and the size of the positively invariant set.



# HIGH AMPLITUDE TRACKING CONTROL

## Chapter 1 - Introduction

### 1.1 Overview

Real world dynamical control systems are generally subject to hard nonlinearities. For example, aircraft control surface actuators are subject to hard displacement and rate constraints. These physical constraints translate into hard state (actuator rate) and control (actuator displacement) constraints on linear system models of the physical systems. While state and control constraints are most often due to actuator displacement and rate limits, they can also result from physical limits on other system parameters as well. Linear system models cannot account for hard state and control constraints. As a result, linear controller design methodologies often produce feedback controllers which generate infeasible or inadmissible control signals for relatively small state and reference signal perturbations. This in turn can lead to windup of dynamic compensation, degraded closed-loop system performance, and even unstable closed-loop system responses. This problem is exacerbated by high gain controllers, high amplitude dynamic reference signals, and open-loop unstable plants.

Optimal and robust control methodologies often lead to high gain controllers. This is especially true when tight tracking specifications are coupled with robustness requirements, viz., robust performance requirements [27]. In the absence of state and control constraints these high gain controllers result in closed-loop systems with admirable performance and robustness characteristics. However, due to high transmission loop gains, even small perturbations may result in infeasible control signals. Then, actual implementation of these controllers requires significant post-design ad hoc modifications. As a result, small signal performance is degraded, and closed-loop system stability is generally difficult or impossible to guarantee. Actuator saturation is not limited to high-

gain feedback control systems. Even low-gain tracking control systems may experience periods of saturation during slewing operations. Thus, high amplitude dynamic reference signals may also result in saturation.

Most investigations have focused on open-loop stable plants subject to actuator displacement constraints. In these cases conditions can be devised to obtain global stability in the face of actuator saturation and windup [23]. However, open-loop unstable plants require feedback control for closed-loop system stability. In this case, violation of hard control and state constraints opens the feedback loop. This in turn results in open-loop operation of the plant, and an unstable control system response. While the effects of constraint violation are magnified in the case of open-loop unstable plants, it is important to note that even control systems with open-loop stable plants can produce unstable system responses in the presence of state and/or control constraint violations.

State and control signal saturations may be categorized as either transient or steady state in nature. Transient saturations are those state and control constraint violations that occur in the course of a system's transient response to a disturbance or exogenous input, and result from initial large errors or overshoots associated with underdamped systems. Steady state saturations are those constraint violations associated with inadmissible equilibrium points. An equilibrium point is inadmissible if it can not be maintained without violation of a system's state and/or control constraints. Thus, a constant exogenous reference signal that attempts to drive a system to an inadmissible equilibrium point will result in steady state saturation. In this case, the exogenous reference signal is said to be statically inadmissible. It is important to note that transient saturations can occur even when the initial system state, final system state, and exogenous reference signal are all statically admissible.

Saturation effects mitigation is a current research topic, and a large body of literature exists which addresses this problem. However, with a few notable exceptions, little work has been accomplished which is applicable to the problem of high amplitude dynamic reference signal tracking

control in the face of hard state and control constraints and open-loop unstable plants. Moreover, methods that do address the stated problem generally limit the exogenous reference signal to statically admissible values to obtain BIBO stability guarantees. The concepts of static admissibility and positively invariant sets play important roles in these methodologies, and in the methodology proposed here.

Important results of this research include the development of a reference signal governor which produces a BIBO stable closed-loop system, but does not restrict the modified reference signal to statically admissible values. Also, a method to reduce on-line computational burden is developed. The reference governor system architecture is also applied to the constrained regulator problem. This novel approach can substantially increase the size of the statically admissible set of initial states, for a given control law, without degrading small signal performance.

## **1.2 Previous Work**

Mitigation of state and control constraint violation effects is an active research area, and numerous controller design and modification methodologies exist which address linear time invariant (LTI) systems with state and / or control constraints. Both linear and nonlinear feedback control law methodologies have been proposed. Linear control law paradigms are generally limited to the constrained regulator and set point control problems. Even then, linear control laws provide very conservative results since small signal performance must be sacrificed to insure constraint violation is avoided for large initial disturbances [7]. Nonlinear control laws have been obtained by embedding the stated problem in an optimal control paradigm [2], [7], [13], [15], [25]. However, the resulting bang-bang controllers are impractical to implement, and are overly sensitive to modeling errors [11]. Thus, many ad hoc constraint effects mitigation methodologies have been developed.

Constraint effects mitigation methodologies may be classified as either those that allow constraint violation, or saturation, to occur, and those that avoid constraint violation completely.

### **1.2.1 Performance Enhancement Methods**

Those constraint effects mitigation methods that allow saturation are referred to here as performance enhancement techniques. These methods improve closed-loop system performance by reducing the number and duration of periods of saturation. Performance enhancement methods are generally limited to systems with open-loop unstable plants. In these cases conditions can be devised, using the generalized Nyquist criterion developed in [23], so that global BIBO stability is guaranteed..

Anti-windup methods may be the largest class of performance enhancement schemes. Windup refers to growth of the controller states during periods of actuator saturation. During periods of actuator saturation the controller states are driven by the error signal while the constrained control signal remains constant. Thus, the error signal ceases to have an impact on the plant input. Moreover, when the error signal is removed, or changes sign, a period of time is required for the controller states to decay, or unwind. Many anti-windup methods are based on the concept of conditional or intelligent integration [1], [12], [14]. Often these methods involve feeding back the difference between the actual and commanded control signal.

The purpose of anti-windup and other ad hoc methods is to improve closed-loop system performance by reducing the number and duration of actuator saturations. Since they do not prevent constraint violations completely these methods are not applicable to control systems with open-loop unstable plants. Constraint violation in this case will result in an unstable system response. Thus, constraint avoidance methods must be considered.

### 1.2.2 Constraint Avoidance Methods

A Linear programming approach to constraint avoidance is proposed in [6] and [22]. A Linear Programming methodology is used to translate actuator constraints into constraints on the exogenous reference signal. Then a nonlinear reference signal governor generates a modified reference signal based on the state of the controlled process and the exogenous reference signal. This results in a dual-loop controller architecture with an the inner linear feedback loop and an outer nonlinear supervisory loop, Fig. 1.

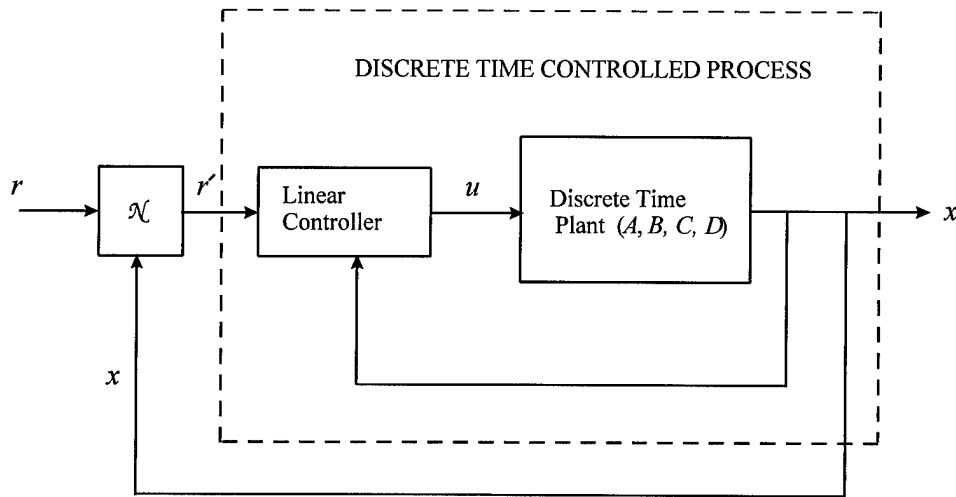


Figure 1. Dual-loop tracking controller architecture

In [17], [18], [19], and [21] the constrained tracking control problem is addressed using reference signal extrapolation and a receding horizon optimal control methodology. Polynomial extrapolation and interpolation is used to obtain the predicted reference signal over the optimization horizon. Then, within each Receding Horizon Window (RHW), the predicted reference signal vector,  $\hat{r}$ , is known, so that the LQ optimal control methodology may be applied to each of the finite horizon control problems. The resulting finite horizon LQ optimal control problem is solved within each RHW. In this way the indefinite horizon control problem is broken up into an open ended se-

quence of finite horizon control problems of length  $N$ . Where  $N$  is the number of samples within each RHW.

In [17] it is also shown that the optimal control vector,  $\mathbf{u}^* = [u_o^*, u_1^*, \dots, u_{N-1}^*]$ , obtained within each RHW is a linear function of the predicted reference vector,  $\hat{\mathbf{r}} = [r_1, \hat{r}_2, \dots, \hat{r}_N]$ , and the initial plant state,  $\mathbf{x}_o$ . This allows transformation of the actuator constraints into reference vector constraints. Moreover, the predicted reference vector may be written in terms of  $r_1$ , the current pilot demanded reference signal, so the actuator constraints can be transformed into constraints on  $r_1$ . Note that  $r_1$  is the actual pilot demanded reference signal, not a predicted value. Now, if  $r_1$  violates any of the transformed constraints it may be optimally scaled such that the modified reference signal,  $r'_1$ , satisfies all constraints. By optimally scaled, we mean that  $r'_1$  is chosen so that  $|r_1 - r'_1|$  is minimized subject to the transformed actuator constraints. If  $r_1$  satisfies all constraints then clearly  $r'_1 = r_1$ . Thus, small signal operation is not sacrificed.

In accordance with the receding horizon modulus operandi,  $r_1$  (or  $r'_1$ ) is the only reference value actually tracked, and  $u_o$  is the only control which is actually applied to the system within each RHW. Thus, many of the constraints on  $r_1$  are induced by constraints on subsequent elements of the predicted reference vector,  $(\hat{r}_2, \hat{r}_3, \dots, \hat{r}_N)$  which are not realized. Since enforcement of these "down-stream" constraints may lead to an overly conservative scaling of  $r_1$ , in [17] feasibility criteria are proposed which do not necessarily include all down-stream constraints. The resulting control law depends on the reference vector prediction and feasible reference vector determination methods as well as the selection of feasibility criteria. In particular, BIBO stability is only guaranteed under certain conditions, one of which is that the modified reference signal be limited to statically admissible values.

The concepts of static admissibility and positive invariance play central roles in many constraint avoidance techniques. In [4], [26], [5], [3], and [10] the constrained regulator problem

is addressed using the concept of positively invariant sets. In these investigations both quadratic and non-quadratic Lyapunov functions are used to obtain a state feedback matrix,  $k_x$ , for which a predetermined set of initial states,  $X_o \subset \mathfrak{R}^n$ , is contained in a set,  $W \subset \mathfrak{R}^n$ , which is admissible and positively invariant with respect to the closed-loop system. While this method does avoid constraint violation for all initial states in  $W$ , it is overly conservative in that for a given  $k_x$  and  $W$  determined in this manner, larger positively invariant sets generally exist. This is because the existence of a Lyapunov function is a sufficient but not necessary condition for asymptotic stability. More importantly, these methods provide conservative results because only linear control laws are considered. Thus, a trade-off must be made between the size of  $W$  and small signal performance [7]. In the sequel this inherent trade-off is alleviated by implementing a non-linear dual-loop controller architecture.

In [8] and [9] maximal output admissible sets are characterized for discrete-time systems subject to state and control constraints. Under appropriate conditions maximal output admissible sets may be characterized by a finite number of inequality constraints. Moreover, if the constraint set is polyhedral, then the inequalities are linear, and the maximal output admissible set may be determined by solving a set of Linear Programming problems [8]. In [9] the concept of maximal output admissible sets is applied to the development of a discrete time reference governor (DTRG). The DTRG is a low-pass filter with a variable bandwidth parameter,  $\lambda \in [0, 1]$ , which generates a modified reference signal to avoid saturation of state and control constraints. A dual-loop controller architecture is used, as in Fig. 1, which consists of a linear inner feedback controller loop, and an outer nonlinear supervisory loop. At each time increment,  $t$ , the modified reference signal,  $r'(t)$ , is chosen such that it is statically admissible. While the DTRG control concept eliminates saturation and results in a BIBO stable closed-loop system, it is somewhat conservative in that it also restricts the modified reference signal to statically admissible values. However, there are certainly cases where a reference input that is not statically admissible over a finite time interval would not neces-

sarily result in saturation. Thus, achievable tracking performance is sacrificed to insure constraint violation does not occur.

The results of [8] and [9] play an important role in the methodology developed here. Moreover, the DTRG methodology is a valid benchmark since it exploits the augmented system's maximal output admissible set, and the on-line algorithm is computationally efficient. Because of their importance to this work, the relevant results of [8] and [9] are presented in detail in Chapter 2.

### 1.2.3 Invariance Based Constraint Avoidance

In [17] it is shown that a sufficient condition for constraint avoidance and BIBO stability is that the constrained control system state vector be restricted to an admissible positively invariant set. In particular, Miller shows that it is not necessary to explicitly restrict the exogenous reference signal to statically admissible values to obtain a BIBO stable closed-loop system. The construction of admissible invariant sets is a crucial step in the design of tracking control laws. Our insights into the critical role played by the above mentioned invariant sets lets us focus on design for tracking performance, while at the same time guaranteeing the BIBO stability of the control system. Miller demonstrates the application of admissible positively invariant sets to saturation avoidance and BIBO stability enforcement with the following scalar example.

The scalar constrained control system given by

$$\dot{x} = ax + bu, \quad x(0) = x_o, \quad (1)$$

$$u = k_x x + k_r r', \quad -1 \leq u \leq 1 \quad (2)$$

is considered. Combining (1) and (2) gives,

$$\dot{x} = a_{cl}x + bk_r r', \quad a_{cl} = a + bk_x \quad (3)$$

Now, choose  $k_r$  such that tracking is enforced, viz.,

$$k_r = -\frac{a}{b} - k_x$$



Then,

$$\dot{x} = a_{cl}(x - r'), \quad (4)$$

and

$$u = k_x x - \frac{a_{cl}}{b} r' \quad (5)$$

The dual-loop controller architecture of Fig. 1 is implemented using a static nonlinearity,  $N$ , whose synthesis is outlined here. Unlike the DTRG constraint avoidance method, the controlled process state is restricted to an admissible positively invariant set without restricting the modified reference signal to statically admissible values. The flexibility afforded by this less conservative and more general approach makes it possible to consider two separate tracking control concepts.

The first tracking control concept, outlined in [17], is  $\min_{r'} |r - r'|$  subject to the control constraints. The second control concept for the selection of  $r'$ , outlined in [16], attempts to drive  $x$  to  $r$  as quickly as possible, subject to the control constraints.

**Control Concept 1:** Chose  $r'$  such that  $|r - r'|$  is minimized subject to the control constraints.

Assuming  $k_x < 0$ ,  $a_{cl} < 0$ , and  $b > 0$  results in the explicit (nonlinear) control law

$$r'(x, r) = \begin{cases} r & , \quad \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \leq r \leq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} \\ \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} & , \quad r \geq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} \\ \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} & , \quad r \leq \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \end{cases} \quad (6)$$

and substituting (6) into (4) results in the closed-loop system

$$\dot{x} = \begin{cases} a_{cl}(x - r), & \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \leq r \leq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} \\ ax + b, & r \geq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} \\ ax - b, & r \leq \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \end{cases} \quad (7)$$

The saturation avoidance control law (6) does not guarantee BIBO stability in the case of an open-loop unstable plant. However, we can obtain the desired BIBO stability through an additional invariance requirement.

A bounded set  $X_I \subset X = \{x : x \in \mathfrak{R}\}$  is invariant with respect to the system given by (1), (2), and (7) if and only if on the boundary of  $X_I$ ,  $x\dot{x} \leq 0$ . Thus, if we can characterize a bounded

invariant set

$$X_I = \left\{ x \in \mathfrak{R} : \exists r' \text{ s.t. } -1 \leq u = k_x x - \frac{a_{cl}}{b} r' \leq 1 \text{ and } \dot{x} \leq 0 \right\}$$

then, by restricting  $x$  to  $x \in X_I$ , we can guarantee BIBO stability. First, consider the case of an open-loop unstable plant,  $a > 0$ . For  $\frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \leq r \leq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}$  we have  $r' = r$ , and  $\dot{x} = a_{cl}(x - r)$ . Hence,  $r < x$  results in  $\dot{x} < 0$ , and  $r > x$  results in  $\dot{x} > 0$ . Second, for  $r \geq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}$  we have  $\dot{x} = ax + b$ . In this case  $x > -\frac{b}{a}$  results in  $\dot{x} > 0$ , and  $x < -\frac{b}{a}$  results in  $\dot{x} < 0$ . Finally, for  $r \leq \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}$  we have  $\dot{x} = ax - b$ . Therefore,  $x > \frac{b}{a}$  results in  $\dot{x} > 0$ , and  $x < \frac{b}{a}$  results in  $\dot{x} < 0$ . This is summarized in Fig. 2 where the directions of the arrows represent the sign of  $\dot{x}$  in the Cartesian product space,  $V$ , defined by

$$V = \left\{ v \in \mathfrak{R}^2 : v = [r, x]^T, r, x \in \mathfrak{R} \right\}$$

From Figure 2 and the above discussion it is clear that saturation avoidance requires that

$$\frac{(k_x x(t) - 1)b}{a_{cl}} \leq r'(t) \leq \frac{(k_x x(t) + 1)b}{a_{cl}}$$

In addition, if the control law allows  $|x| > \frac{b}{a}$  the system will diverge due to the constraint on the control signal,  $u$ . Thus, saturation avoidance alone will not result in a BIBO stable system. To obtain BIBO stability we must restrict  $x$  to the set

$$X_I = \left\{ x \in \mathfrak{R} : |x| \leq \frac{b}{a} \right\}$$

Also, if the system ever achieves either of the valid equilibrium points  $v = [\pm \frac{b}{a}, \pm \frac{b}{a}]^T$  it becomes “stuck”. Thus, in addition to control law (6), we must limit  $x$  such that

$$x \in X_I^\varepsilon = \left\{ x \in \mathfrak{R} : -\frac{b}{a} + \varepsilon \leq x \leq \frac{b}{a} - \varepsilon \right\} \quad (8)$$

then saturation will be avoided, and the system output will be bounded. Hence, the control law (6) is modified as follows.

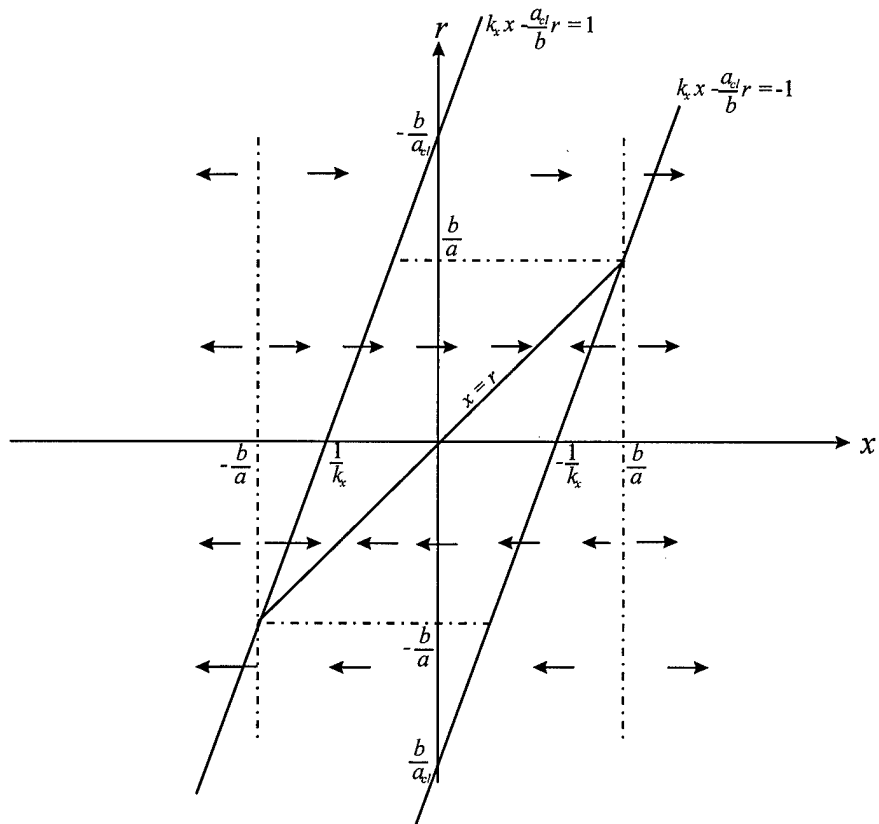


Figure 2. Cartesian product space,  $V$ , with arrows indicating the sign of the derivative of  $x$

$$r'(x, r) = \begin{cases} r, & \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \leq r \leq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, -\frac{b}{a} + \varepsilon \leq x \leq \frac{b}{a} - \varepsilon \\ \frac{b}{a} - \varepsilon, & x > \frac{b}{a} - \varepsilon, r \geq \frac{b}{a} - \varepsilon \\ -\frac{b}{a} + \varepsilon, & x < -\frac{b}{a} + \varepsilon, r \leq -\frac{b}{a} + \varepsilon \\ \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, & r > \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, x \leq \frac{b}{a} - \varepsilon \\ \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}, & r \leq \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}, x \geq -\frac{b}{a} + \varepsilon \end{cases} \quad (9)$$

Now, under the explicit (nonlinear) control law (9),  $x \in X_I$  is enforced, the system will not become “stuck” at  $x = \pm \frac{b}{a}$ , and saturation avoidance plus invariance in a bounded subset of the state space yields the BIBO stability guarantee.

A similar analysis for the case of an open-loop stable plant,  $a < 0$ , shows that the saturation avoidance set is  $X_I = \mathfrak{R}^1$ . Hence, the simpler control law (6) yields global BIBO stability in this case.

**Control Concept 2:** Chose  $r'$  so as to either maximize or minimize  $\dot{x}$ , based on the sign of  $r - x$ , and subject to the control constraints; thus, maximizing the instantaneous reduction in tracking error. Also, if  $r = x$  then minimize  $\dot{x}^2$ , subject to the control constraints in eq. (2). Thus,

$$\begin{aligned} \text{If } r > x \text{ then } \max_{r'} (\dot{x}) \\ \text{If } r < x \text{ then } \min_{r'} (\dot{x}) \\ \text{If } r = x \text{ then } \min_{r'} (\dot{x}^2) \end{aligned}$$

From (4) and the assumption that  $a_{cl} < 0$ , this results in

$$\begin{aligned} \text{If } r > x \text{ then } \max r' \\ \text{If } r < x \text{ then } \min r' \\ \text{If } r = x \text{ then } \min (x - r')^2 \\ \text{subject to } \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}} \leq r' \leq \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}} \end{aligned} \quad (10)$$

Hence, the explicit control law is finally obtained

$$r' = \begin{cases} \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, & r > x \\ \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}, & r < x \\ r, & r = x, -\frac{b}{a} \leq x \leq \frac{b}{a} \\ \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, & r = x, x < -\frac{b}{a} \\ \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}, & r = x, x > \frac{b}{a} \end{cases} \quad (11)$$

and substituting (11) into (4) results in the “closed-loop” system,

$$\dot{x} = \begin{cases} ax + b, & r > x \\ ax - b, & r < x \\ 0, & r = x, -\frac{b}{a} \leq x \leq \frac{b}{a} \\ ax + b, & r = x, x < -\frac{b}{a} \\ ax - b, & r = x, x > \frac{b}{a} \end{cases} \quad (12)$$

If the open-loop system is stable, viz.,  $a < 0$ , then control law (11) results in a globally BIBO stable closed-loop system. However, if the open-loop system is unstable,  $a > 0$ , then, as before, we must restrict  $x$  such that  $x \in X_I$ , where  $X_I$  is given by (8). In this case, control law (11) is modified as follows.

$$r' = \begin{cases} \frac{bk_x}{a_{cl}}x - \frac{b}{a_{cl}}, & r > x, x \leq \frac{b}{a} - \varepsilon \\ \frac{bk_x}{a_{cl}}x + \frac{b}{a_{cl}}, & r < x, x \geq -\frac{b}{a} + \varepsilon \\ r, & r = x, -\frac{b}{a} + \varepsilon \leq x \leq \frac{b}{a} - \varepsilon \\ \frac{b}{a} - \varepsilon, & r > x, x > \frac{b}{a} - \varepsilon \\ -\frac{b}{a} + \varepsilon, & r < x, x < -\frac{b}{a} + \varepsilon \end{cases} \quad (13)$$

Now,  $x \in X_I$  is enforced, the system will not become “stuck” at  $x = \pm \frac{b}{a}$ , and BIBO stability is achieved.

Notice that while it is necessary to enforce  $x \in X_I$  for the above control laws when the open-loop plant is unstable, neither (9) or (13) require that  $r'$  be statically admissible, viz.,  $|r'| > \frac{b}{a}$  is allowed when  $x \in \text{int}(X_I)$ . This is in contrast to Gilbert’s DTRG, and the globally BIBO stable LQT controller constructed in [9] and [17], which restrict the feasible reference signal to statically admissible values. By relaxing the requirement that the modified reference signal be statically admissible, tracking performance is enhanced.

In [17] simulation results are presented that show control concept 1 provides only a slight improvement in tracking performance over that obtained with the DTRG. This is because control concept 1 does not focus on driving the system as hard as possible. Rather, with control concept 1 the focus is on generating a modified reference signal that is as close to the exogenous reference signal as possible. On the other hand, as shown in [16], control concept 2 does drive the system

as hard as possible, and provides substantially improved tracking performance over that obtained with the DTRG. Presentation of additional simulation results comparing the performance of control concept 2 to that of the DTRG is deferred until Chapter 2 so that first the results of [8] and [9] may be summarized.

#### **1.2.4 Summary**

Constraint mitigation methodologies may be broadly classified as either performance enhancement or constraint avoidance methods. Performance enhancement methods attempt to reduce the number and duration of saturations, but do not fully avoid saturation. Although saturation is allowed with these methods, in many cases conditions can be determined, based on the generalized Nyquist criterion developed in [23], such that the closed-loop system is globally BIBO stable. Since constraint violation is not fully prevented, performance improvement methods do not perform well in the case of open-loop unstable plants.

In the case of open-loop unstable plants, many saturation effects mitigation methods focus on saturation avoidance. Statically admissible and positively invariant sets play central roles in these methods. A common technique is to use a dual-loop controller architecture with an outer loop reference signal governor. The reference signal governor is a nonlinear element that modifies the exogenous reference signal so that the controlled process state is restricted to an admissible positively invariant set. For a simple scalar tracking control system, it is relatively easy to restrict the controlled process state to an admissible positively invariant set without restricting the modified reference signal to statically admissible values. However, higher dimension problems present a much more difficult task. To date, to obtain a guaranteed BIBO stable closed-loop system using these methods, the modified reference signal must be restricted to statically admissible values. This sacrifices achievable tracking performance.

### 1.3 Problem Statement

The problem of tracking control in the face of high amplitude dynamic reference signals, hard state and control constraints, and open-loop unstable plants is investigated. Hard state and control constraints normally arise from plants with actuator dynamics and actuator displacement and rate constraints. This is certainly true in the case of manual flight control. Hard constraints may also arise from other sources as well. For example, other system components may have physical limitations which should not be exceeded.

The objectives here are to maintain small signal performance, emphasize tracking performance, and to obtain a BIBO stable closed-loop system regardless of the exogenous reference signal. At the same time, the modified reference signal should not be restricted to statically admissible values, and the largest possible set of stable initial states should be allowed.

### 1.4 Approach

The stated problem is addressed using a nonlinear dual-loop controller architecture, Fig. ???. The controlled process of Fig. ??? consists of the bare plant and a predetermined linear control law that provides good small signal performance in the absence of saturation. The controlled process is augmented with an outer supervisory loop which contains the nonlinear reference signal governor,  $N$ . The purpose of the reference signal governor is to modify the exogenous reference signal,  $r$ , so that the controlled process state and control constraints are not violated.

For the stated problem, the state space may be divided into two distinct regions. One region,  $X_I \subset \mathfrak{R}^n$ , is the set of initial states that is positively invariant with respect to the controlled process. This set consists of all initial states for which there exists a reference signal, possibly time varying, such that the resultant state trajectory remains in  $X_I$  for all time. Here it is assumed that  $n$  is the dimension of the controlled process state vector. The second region,  $\mathfrak{R}^n - X_I$ , consists of all initial

states for which an unbounded system response ensues regardless of the reference signal. Clearly, to obtain a BIBO stable closed-loop system the controlled process state must be restricted to  $X_I$ .

Unfortunately,  $X_I$  is not necessarily convex, and is not easily characterized for higher order systems. Thus, characterizations of positively invariant subsets of  $X_I$  are sought. One such subset is the maximal statically admissible set,  $X_s \subset X_I$ , which is the set of all initial states for which there exists a constant reference signal such that the ensuing state trajectory remains in  $X_s$  for all time. In [9] a dual-loop controller architecture is used to restrict the controlled process state to an arbitrarily close approximation of  $X_s$ . However, this methodology also restricts the modified reference signal to statically admissible values.

Here the objective is to restrict the controlled process state to  $X_s$  without unnecessarily restricting the modified reference signal to statically admissible values. Then, aggressive control strategies may be implemented which have the potential for improved tracking performance. Thus, a dual-loop nonlinear control law is developed that allows statically inadmissible reference signals, when  $x \in \text{int}(X_s)$ . The dual-loop nonlinear control law provides improved tracking performance over that achievable with linear control laws, and a BIBO stable closed-loop system. Moreover, on-line implementation is practical.

## 1.5 Scope and Assumptions

The methodology proposed here is a model-based approach, and is concerned with discrete-time systems with point-wise in-time state and control constraints. Thus, appropriate equivalent discrete time (EDT) systems are developed for continuous time systems. State and control constraints are represented by a single set inclusion, viz.,

$$y_c(t) = C_c x(t) + D_c u(t) \in Y \subset \mathbb{R}^P$$



where  $y_c(t) \in \mathfrak{R}^P$  is the constrained quantity, and  $Y$ , a polytope that contains the origin, is the constraint set. Open-loop unstable plants are assumed, but the controlled process, consisting of the bare plant and inner loop linear control law, is assumed to be asymptotically stable in the absence of saturation. Let the controlled process and its constraint be represented by

$$x(t+1) = A_{cl}x(t) + B_{cl}r'(t)$$

and

$$y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \in Y \subset \mathfrak{R}^P$$

then, the pair  $(A_{cl}, C_{ccl})$  is assumed to be observable.

The above assumptions are nonrestrictive and are generally satisfied by tracking control systems that are subject to actuator displacement and rate constraints.

## 1.6 Organization

The formal problem formulation and several pertinent definitions are presented in Chapter 2. Also, the results of [8] and [9] are summarized, and a discrete time reference governor is developed for the scalar system of Section 1.2.3. In Chapter 3 an arbitrarily close approximation to the set,  $X_s$ , is characterized with a finite set of linear inequalities. A recursive convex hull algorithm is developed which generates the desired linear inequalities. The method is then demonstrated with a second-order example. Chapter 4 is devoted to developing explicit reference signal governor algorithms which are also demonstrated with several second-order examples. The proposed methodology is applied to a fourth-order flight control problem in Chapter 5. The nonlinear dual-loop controller architecture is applied to the constrained regulator problem in Chapter 6, and conclusions and recommendations are presented in Chapter 7.

## 1.7 Summary

The literature contains numerous constraint effects mitigation methodologies. Many of these methods may be classified as anti-windup or performance improvement methods. In these cases saturation is not completely avoided and the objective is to improve system performance by decreasing the number and duration of periods of saturation. Since anti-windup and performance improvement methods do not prevent constraint violation completely, they do not perform well in the case of open-loop unstable plants.

In the case of open-loop unstable plants constraint avoidance methods must be considered. Lyapunov functions may be used to obtain linear control laws that provide local stability for the constrained regulator control problem. However, a trade-off exists between the size of the set of stable initial states, and closed-loop system performance. Also, this methodology is not applicable to the constrained tracking control problem. On the other hand, Bang-Bang controllers obtained from solution of optimal control problem formulations are generally not practical to implement.

Thus, several recent research efforts have attacked the constrained tracking control problem using a dual-loop controller architecture. In these investigations a linear inner loop control law is implemented which provides good small signal tracking performance and stability in the absence of constraint violation. Then the controlled process is augmented with a nonlinear outer loop which contains a referenced signal governor. The reference signal governor prevents constraint violation by generating a modified reference signal which depends on the exogenous reference signal and the controlled process state.

A common characteristic of current reference governor methods is that to obtain a BIBO stable closed-loop system, the modified reference signal is restricted to statically admissible values. However, This is not strictly necessary. As shown in [17], it is sufficient to restrict the controlled process state vector to a positively invariant set to obtain a BIBO stable closed-loop system.

## Chapter 2 - Static Admissibility and Invariance Concepts

### 2.1 Overview

The concepts of static admissibility and positive invariance are developed in this chapter. A precise statement of the problem and several important definitions are provided to aid the discussion. The results of [8] and [9] are reviewed and a DTRG is developed for the scalar example of Section 1.2.3. The performance of the DTRG is compared to that of the invariance based saturation avoidance method, eq. (13), developed in Section 1.2.3.

### 2.2 Problem formulation and Definitions

Consider the tracking control problem and prespecified linear control law

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t), \quad x(t) \in \mathfrak{R}^n, \quad u(t) \in \mathfrak{R}^1 \\u(t) &= k_x x(t) + k_r r'(t), \quad r(t) \in \mathfrak{R}^1 \\y(t) &= Cx(t) + Dr'(t), \quad y(t) \in \mathfrak{R}^1\end{aligned}\tag{14}$$

where the open-loop dynamics matrix,  $A$ , may have one or more eigenvalues outside the unit circle. Assume there exist hard point-wise in-time constraints on both the state and control variables, including linear combinations of them. It is convenient to express these constraints in terms of a constrained output,  $y_c$ . This is accomplished by making the appropriate choices of matrices  $C_c$  and  $D_c$ , and constraint set  $Y \subset \mathfrak{R}^p$ , viz.,

$$y_c(t) = C_c x(t) + D_c u(t) \in Y\tag{15}$$

The ensuing closed-loop system and output constraint are given by

$$\begin{aligned}x(t+1) &= A_{cl}x(t) + B_{cl}r'(t) \\y(t) &= C_{cl}x(t) + D_{cl}r'(t)\end{aligned}$$

$$y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \in Y \subset \mathbb{R}^p \quad (16)$$

where,  $A_{cl} = A + Bk_x$ ,  $B_{cl} = Bk_r$ ,  $C_{cl} = C + Dk_x$ ,  $D_{cl} = Dk_r$ ,  $C_{ccl} = C_c + D_c k_x$ , and  $D_{ccl} = D_c k_r$ .

The constrained tracking control system of eq. (16) is representative of a large class of practical problems. In particular, tracking control systems subject to actuator displacement and rate constraints are accommodated. Also, while eq. (14) includes a state feedback control law, the controlled process of eq. (16) may result from any prespecified LTI control law that satisfies certain nonrestrictive assumptions; viz.,

1.  $A_{cl}$  is asymptotically stable.
2. The pair  $(A_{cl}, C_{ccl})$  is observable.
3.  $0 \in \text{int}(Y)$ .
4.  $Y$  is polyhedral; viz.,

$$Y = \{y_c \in \mathbb{R}^p : f_i(y_c) \leq 0, \quad i = 1, 2, \dots, s\} \quad (17)$$

where the  $s$  functionals,  $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ , are linear or affine in  $y_c$ , with  $f_i(0) \leq 0$ .

The concepts of positive invariance and static admissibility play important roles in the proposed nonlinear controller synthesis methodology. With this in mind, the following definitions are provided to facilitate our discussion here.

**Definition 1. Admissible State:** *At time increment,  $t$ , the state vector,  $x(t)$ , is admissible with respect to the controlled process of eq. (16) if there exists a reference signal,  $r'(t)$ , such that constraint violation does not occur at time  $t$ , that is, such that  $y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \in Y$ .*

Obviously, the admissibility of  $x(t)$  is dependant on the existence of a special reference signal, viz., a feasible reference signal.

**Definition 2. Feasible Reference Signal:** *Given the admissible state vector,  $x(t)$ , at time increment,  $t$ , the reference signal,  $r'(t)$ , is feasible if it does not result in constraint violation at time  $t$ . That is,  $r'(t)$  is feasible if  $y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \in Y$*

For a given admissible system state,  $x(t)$ ,  $r'(t)$  may only be feasible at time,  $t$ , but not at time  $(t + 1)$ . That is, given  $x(t)$  and a feasible  $r'(t)$ , if  $r'(t + 1) = r'(t)$  it may transpire that  $y_c(t + 1) = C_{ccl}x(t + 1) + r'(t + 1) \notin Y$ . This brings up the concepts of admissible and statically admissible reference signals.

**Definition 3. Admissible Reference Signal:** *Let  $X \subset \mathbb{R}^n$ , be a set of state vectors that are admissible with respect to the controlled process of eq. (16). Given the state vector  $x(t) \in X$ , the reference signal,  $r'$ , is admissible with respect to  $X$  at time,  $t$ , if it results in  $x(t + 1) \in X$ .*

Notice that the admissibility of  $r'$  is dependent on both the set,  $X$ , and the state,  $x(t) \in X$ . In particular, although  $r'$  may be admissible with respect to  $X$  for a given state,  $x_1(t) \in X$ , it is not necessarily admissible with respect to  $X$  for some other state,  $x_2(t) \in X$ . Moreover, although  $r'$  is admissible with respect to  $X$  at time,  $t$ , for a given  $x(t) \in X$ ,  $r'$  is not necessarily admissible with respect to  $X$  at time,  $(t + 1)$ , given the ensuing  $x(t + 1) \in X$ . This requires the concept of statically admissible reference signals.

**Definition 4. Statically Admissible Reference Signal:** *Let  $X \subset \mathbb{R}^n$ , be a set of state vectors that are admissible with respect to the controlled process of eq. (16). Given an admissible state vector  $x(t) \in X$ , the constant reference signal,  $r'$ , is statically admissible with respect to  $X$  if it results in  $x(\tau) \in X$  for all  $\tau \geq t$ .*

A particular reference signal may be statically admissible with respect to  $X$  for one element of  $X$  and not for another. Also, note that for a given  $x(t) \in X$  the set of statically admissible reference signals is a subset of the set of admissible reference signals. If a particular reference signal is statically admissible for a given  $x(t) \in X$ , then the ensuing equilibrium point must be an element of  $X$ . Clearly, if the equilibrium point,  $x_{ss}$ , associated with a particular constant reference signal,  $r_o$ , is admissible with respect to the system of eq. (16), then there exists an admissible set of states,  $X$ , and an initial state,  $x(0) \in X$ , for which  $r_o$  is statically admissible. Thus, the set of

reference signals that are statically admissible with respect to the system of eq. (16) are those that result in an admissible equilibrium point.

**Definition 5. Statically Admissible Reference Signal Set:** *The set of statically admissible reference signals,  $R_s$ , for the controlled process defined by eq. (16), is the set of constant reference signals for which the associated equilibrium point is admissible. That is,*

$$R_s = \{r' \in \mathbb{R}^1 : H_o r' \in Y\} \quad (18)$$

where  $H_o \equiv C_{cl} (I - A_{cl})^{-1} B_{cl} + D_{cl}$ .

The next several definitions deal with the concepts of positively invariant and statically admissible sets. These concepts are central to the constraint avoidance methodology developed in Chapter 4.

**Definition 6. Positively Invariant Set:** *Given a reference signal,  $r'$  (possibly non-constant), the set  $X_{r'} \subset \mathbb{R}^n$  is positively invariant with respect to the controlled process of eq. (16) if for each  $x_o \in X_{r'}$ ,  $x(t) \in X_{r'}$  for all  $t \in Z^+ = \{0, 1, 2, \dots\}$ .*

**Definition 7. Maximal Positively Invariant Set:** *The maximal positively invariant set of initial states,  $X_I$ , for the controlled process defined by eq. (16), is the union of all positively invariant sets. That is,  $X_I = \bigcup_{r' \in \mathbb{R}^1} X_{r'}$ .*

In other words, the maximal positively invariant set of initial states for the controlled process defined by eq. (16), is the set of all initial states for which there exists a reference signal (not necessarily constant) such that  $x(t) \in X$  for all  $t \in Z^+$ . Unfortunately,  $X_I$  is generally not convex, and thus is difficult or impossible to characterize. However, convex positively invariant subsets of  $X_I$  do exist. One of these is the maximal statically admissible set.

**Definition 8. Statically Admissible State:** *A state vector,  $x$ , is statically admissible with respect to the system of eq. (16) if there exists a statically admissible reference signal,  $r' \in R_s$ , such that  $y_c(t) \in Y$  for all  $t \in Z^+$ .*

If  $x(0)$  is statically admissible with respect to the controlled process of eq. (16), then there exists a statically admissible (constant) reference signal,  $r'$ , and a positively invariant set,  $X_{r'}$ , such that  $x(t) \in X_{r'}$  for all  $t \in Z^+$ . Not only is  $X_{r'}$  a positively invariant set, but in this case it is also

a statically admissible set since the given reference signal is statically admissible (constant). In the future, a statically admissible set will be denoted by  $X_{r',s}$ .

**Definition 9. Statically Admissible State Set:** *Given a statically admissible reference signal,  $r' \in R_s$ , the set,  $X_{r',s} \subset \mathbb{R}^n$  is statically admissible with respect to the controlled process of eq. (16) if for each  $x(0) \in X_{r',s}$ ,  $y_c(t) \in Y$  for all  $t \in Z^+$ .*

For a given  $x(0) \in X$  and  $r' \in R_s$ , it is not necessary that  $x(t) \in X$  for all  $t \in Z^+$ , but only that  $y_c(t) \in Y$  for all  $t \in Z^+$ . However, if  $X$  is the set of all statically admissible states, then for each combination of  $x(0) \in X$  and  $r' \in R_s$  that results in  $y_c(t) \in Y$  for all  $t \in Z^+$ , it must transpire that  $x(t) \in X$  for all  $t \in Z^+$ .

**Definition 10. Maximal Statically Admissible State Set:** *The maximal statically admissible set of states,  $X_s$ , for the controlled process defined by eq. (16), is the union of all statically admissible state sets. That is,  $X_s = \bigcup_{r' \in R_s} X_{r',s}$ .*

Clearly, The maximal statically admissible set of states for the controlled process defined by eq. (16) is the set of all initial states for which there exists a statically admissible constant reference signal such that the ensuing trajectory does not violate the system's state and control constraints for all time. That is,

$$X_s = \{x \in \mathbb{R}^n : \exists r' \in R_s \ni y_c(t) \in Y \forall t \in Z^+\}$$

It is important to note that  $X_s$  is positively invariant since, if  $x(0) \in X_s$ , there exists a constant statically admissible reference input,  $r' \in R_s$ , such that  $x(t) \in X_s$  for all  $t \in Z^+$ . It is also important to note that  $X_s \subset X_I$ . With the above assumptions on the closed-loop system and constraint set,  $X_s$  is convex; viz., a polytope. However,  $X_I$  is generally not convex. This will be demonstrated with a two dimensional problem in Chapter 3.

### 2.3 Maximal Output Admissible Sets

In [8] and [9] the concepts of static admissibility and positively invariant sets are employed to develop reference governors for discrete-time systems. Since the results of [8] and [9] play an important role in the proposed methodology, a summary of these results is presented here.

Initially, the regulation problem is exclusively analyzed, and the construction of the maximal statically admissible set is presented. The LTI discrete-time system

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, \quad t = 0, 1, 2, \dots (= Z^+) \quad (19)$$

is considered, and the prespecified, "small signal" linear state feedback control law is

$$u(t) = k_x x(t).$$

Also, assume there are constraints on both the state and control vectors, including linear combinations of them. Thus, actuator rate constraints are accommodated. Now, with appropriate choices of matrices  $C_c$  and  $D_c$ , and constraint set  $Y$ , these rather general constraints are represented as

$$y_c(t) = C_c x(t) + D_c u(t) \in Y \subset \mathbb{R}^p \quad (20)$$

Let  $A_{cl} = A + Bk_x$  and  $C_{c_{cl}} = C_c + D_c k_x$ . Then, (19) and (20) become

$$\begin{aligned} x(t+1) &= A_{cl} x(t), \quad x(0) = x, \quad t \in Z^+ \\ y_c(t) &= C_{c_{cl}} x(t) \in Y \subset \mathbb{R}^p \end{aligned} \quad (21)$$

Hence, the saturation avoidance problem has been transformed into a feasibility problem concerning an unforced LTI discrete-time system with an output constraint. In [8] the maximal output admissible set associated with system (21) is defined as the set of all initial states  $x \in \mathbb{R}^n$  such that the unforced closed-loop linear system's response does not violate the system output constraints for all  $t \in Z^+$ . Thus, the maximal output admissible set is the largest set of initial states,  $x \in \mathbb{R}^n$ , that is positively invariant with respect to system (21). Evidently, the maximal output admissible set,

$$O_\infty(A_{cl}, C_{c_{cl}}, Y) = \{x \in \mathbb{R}^n : C_{c_{cl}} A_{cl}^t x \in Y \quad \forall t \in Z^+\} \quad (22)$$



**Example 1.** The output constraint set  $Y = \{0\}$ . Then the maximal output admissible set is the subspace of unobservable states that corresponds to the pair  $(A_{cl}, C_{c_{cl}})$ , viz.,

$$O_{\infty} = (\mathbf{N}(C_{c_{cl}}))_{A_{cl}};$$

this is the largest subspace contained in the null space of the matrix  $C_{c_{cl}}$  which is invariant under  $A_{cl}$ .

**Example 2.** The set  $Y$  is a cone, as in the situation where we have one-sided control constraints, e.g.,  $0 \leq u(t)$  - in which case  $C_c = 0$ ,  $D_c = I$ ,  $C_{c_{cl}} = k_x$  and the cone  $Y = \mathfrak{R}_+^m$  ( $\equiv$  nonnegative orthant in  $\mathfrak{R}^m$ ). Then

$$O_{\infty}(A_{cl}, C_{c_{cl}}, Y) = \{0\}$$

iff

1. The pair  $(A_{cl}, S)$  is observable, where the  $n$ -column matrix  $S$  is such that  $Sx = 0$  implies that  $C_{c_{cl}}x \in Y$ , and
2.  $A_{cl}$  does not have an eigenvector  $v$  which corresponds to a nonnegative eigenvalue of  $A_{cl}$  such that  $C_{c_{cl}}v \in Y$ ;

for a proof see e.g., [20]. □

We are interested in the case where the constraint set  $Y$  is a polyhedron, viz.,

$$Y = \{y \in \mathfrak{R}^p : f_i(y) \leq 0, \quad i = 1, 2, \dots, s\} \quad (23)$$

i.e., the  $s$  functions  $f_i(y)$ ,  $f_i : \mathfrak{R}^p \rightarrow \mathfrak{R}$  are linear or affine in  $y$ , with  $f_i(0) \leq 0$ . Obviously

$$O_{\infty}(A_{cl}, C_{c_{cl}}, Y) = \{x \in \mathfrak{R}^n : f_i(C_{c_{cl}}A_{cl}^t x) \leq 0, \quad i = 1, 2, \dots, s, \text{ and } t \in Z^+\} \quad (24)$$

Although  $Y$  is a polyhedron, eqs. (22) and (24) each represent an infinite number of constraints.

However, if there exists a finite  $t_i^* \in Z^+$  for each inequality in (24) for which the constraints associated with the  $i^{\text{th}}$  inequality constraint are inactive for  $t > t_i^*$ , then  $O_{\infty}(A_{cl}, C_{c_{cl}}, Y)$  is characterized by a finite number ( $\leq st^*$ ) of inequality constraints, where  $t^* = \max_{1 \leq i \leq s} t_i^*$ . In this case,

$O_\infty(A_{cl}, C_{ccl}, Y)$  is said to be finitely determined - and determination of whether a particular initial state vector,  $x$ , is an element of  $O_\infty(A_{cl}, C_{ccl}, Y)$  involves evaluation of a finite set of linear inequalities. Moreover, it may transpire that the constraints associated with one or more of the inequalities,  $f_i$ , that define  $Y$  are inactive for all  $t \in Z^+$ . Thus, let  $S^*$  denote the set of inequality constraints that are active for some  $t \in Z^+$ , viz.,  $S^* \subset \{1, 2, \dots, s\}$ . Then,  $O_\infty(A_{cl}, C_{ccl}, Y)$  may be written as

$$O_\infty(A_{cl}, C_{ccl}, Y) = \{x \in \mathfrak{R}^n : f_i(C_{ccl}A_{cl}^t x) \leq 0, \quad t = 0, 1, \dots, t_i^*, \text{ and } i \in S^*\} \quad (25)$$

In the discrete time case under consideration, sufficient conditions for the existence of a finite  $t^*$ , viz., sufficient conditions for the finite determination of  $O_\infty$ , are (i)  $A_{cl}$  is asymptotically stable, (ii)  $0 \in \text{int}(Y)$ , (iii)  $Y$  is bounded, and (iv) the pair  $(A_{cl}, C_{ccl})$  is observable. If conditions (i - iv) are satisfied, and  $Y$  is a polytope, then  $t^*$  may be obtained using the following algorithm from [8].

### Gilbet's Algorithm

1. Set  $t = 0$ .
2. Solve the following Linear Programs for  $i = 1, \dots, s$ :

$$\max_{x \in \mathfrak{R}^n} (J_i(x)) = \max_{x \in \mathfrak{R}^n} f_i(C_{ccl}A_{cl}^{t+1}x)$$

*subject to the constraints  $f_j(C_{ccl}A_{cl}^k x) \leq 0, j = 1, \dots, s$  and  $k = 0, \dots, t$ .*

*Let  $J_i^*$  be the maximum value of  $J_i(x)$ . If  $J_i^* \leq 0$  for  $i = 1, \dots, s$ , then stop and set  $t^* = t$ .*

*Otherwise, continue with step 3.*

3. Replace  $t$  by  $t + 1$  and return to step 2.
4. After  $t^*$  has been determined in steps 1 - 3, then  $S^*$  and each  $t_i^*$  may be obtained. Let

$$J_{it}^* = \max f_i(C_{ccl}A_{cl}^t x)$$

*such that  $f_j(C_{ccl}A_{cl}^k x) \leq 0, j = 1, \dots, s$  and  $k = 0, \dots, t^*$ .*

*Then, for each  $i \in \{1, \dots, s\}$  determine  $t_i^* \in Z^+$  so that  $J_{it}^* \leq 0$  for  $t \geq t_i^*$  and  $J_{it}^* > 0$  for*

$t < t_i^*$ . If  $J_{it}^* \leq 0$  for all  $t = 0, \dots, t^*$ , then  $i \notin S^*$ .

The above algorithm is easily implemented using any of the many existing Linear Programming packages.

### 2.3.1 Discrete Time Reference Governor

The concept of maximal output admissible sets is only applicable to unforced systems. Thus, it is not directly applicable to the tracking problem without some modification. In [9] the concept of maximal output admissible sets is adapted for use with the tracking control problem by using a nonlinear element in the DTRG which has first-order dynamics, viz., a first-order lag filter with a variable bandwidth parameter  $\lambda \in [0, 1]$  is used to prefilter the exogenous reference signal,  $r$ . The controlled process state vector is then augmented with the prefilter state, viz., the modified reference signal,  $r'$ , and a set of constraints is developed which characterizes the maximal statically admissible set for the augmented system. The DTRG then scales the reference signal's increments so that the augmented system's constraints are not violated. By choosing a modified reference signal such that the discrete-time system's state update satisfies these constraints, both saturation avoidance and BIBO stability are enforced; the latter also requires that the maximal statically admissible set be bounded. The end result is an augmented system whose exogenous input can be turned off by setting  $\lambda = 0$ .

Thus, consider the tracking control problem

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^m \\ u(t) &= k_x x(t) + k_r r(t), \end{aligned} \tag{26}$$

with state and control constraints

$$y_c(t) = C_c x(t) + D_c u(t) \in Y \subset \mathbb{R}^p \tag{27}$$

and where the control constraint set  $Y$  is given by (23). The ensuing closed-loop system is

$$\begin{aligned} x(t+1) &= A_{cl}x(t) + B_{cl}r(t) \\ y_c(t) &= C_{c_{cl}}x(t) + D_{c_{cl}}r(t) \in Y \subset \mathfrak{R}^p \end{aligned} \quad (28)$$

where, as before,  $A_{cl} = A + Bk_x$ ,  $B_{cl} = Bk_r$ ,  $C_{c_{cl}} = C_{c_{cl}} + D_c k_x$ , and  $D_{c_{cl}} = D_c k_r$ . To transform this problem into one which allows use of the concept of maximal output admissible sets, the exogenous reference signal,  $r$ , is prefiltered by the first-order lag filter given by

$$r'(t+1) = r'(t) + \lambda(r(t), x_g(t)) (r(t) - r'(t)) \quad (29)$$

where  $\lambda(r(t), x_g(t)) \in [0, 1]$ ; and in eq. (28),  $r(t)$  is replaced by the filter's output,  $r'(t)$  (the modified reference signal). Then, the augmented state vector is

$$x_g = \begin{bmatrix} r' \\ x \end{bmatrix} \in \mathfrak{R}^{n+1} \quad (30)$$

and the augmented system dynamics and the output constraints are given by

$$\begin{aligned} x_g(t+1) &= A_g x_g(t) + B_g \lambda(r(t), x_g(t)) (r(t) - [I \ 0] x_g(t)) \\ y_c(t) &= C_g x_g(t) \in Y \subset \mathfrak{R}^p \end{aligned} \quad (31)$$

where,

$$A_g = \begin{bmatrix} I & 0 \\ B_{cl} & A_{cl} \end{bmatrix}, \quad B_g = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad C_g = [D_{c_{cl}} \ C_{c_{cl}}].$$

From eq. (29) notice that if  $\lambda(r(t), x_g(t)) = 1$  the exogenous reference signal,  $r(t)$ , is passed through unmodified, but with a one time-step delay. More importantly, if  $\lambda(r(t), x_g(t)) = 0$  the current modified reference signal,  $r'(t)$ , remains unchanged, and (31) becomes an unforced system with output constraints. Thus, if at time  $t = \tau$  and for all "initial states"  $x_g(\tau) \in O_\infty(A_g, C_g, Y)$  we can choose  $\lambda(r(\tau), x_g(\tau))$  such that the updated state  $x_g(\tau+1) \in O_\infty(A_g, C_g, Y)$ , it is possible to guarantee that the controlled system's state and control constraints will not be violated in the future. Moreover, assume that  $O_\infty(A_g, C_g, Y)$  is bounded. Then because  $\lambda = 0$  is always an option, and because the new reference signal is chosen such that it does not remove the state of the controlled

process from the compact set of statically admissible states, the BIBO stability of tracking control systems which employ the DTRG is guaranteed.

Now, the maximal output admissible set for the augmented system is concerned with the homogeneous system ( $\lambda(k) = 0$ ), and is defined as

$$O_\infty(A_g, C_g, Y) = \{x_g \in \mathbb{R}^2 : f_i(C_g A_g^t x_g) \leq 0, t = 0, \dots, t_i^*, i \in S^*\} \quad (32)$$

where the constraints,  $f_i(C_g A_g^t x_g) \leq 0$ , are inactive for  $t > t_i^*$  and  $i \notin S^*$ . Unfortunately,  $A_g$  is only Lyapunov stable: application of the concept of maximal output admissible sets to the tracking control problem, results in an augmented dynamics matrix which is only Lyapunov stable and thus,  $O_\infty(A_g, C_g, Y)$  is generally not finitely determined. The reason  $O_\infty(A_g, C_g, Y)$  is not finitely determined stems from the fact that the unforced response of a Lyapunov stable linear system does not decay to the origin, unlike that of an asymptotically stable linear system. In fact, the unforced response of the Lyapunov stable linear system does converge to an a priori unknown equilibrium point. In general, the equilibrium point will not be reached in a finite number of time intervals, and thus,  $t^*$  cannot be bounded. However, if we somehow know that the unforced system given by (31) will converge to an equilibrium point  $x_g \in \text{int}(O_\infty(A_g, C_g, Y))$ , then we need only consider the transient response for a finite number of time-steps, until the peak-to-peak magnitude of constrained quantity oscillations decay sufficiently. Toward this end, let  $R_s \subset \mathbb{R}^m$  denote the set of reference signals that are statically admissible with respect to the controlled process (28), see Definition 1 in Section 2.2. Now, if  $r'$  is restricted to a set,  $R_s^\varepsilon \subset \text{int}(R_s)$ , then we are assured that  $x_{g,s} \in \text{int}(O_\infty(A_g, C_g, Y))$ . Then, as long as constraint violation does not occur during the transient response,  $x_g(t) \in \text{int}(O_\infty(A_g, C_g, Y))$  for all  $t \in Z^+$ .

Now, given  $\varepsilon > 0$ , define  $Y(\varepsilon) \subset Y$  by

$$Y(\varepsilon) = \{y : f_j(y_c) \leq -\varepsilon, \quad j = 1, \dots, s\}$$

Also,  $\varepsilon$  is chosen so that  $0 < \varepsilon < \min \{-f_j(0) : j = 1, \dots, s\}$  to insure  $Y(\varepsilon)$  contains the origin.

Recall that

$$H_o = C_{ccl} (I - A_{cl})^{-1} B_{cl} + D_{ccl}.$$

Then, to restrict  $r'$  to the set

$$R_s^\varepsilon = \{r' \in \mathfrak{R}^m : H_o r' \in Y(\varepsilon)\} \subset \text{int}(R_s)$$

append the additional constraints

$$f_j \left( \begin{bmatrix} H_o & 0 \end{bmatrix} x_g \right) \leq -\varepsilon, \quad j = 1, \dots, s \quad (33)$$

to those that define  $O_\infty(A_g, C_g, Y)$ . Denoting  $\text{int}(O_\infty(A_g, C_g, Y))$  by  $O_\infty^\varepsilon$  we have

$$\begin{aligned} O_\infty^\varepsilon &= \{x_g \in \mathfrak{R}^{n+m} : f_i(C_g A_g^t x_g) \leq 0, \quad t = 0, \dots, t_i^*, \quad i \in S^*, \\ &\text{and } f_j \left( \begin{bmatrix} H_o & 0 \end{bmatrix} x_g \right) \leq -\varepsilon, \quad j = 1, \dots, s\}. \end{aligned} \quad (34)$$

Now,  $O_\infty^\varepsilon$  is characterized by a finite set of inequality constraints, and the algorithm of Section 2.3 may be used to determine  $t_i^*$  and  $S^*$ .

On-line implementation of the DTRG is practical since the upper limit imposed on  $\lambda(r(t), x_g(t)) \in [0, 1]$  by each inequality, so that  $x_g(t+1) \in O_\infty^\varepsilon$ , is given by a simple algebraic formula. Thus, if  $x_g(0) \in O_\infty^\varepsilon$ , and at each time-step we choose  $\lambda(r(t), x_g(t)) \in [0, 1]$  such that it satisfies the minimum of the upper limits imposed by all inequality constraints in (34), then  $x_g(t) \in O_\infty^\varepsilon$  for all  $t \in Z^+$ .

Note that  $O_\infty^\varepsilon$  is composed of two sets of constraints. The first set,  $f_i(C_g A_g^t x_g) \leq 0, \quad t = 0, \dots, t_i^*, \quad i \in S^*$ , deals with saturations during the transient response, and the second set,  $f_j \left( \begin{bmatrix} H_o & 0 \end{bmatrix} x_g \right) \leq -\varepsilon, \quad j = 1, \dots, s$ , deals with steady state saturations by limiting the modified reference signal to statically admissible values. Also, since  $Y$  is a polytope that contains the origin,  $O_\infty^\varepsilon$  is likewise a polytope that contains the origin. Thus, an equivalent expression for  $O_\infty^\varepsilon$  is

$$O_\infty^\varepsilon = \{x_g \in \mathfrak{R}^{n+1} : \Gamma_g x_g \leq \beta_g\} \quad (35)$$

where  $\Gamma_g$  is an  $\left(s + \sum_{i \in S^*} (t_i^* + 1)\right) \times (n + m)$  matrix, and  $\beta_g$  is an  $\left(s + \sum_{i \in S^*} (t_i^* + 1)\right)$  vector. This representation will be useful in the sequel.

The linear inequalities,  $\Gamma_g x_g \leq \beta_g$ , represent a finite set of half-spaces in  $\mathfrak{R}^{n+1}$  whose intersection is  $O_\infty^\varepsilon \subset \mathfrak{R}^{n+1}$ . Exploiting these inequalities in the constraint avoidance strategy necessarily results in limiting both the controlled system state vector and the modified exogenous input vector,  $r'$ , to statically admissible values. This sacrifices potential tracking performance since there are certainly cases where a statically inadmissible reference input, over a finite time interval, would not result in saturation. To illustrate this, the performance of the DTRG is compared to that of control law (13) for the scalar system of Section 1.2.3.

### 2.3.1.1 Scalar Example

Application of the DTRG constraint mitigation scheme to the scalar system of Section 1.2.3 requires development of an equivalent discrete-time system. With  $a = 2$ ,  $b = 1$ ,  $k_x = -3$ ,  $k_r = 1$ , and a sampling interval of  $T = 0.001$  sec, the equivalent discrete-time controlled process and constraint are

$$x(t+1) = a_d x(t) + b_d r'(t) \quad (36)$$

$$y_c(t) = u(t) = c_{c,el} x(t) + d_{c,el} r'(t) \in Y \subset \mathfrak{R} \quad (37)$$

$$Y = \{y_c \in \mathfrak{R} : f_i(y_c) \leq 0, i = 1, 2\} \quad (38)$$

where  $a_d = 0.999$ ,  $b_d = 9.995 \times 10^{-4}$ ,  $c_{c,el} = k_x = -3$ ,  $d_{c,el} = k_r = 1$ ,  $r'(t)$  is the feasible reference signal,  $f_1(y_c) = y_c - 1$ , and  $f_2(y_c) = -y_c - 1$ .

The reference governor is

$$r'(t+1) = r(t) + \lambda(t) [r(t) - r'(t)]. \quad (39)$$

Combining the closed-loop system and reference governor dynamics results in the augmented two dimensional system

$$\begin{aligned}x_g(t+1) &= A_g x_g(t) + B_g \lambda(t) (r(t) - [1 \ 0] x_g(t)) \\y_c(t) &= C_g x_g(t) \in Y\end{aligned}\quad (40)$$

where the augmented state

$$\begin{aligned}x_g(t) &= \begin{bmatrix} r'(t) \\ x(t) \end{bmatrix}, \quad B_g = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C_g = [1, -3] \\A_g &= \begin{bmatrix} 1 & 0 \\ 9.995 \times 10^{-4} & 0.999 \end{bmatrix}.\end{aligned}$$

Now, the maximal output admissible set for the augmented system is concerned with the homogeneous system ( $\lambda(t) = 0$ ), and is defined as

$$O_\infty = \{x_g \in \mathfrak{R}^2 : f_i(C_g A_g^t x_g) \leq 0, t = 0, \dots, t_i^*, i \in S^*\} \quad (41)$$

where the constraints,  $f_i(C_g A_g^t x_g) \leq 0$ , are inactive for  $t > t_i^*$  and  $i \notin S^* \subset \{1, 2\}$ . Since the augmented system is only Lyapunov stable  $t_i^*$  may be unbounded, so a finitely determined approximation,  $O_\infty^\varepsilon$ , to the maximal output admissible set is needed. Let  $Y(\varepsilon) = \{y : f_i(y) \leq -\varepsilon, i = 1, 2\}$  where  $\varepsilon = 0.05$ . Then, the additional constraints are added

$$f_j([H_0 \ 0] x_g) \leq -\varepsilon, j = 1, \dots, s \quad (42)$$

where

$$H_0 = d_c + c_c (1 - a_d)^{-1} b_d = -2. \quad (43)$$

Then,  $O_\infty^\varepsilon$  is given by

$$\begin{aligned}O_\infty^\varepsilon &= \{x_g \in \mathfrak{R}^2 : f_i(C_g A_g^t x_g) \leq 0, t = 0, \dots, t_i^*, i \in S^*, \\&f_j([H_0 \ 0] x_g) \leq -0.05, j = 1, 2\}.\end{aligned}\quad (44)$$

The LP of Section 2.3 is used to determine  $t_i^*$  and  $S^*$ . In this case  $t_1^* = t_2^* = 0$ , and  $S^* = \{1, 2\}$ . This should be expected because the controlled process of eqs. (36) and (37) is a stable first-order (over damped) system. Thus, the inequalities that are concerned with transient saturations,



$(f_i(C_g A_g^t x_g) \leq 0, t = 0, \dots, t_i^*, i \in S^*)$ , need only consider the initial error,  $r'(t) - x(t)$ , at each time step. Now, (44) becomes

$$O_\infty^\varepsilon = \{x_g \in \mathfrak{R}^2 : f_i(C_g x_g) \leq 0, i = 1, 2, \\ f_j([H_0, 0]x_g) \leq -0.05, j = 1, 2\}. \quad (45)$$

In this case implementation of the DTRG is made easier by expressing  $O_\infty^\varepsilon$  in terms of eq. (35); viz.,

$$O_\infty^\varepsilon = \{x_g \in \mathfrak{R}^2 : \Gamma_g x_g \leq \beta_g\} \quad (46)$$

where,

$$\Gamma_g = \begin{bmatrix} 1 & -3 \\ -1 & 3 \\ 2 & 0 \\ -2 & 0 \end{bmatrix} \text{ and } \beta_g = \begin{bmatrix} 1 \\ 1 \\ 0.95 \\ 0.95 \end{bmatrix}.$$

Notice that the first two inequalities are concerned with transient response saturations and insure that at each time step  $r'(t)$  is chosen so that  $-1 \leq y_c(t) = u(t) \leq 1$ . The second two inequalities are concerned with steady state response saturations and restrict  $r'(t)$  to  $R_s^\varepsilon$ ; viz.,

$$-0.475 \leq r' \leq 0.475. \quad (47)$$

Now, assuming  $x_g(0) \in O_\infty^\varepsilon$ , the DTRG restricts  $x_g$  to  $O_\infty^\varepsilon$  for all  $t \in Z^+$  by choosing  $\lambda(t) \in [0, 1]$  in eq. (39) so that  $x_g(t+1) \in O_\infty^\varepsilon$ . Combining eqs. (40) and (46), this translates into

$$\lambda(t)\Gamma_g B_g (r(t) - r'(t)) \leq \beta_g - \Gamma_g A_g x_g(t). \quad (48)$$

Thus, the DTRG is implemented by choosing

$$\lambda(t) = \min \{\alpha_1(t), \alpha_2(t), \alpha_3(t), \alpha_4(t)\} \quad (49)$$

where, for each  $i = 1, 2, 3, 4$ ,  $\alpha_i(t)$  is given by

$$\alpha_i(t) = \begin{cases} 1 & \text{if } \theta_i(t) \leq 0 \\ \frac{\theta_i(t)}{\phi_i(t)} & \text{if } \theta_i(t) > 0 \end{cases} \quad (50)$$

and  $\Theta(t)$  and  $\Phi(t)$  are given by

$$\Theta(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \\ \theta_4(t) \end{bmatrix} = \Gamma_g B_g (r(t) - r'(t)),$$

and

$$\Phi(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \phi_3(t) \\ \phi_4(t) \end{bmatrix} = \beta_g - \Gamma_g A_g x_g(t).$$

The performance of the DTRG is compared to that of control concept 2 (eq. (13) of Section 1.2.3). Recall that eq. (13) restricts  $x$  such that

$$-\frac{b}{a} + \varepsilon \leq x \leq \frac{b}{a} - \varepsilon, \quad (51)$$

but, unlike the DTRG, does not otherwise restrict the modified reference signal. Here, a value of  $\varepsilon = 0.5$  is used in eq. (13). Figures 3, 4, and 5 show system responses to a statically admissible exogenous pulse input for both the DTRG and control concept 2. Control concept 2 clearly provides improved tracking of the pulse input over that obtained with the DTRG. Figs. 4 and 5 show that control concept 2 generates statically inadmissible modified reference signals, and drives the control signal to its limits whenever  $r(t) - x(t) \neq 0$  and eq. (51) is satisfied. Also, Fig. 5 shows that control concept 2 behaves very much like a Bang-Bang controller.

Figures 6, 7, and 8 show system responses to a statically inadmissible pulse input. While the statically inadmissible input can not be tracked by either control law, the response of the system using control law (13) is again much faster than that obtained using the DTRG. Figures 6 and 7 show that while control law (13) restricts the system state so that  $|x| \leq \frac{b}{a} - \varepsilon = 0.45$ , it does allow statically inadmissible modified reference signals. This results in improved tracking performance as well as a BIBO stable closed-loop system.

### 2.3.2 Output Admissible Sets Versus Statically Admissible Sets

In the case of the tracking control problem the maximal output admissible set is concerned with the unforced ( $\lambda = 0$ ) augmented system, and may be defined as the set of all initial augmented states,  $x_g(0) = \begin{bmatrix} r' \\ x(0) \end{bmatrix}$ , such that  $x(t) \in X_s$  and  $y_c(t) \in Y$  for all  $t \in Z^+$ . Here  $x(t)$  and  $y_c(t)$  are the controlled process state and output constraint trajectories resulting from the initial

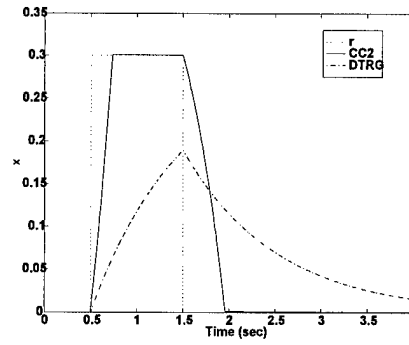


Figure 3. State response to a statically admissible pulse.

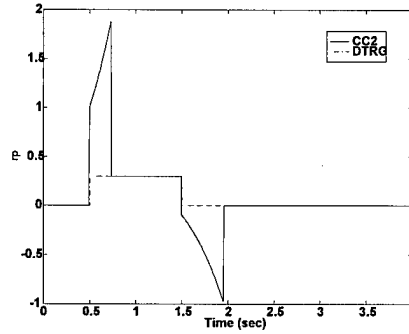


Figure 4. Modified reference signal response to a statically admissible pulse.

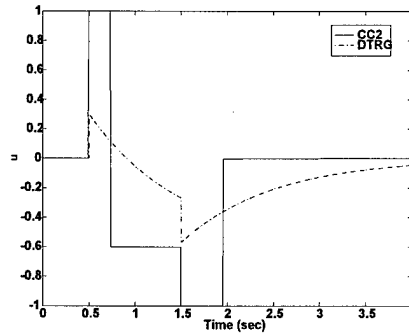


Figure 5. Control signal response to a statically admissible pulse.

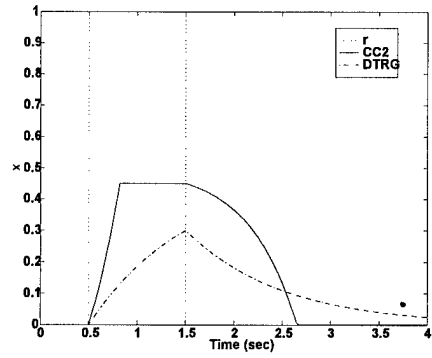


Figure 6. State response to a statically inadmissible pulse.

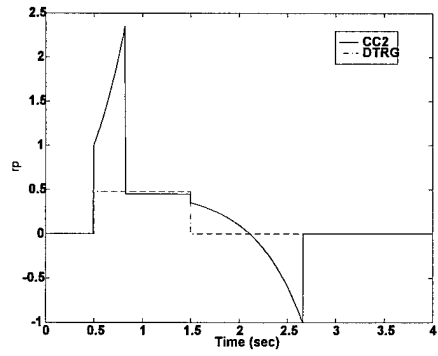


Figure 7. Modified reference signal response to a statically inadmissible pulse.

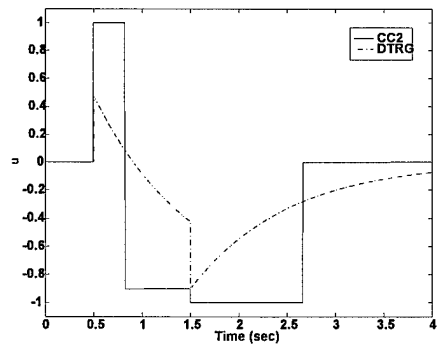


Figure 8. Control signal response to a statically inadmissible pulse.

condition,  $x_g(0) = \begin{bmatrix} r' \\ x(0) \end{bmatrix}$ . From this definition of  $O_\infty(A_g, C_g, Y)$  it is clear that if  $x_g(0) \in O_\infty(A_g, C_g, Y)$ , then  $x(0) \in X_s$  and  $r' \in R_s$ . Also, from the definition of  $X_s$  we have that  $x(0) \in X_s$  if and only if there exists a constant reference signal,  $r'_0 \in R_s$ , such that  $x_g(0) = \begin{bmatrix} r'_0 \\ x(0) \end{bmatrix} \in O_\infty(A_g, C_g, Y)$ . Thus, every  $x_0 \in X_s \subset \mathfrak{R}^n$  may be obtained from the lower partition of a corresponding  $x_{g_0} \in O_\infty(A_g, C_g, Y) \subset \mathfrak{R}^{n+m}$ . Moreover, the set,  $X_s$ , is the projection of  $O_\infty(A_g, C_g, Y)$  onto the subspace  $r' = 0$ .

Similarly, an  $\varepsilon$  approximation to  $X_s$ ,  $X_s^\varepsilon \subset X_s$ , may be obtained by projecting the set  $O_\infty^\varepsilon \subset O_\infty$  onto the  $r' = 0$  subspace. It also turns out that  $X_s^\varepsilon$  is positively invariant with respect to the controlled process. This is proven in Chapter 4.

## 2.4 Summary

A mathematical formulation of the constrained tracking control problem was presented, and precise definitions concerning the concepts of static admissibility and positive invariance were provided. The results of [8] and [9]; viz., the concept of maximal output admissible sets and the development of a discrete time reference governor, were summarized in this chapter. While the DTRG constraint avoidance methodology results in a BIBO stable closed-loop system, it is somewhat conservative in that it restricts the modified reference signal to statically admissible values. This was demonstrated by comparing the performance of the DTRG to that of control law (13) for the scalar system of Section 1.2.3.

Finally, it was noted that the maximal statically admissible set,  $X_s \subset \mathfrak{R}^n$ , is the projection of the maximal output admissible set,  $O_\infty \subset \mathfrak{R}^{n+m}$ , onto the  $r = 0$  subspace. Similarly,  $X_s^\varepsilon \subset X_s$  is the projection of  $O_\infty^\varepsilon \subset O_\infty$  onto the  $r = 0$  subspace. This fact will be exploited in Chapter 3 to obtain a finite set of linear inequalities that characterize  $X_s^\varepsilon \subset X_s$ .

## Chapter 3 - Projection of Polytopes onto a Subspace

### 3.1 Overview

In this Chapter an algorithm is presented which projects the linear inequality constraints, eq. (35), onto the  $\mathfrak{R}^n$  subspace to obtain a finite set of inequality constraints that characterize the set,  $X_s^\varepsilon \subset \mathfrak{R}^n$ . The set,  $X_s^\varepsilon$ , is given by

$$X_s^\varepsilon = \{x \in \mathfrak{R}^n : \Gamma_\varepsilon x \leq \beta_\varepsilon\} \subset \mathfrak{R}^n \quad (52)$$

where  $\Gamma_\varepsilon$  is an  $N \times n$  matrix, and  $\beta_\varepsilon \in \mathfrak{R}^N$ . First, the set of extremal points, or vertices, of  $X_s^\varepsilon$  are determined using a series of Linear Programs. Then the desired linear inequality constraints of eq. (52) are generated with a recursive convex hull algorithm.

The number of linear inequalities contained in eq. (52) can grow quite large as the dimension of the controlled process increases. However, it generally transpires that the boundary of  $X_s^\varepsilon$ ,  $\partial(X_s^\varepsilon)$ , has segments that contain many closely spaced vertices. This suggests that a subset of  $X_s^\varepsilon$  could be characterized with significantly fewer linear inequalities by performing the convex hull algorithm on a reduced set of vertices. The set

$$X_s^\delta = \{x \in \mathfrak{R}^n : \Gamma_\delta x \leq \beta_\delta\} \subset X_s^\varepsilon \quad (53)$$

is obtained by performing the recursive convex hull algorithm using a set  $V' \subset V$ . Given  $V$ ,  $V'$  may be obtained by requiring that all elements be a specified Euclidean distance,  $d$ , from each other. Increasing the distance,  $d$ , reduces the number of vertices included in  $V'$ , the number of inequality constraints required to characterize  $X_s^\delta$ , and the on-line computational burden of the on-line reference signal governor algorithm. The trade off is a slight reduction in the volume of  $X_s^\delta$  over that of  $X_s^\varepsilon$ . This loss of volume may degrade closed-loop system performance. However, a substantial savings in on-line computational burden is generally realized with little or no loss in closed-loop system performance.

### 3.2 Projection of the Vertices

Determination of the vertices of  $X_s^\varepsilon$  is accomplished using a series of Linear Programs. The LPs project those extremal points of  $O_\infty^\varepsilon$  that correspond to extremal points of  $X_s^\varepsilon$  onto the  $\mathfrak{R}^n$  subspace. While the procedure presented here is easily extended to the multi-input case,  $r \in \mathfrak{R}^1$  is assumed in the following.

Express  $O_\infty^\varepsilon$  in terms of eq. (35), and assume  $\Gamma_g = \begin{bmatrix} \gamma_{g_1}^T \\ \vdots \\ \gamma_{g_m}^T \end{bmatrix}$  is an  $M \times (n+1)$  matrix. Let  $\gamma_{g_i}^T = \begin{bmatrix} \gamma_{g_{i_1}} & \vdots & \tilde{\gamma}_{g_i}^T \end{bmatrix}$ ; where,  $\tilde{\gamma}_{g_i}^T = \begin{bmatrix} \gamma_{g_{i_2}} & \gamma_{g_{i_3}} & \cdots & \gamma_{g_{i_{(n+1)}}} \end{bmatrix}$ . Then, extremal points, or vertices, of the polytope,  $X_s^\varepsilon$ , may be obtained by solving the Linear Programs

$$\max_{x_g} c_i^T x_g \quad \text{such that } \Gamma_g x_g \leq \beta_g, \quad i = 1, 2, \dots, M \quad (54)$$

where

$$c_i^T = \begin{bmatrix} 0 & \tilde{\gamma}_{g_i}^T \end{bmatrix}.$$

Cases where  $c_i^T = 0$  are ignored. There are generally  $s$  of these cases which arise from the constraints that limit  $r'$  to statically admissible values. Each  $c_i^T$  produces an extremal vector,  $x_{g_i}^* = \begin{bmatrix} r_i^* \\ x_i^* \end{bmatrix}$ . Moreover,  $x_i^*$  is the projection of  $x_{g_i}^* \in \mathfrak{R}^{n+1}$  onto the  $\mathfrak{R}^n$  subspace. Also, several  $c_i^T$  may produce the same extremal point. Thus, the Linear Program (54) generally generates  $K < M$  unique vertices.

Important properties of the set of unique vertices,  $V = \{x_1^*, \dots, x_K^*\} = \{v_1, \dots, v_K\}$ , obtained from the solution of the Linear Program (54) include the following:

1.  $v_i \in \text{co}(V) \quad \forall i = 1, \dots, K$ ;
2. Each hyperplane that supports  $X_s^\varepsilon$  contains at least  $n$  vertices from  $V$ ;
3. Each  $v_i \in V$  is contained in at least  $n$  of the hyperplanes that support  $X_s^\varepsilon$ .

Property 1 follows from the fact that LP (54) generates extremal points of  $X_s^\varepsilon$ . This simplifies the convex hull algorithm since it is not necessary to determine which vertices are interior points and which are extremal points. Property 2 follows from the fact that hyperplanes are co-dimension 1 objects, and are uniquely defined by  $n$  vertices. This property leads to  $n$  equations to determine the  $n$  elements of each row of  $\Gamma_\varepsilon$ . Finally, Property 3 follows from the fact that  $X_s^\varepsilon$  is bounded; viz., a polytope. Of course, in problems where  $n \geq 3$  a particular vertex may be contained in more than  $n$  supporting hyperplanes.

### 3.3 Recursive Convex Hull Algorithm

A recursive algorithm is developed that constructs  $X_s^\varepsilon$ , or  $X_s^\delta$ , given the set of vertices,  $V$ , or  $V'$ . First, some brief comments concerning notation and definitions are made to aid in the following discussion. For a thorough presentation of relevant definitions and concepts concerning polyhedral sets see Chapter 3 of [24].

The convex hull of a finite set of points in  $E^d$  is a convex polytope. Moreover, a polyhedral set in  $E^d$  is the intersection of a finite set of closed half-spaces. Thus, the linear inequalities of eq. (52) may be obtained by constructing the convex hull of  $V$ . A convex polytope is described by means of its boundary, which consists of *faces*. Each *face* of a convex polytope is a convex set itself. A  $k$ -*face* denotes a  $k$ -dimensional *face*, that is a *face* whose affine hull has dimension  $k$ . If a polytope is  $d$ -dimensional, its  $(d-1)$ -faces are called facets, its  $(d-2)$ -faces are called subfacets, its 1-faces are called edges, and its 0-faces are vertices. Note, that the affine hull of a facet is a hyperplane. A  $d$ -polytope is called a simplex if it is the convex hull of  $(d+1)$  affinely independent points. Moreover, a  $d$ -polytope is called simplicial if each of its facets is a simplex. Finally, the convex hull of a finite set of points in general position is a simplicial polytope.



The following algorithm constructs  $X_s^\varepsilon$  in a simplicial manner. That is,  $X_s^\varepsilon$  is constructed such that each facet is a simplex. An initial  $n$ -simplex is formed from  $(n + 1)$  elements of  $V$ . Then additional vertices are incorporated one at a time. As each new vertex is added, one or more facets are invalidated, and new facets are constructed to form a new simplicial  $n$ -tope. The algorithm terminates after all vertices are incorporated. The elements of  $V$  are not necessarily in general position, and thus  $X_s^\varepsilon$  is not necessarily a simplicial polytope. However, this does not preclude the use of a simplicial construction method. If the elements of  $V$  are not in general position there may be cases where the affine hulls of two or more facets are the same. That is, facets that are not simplexes are partitioned into two or more facets that are simplexes. This, in turn, will result in redundant inequalities in eq. (52). Redundant inequalities are easily eliminated after the algorithm terminates. There may also be cases where a new vertex,  $v_j$ , is contained in an existing facet. In this case no facets are invalidated by the addition of  $v_j$ , and  $v_j$  may be discarded.

Now, let  $f_i$  be a facet of the  $n$ -dimensional simplicial polytope,  $X_s^\varepsilon$ . Then the  $(n-1)$ -dimensional affine hull of  $f_i$  is a hyperplane, and is denoted by  $h_i$ . Since the origin is in the interior of  $X_s^\varepsilon$ ,  $h_i$  is uniquely defined by the triple,  $(\gamma_i^T, v_i, b_i)$ , where  $\gamma_i^T$  is a linear functional, and the  $i^{\text{th}}$  row of  $\Gamma_\varepsilon$ ,  $v_i \in V$  is one of the  $n$  vertices contained in  $h_i$ , and  $b_i$  is an arbitrary, non-zero, scalar. Since  $b_i$  must be chosen such that  $\gamma_i^T(0) < b_i$ , then

$$b_i > 0 \quad \forall i = 1, \dots, N$$

Also, note that

$$\gamma_i^T v \leq b_i \quad \forall v \in V. \quad (55)$$

This is due to the convexity of  $X_s^\varepsilon$ . From the preceding discussion it is apparent that associated with each facet,  $f_i$ , is a hyperplane,  $h_i$ , and a linear inequality, or half-space. Of course it is the set of linear inequalities that we seek. The linear functionals,  $\gamma_i^T$ , are always constructed so that the

normal is outward pointing. This insures that the resulting closed half-space is on the correct side of the hyperplane.

Now, if the  $n$  vertices,  $v_{i_1}, \dots, v_{i_n}$ , contained in  $h_i$  are known, and  $b_i$  is chosen such that  $b_i = 1$ , then  $\gamma_i^T \in \mathfrak{R}^n$  is given by

$$\gamma_i^T = [ 1 \quad \dots \quad 1 ] [ v_{i_1} \quad \dots \quad v_{i_n} ]^{-1} \quad (56)$$

and the desired linear inequality is given by

$$\gamma_i^T x \leq 1. \quad (57)$$

The problem then, is to determine the number of facets that make up  $co(V)$ , and which  $n$  vertices are contained in each facet. The following recursive convex hull algorithm solves this problem.

First, an initial  $(n + 1)$ -simplex is constructed from an initial set of  $(n + 1)$  vertices from  $V$ . The simplex is constructed using eq. (56) and the  $(n + 1)$  unique combinations of  $n$  out of  $(n + 1)$  vertices. Moreover, the initial  $(n + 1)$  vertices are chosen so that the resultant simplex, denoted by  $X_{n+1}^e \subset X_s^e$ , contains the origin. This is accomplished by insuring that each of the initial  $n + 1$  inequalities,

$$\gamma_i^T x \leq 1, \quad i = 1, \dots, n + 1 \quad (58)$$

is satisfied for each of the initial  $(n + 1)$  vertices. If an initial inequality is not satisfied for all  $n + 1$  initial vertices, the origin is not an interior point of  $X_{n+1}^e$ . It is not strictly necessary to insure the origin is an interior point of the initial simplex. However, if the origin is not an interior point, then there will be cases where eq. (57) is changed to

$$-\gamma_i^T x \leq -1$$

to insure the normal is outward pointing. Also, each new linear inequality generated during subsequent passes of the algorithm must be tested using a vertex contained in another existing hyperplane to insure the normal is outward pointing. This additional processing is avoided if the initial simplex

contains the origin. Also, an initial set of  $(n + 1)$  vertices is easily found with an automated search routine. Figure 9 shows an example of the initial tetrahedron formed for a third-order problem.

Now, we have the initial convex subset of  $X_s^\varepsilon$ ,

$$X_{n+1}^\varepsilon = \{x \in \mathfrak{R}^n : \Gamma_{n+1}x \leq \beta_{n+1}\} \subset X_s^\varepsilon$$

where  $\Gamma_{n+1} \in \mathfrak{R}^{(n+1) \times n}$  and  $\beta_{n+1} \in \mathfrak{R}^{n+1}$ .

Given  $X_{n+1}^\varepsilon$ , additional vertices from  $V$  are incorporated one at a time until all vertices are included. This is accomplished by first determining which of the existing inequalities are not satisfied for the new vertex. Since half-spaces defined by failed inequalities do not contain the new vertex, and  $X_s^\varepsilon$  is convex, all failed inequalities, and their associated hyperplanes and facets, are invalid. Thus, these inequalities are eliminated. Next, the vertices contained in failed hyperplanes are used together with the new vertex to generate new hyperplanes and associated linear inequalities. Notice that the focus here is on half-spaces and hyperplanes, and not on faces of the polytope.

Determining which inequalities are not satisfied for a new vertex is accomplished as follows.

Assume  $k$  vertices from  $V$  have been incorporated to obtain

$$\Gamma_k = \begin{bmatrix} \gamma_{k_1}^T \\ \vdots \\ \gamma_{k_p}^T \end{bmatrix} \in \mathfrak{R}^{p \times n}, \quad \beta_k \in \mathfrak{R}^p,$$

and the convex set

$$X_k^\varepsilon = \{x \in \mathfrak{R}^n : \Gamma_k x \leq \beta_k\} \subset X_s^\varepsilon.$$

Let  $v_{k+1}$  denote the new vertex. On the first pass  $k = p = n + 1$ . Also, recall that all elements of

$\beta_k$  are 1. Now, compute

$$\zeta_k = \begin{bmatrix} \gamma_{k_1}^T v_{k+1} \\ \vdots \\ \gamma_{k_p}^T v_{k+1} \end{bmatrix} = \begin{bmatrix} \zeta_{k_1} \\ \vdots \\ \zeta_{k_p} \end{bmatrix},$$

and note that if

$$\zeta_{k_i} - 1 > 0, \tag{59}$$

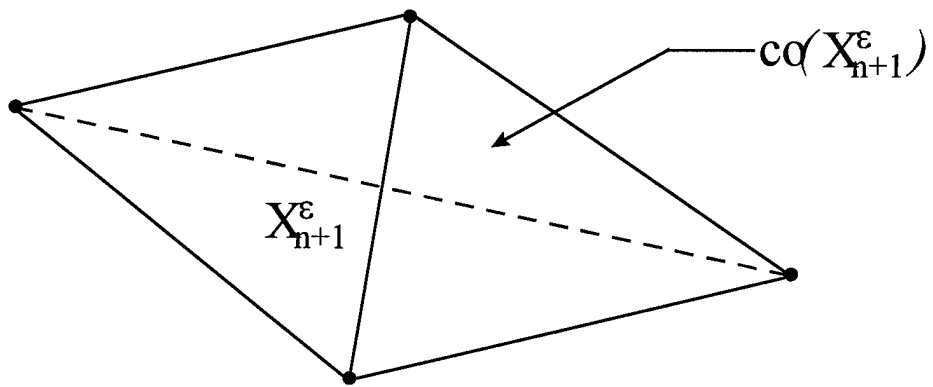


Figure 9. Initial tetrahedron for a problem in  $E^3$ .

then the linear inequality,  $\gamma_{k_i}^T x \leq 1$ , is not satisfied, and is therefore invalid. If  $v_{k+1}$  is contained in an existing facet, then no linear inequalities will be invalidated. In this case,  $v_{k+1}$  may be discarded. If  $v_{k+1}$  is contained in an existing hyperplane, but not the associated facet, then one or more linear inequalities will fail, and the ensuing convex polytope will contain two or more facets that have the same affine hull. That is, two or more hyperplanes and their associated linear inequalities will be identical. The above situations only occur if the elements of  $V$  are not in general position. Also, if the set of vertices is reduced appropriately these events generally will not occur.

Figure 10 shows an example in  $E^2$ . Here, the addition of  $v_4$  results in the elimination of the hyperplane associated with vertices  $v_1$  and  $v_2$ . Two new hyperplanes are generated; one containing vertices  $v_1$  and  $v_4$ , and one containing vertices  $v_2$  and  $v_4$ . Notice that in the two dimensional case the addition of a new vertex results in the elimination of one hyperplane, and the creation of two new hyperplanes. Moreover, it is readily apparent which vertices are contained in each of the new hyperplanes.

This is not the case for problems with  $n \geq 3$ . In general, for  $n \geq 3$ , adding a new vertex may invalidate several existing inequalities. A three dimensional example is shown in Figure 11. In this example, the addition of  $v_8$  invalidates the linear inequalities associated with the three facets  $f_1$ ,  $f_2$ , and  $f_3$ , and their associated hyperplanes. Now, it may not be immediately clear how many new facets / hyperplanes should be created, and which vertices feature together in each new facet / hyperplane.

This issue is resolved by noting the following. Failed facets are contiguous. This is due to the convexity of  $X_k^\varepsilon$ . Also, the intersection of two facets contains  $(n - 1)$  vertices and is a subfacet of  $X_k^\varepsilon$ . Not only does a subfacet contain  $(n - 1)$  vertices, but its affine hull is also contained in exactly two supporting hyperplanes, and has co-dimension two, whereas hyperplanes have co-dimension one. Finally, subfacets contained in failed facets may be categorized as either "failed" subfacets or

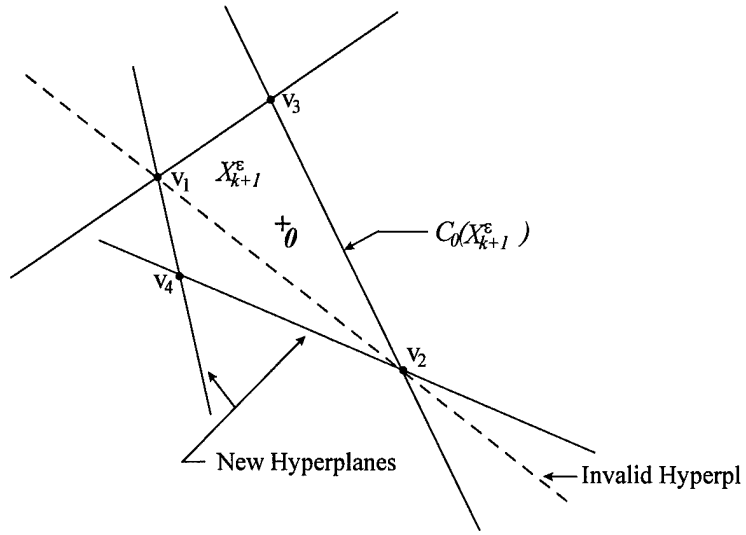


Figure 10. Incorporation of  $v_{k+1}$  for an  $E^2$  arrangement.

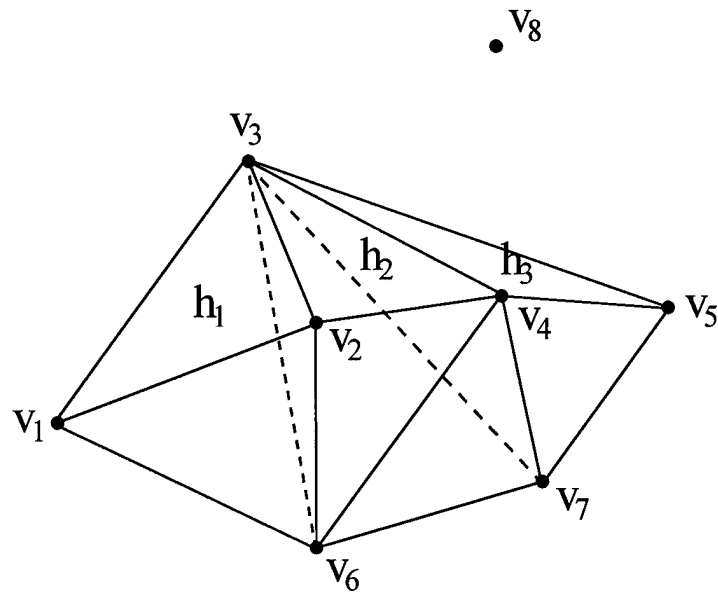


Figure 11. Invalidation of multiple hyperplanes for a problem in  $E^3$ .

“valid” subfacets. Failed subfacets are those subfacets that are contained in the intersection of two failed facets. Valid subfacets are those subfacets that are contained in the intersections of failed and valid facets.

Now it transpires that failed subfacets lie in the shadow of the supporting cone of  $v_{k+1}$ , and do not feature in  $c_o(X_{k+1}^\varepsilon)$ . Valid subfacets, on the other hand, do not lie in the shadow of the supporting cone of  $v_{k+1}$ , and are each contained in exactly one new facet. Also, each new facet contains exactly one valid subfacet. This situation follows from the convexity of  $X_k^\varepsilon$ , the fact that each vertex is contained in  $c_o(X_{k+1}^\varepsilon)$ , and the fact that valid subfacets are contained in existing facets that were not invalidated by  $v_{k+1}$ . Thus, the number of new facets, and hyperplanes, is equal to the number of valid subfacets, and the  $n - 1$  vertices from each valid subfacet are combined with  $v_{k+1}$  to obtain a new hyperplane using eq. (56).

Figure 12 shows the situation after  $v_{k+1} = v_8$  is incorporated in the example of Fig. 11. While three facets / hyperplanes have been eliminated, five new facets / hyperplanes have been created. Moreover, the failed subfacets,  $e_{23}$  and  $e_{34}$ , do not feature in  $C_0(X_{k+1}^\varepsilon)$ , while the valid subfacets,  $e_{12}$ ,  $e_{13}$ ,  $e_{24}$ ,  $e_{35}$ , and  $e_{45}$ , are each contained in exactly one new facet / hyperplane. Also, each new facet / hyperplane contains exactly one valid subfacet.

Since  $V$  does not contain interior points, but only vertices of  $X_s^\varepsilon$ , the convex hull algorithm need not consider this case. However, the above algorithm easily handles sets that include interior points. As the algorithm progresses, if a new point,  $v_{k+1}$ , is interior to the existing polytope,  $X_k^\varepsilon$ , then no linear inequalities will fail, and the point is discarded. An interior point may also be incorporated into  $X_k^\varepsilon$  during a particular pass. Then, during a subsequent pass this point will not be contained in any valid subfacet, and again is discarded.

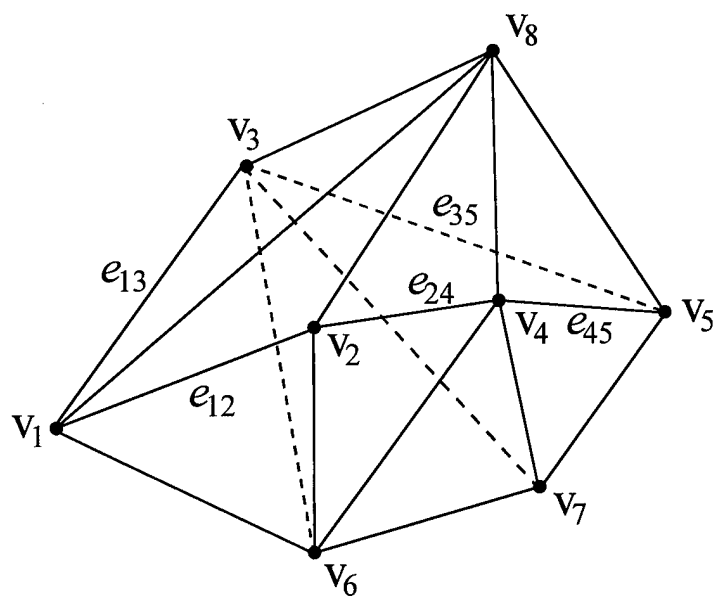


Figure 12. Incorporation of  $v_8$  for the  $E^3$  example.



The recursive convex hull algorithm is now described for the general  $n$ -dimensional problem. Given the set of unique vertices,  $V$ , obtained from the Linear Program (54), perform the following procedure:

1. Generate the linear inequalities that define  $X_{n+1}^\varepsilon$  using eq. (56) and  $n + 1$  initial vertices. Choose the first  $n + 1$  vertices so that  $0 \in \text{int}(X_{n+1}^\varepsilon)$ . Make a list of vertices included in each facet.
2. Set  $k = n + 1$ . That is,  $X_k^\varepsilon = X_{n+1}^\varepsilon$ .
3. Select any remaining  $v \in V$  and set  $v_{k+1} = v$ .
4. Use eq. (59) to determine invalid inequalities and hyperplanes. If no linear inequalities are invalidated, discard  $v_{k+1}$  and select the next element of  $V$ .
5. Delete failed inequalities from the set of inequalities that define  $X_k^\varepsilon$ .
6. Identify failed and valid subfacets from the set of failed facets / hyperplanes. First, generate a list of the  $n - 1$  vertices contained in each of the  $n$  subfacets of each failed facet. This list is generated from the  $n$  combinations of  $n - 1$  out of  $n$  vertices contained in each facet. A subfacet, identified by the  $n - 1$  vertices it contains, fails if it is contained in two failed facets.
7. Generate a new facet / hyperplane for each valid subfacet, using eq. (56), by combining the  $n - 1$  vertices of each valid subfacet with  $v_{k+1}$ . New linear inequalities follow immediately from the new hyperplanes.
8. Combine the new linear inequalities with those remaining from  $X_k^\varepsilon$  to obtain  $X_{k+1}^\varepsilon$ , and update the list of vertices contained in each facet.
9. Repeat steps 3 through 8 until all  $v \in V$  are incorporated.

This algorithm generates  $X_s^\varepsilon$  by constructing a sequence of polytopes,  $X_k^\varepsilon \subset X_s^\varepsilon$ . At each step we have that  $X_k^\varepsilon \subset X_{k+1}^\varepsilon$  since inclusion of the new vertex,  $v_{k+1}$ , cannot render any of the previously incorporated vertices internal, or nonextremal, points of  $X_{k+1}^\varepsilon$ . The set,  $X_s^\varepsilon$ , is built up in a simplicial form, and the volume of  $X_k^\varepsilon$  increases monotonically with  $k$ . Moreover, as noted above, this algorithm is applicable to the general case where the points are not in general position and where there may be those that are interior to the convex hull. If the points are not in general position, facets that are not simplexes are simply partitioned into two or more simplexes. Interior points are eliminated automatically with no additional processing. If  $v_{k+1}$  is interior, then no facets will fail in step 4, and it is discarded. Also, if an interior point,  $p$ , is incorporated into an intermediate polytope during a particular pass, then on a subsequent pass  $p$  will be in the shadow of the supporting cone associated with a new point. In this case  $p$  is only contained in failed subfacets, and is automatically eliminated by virtue of not being included in a new facet.

In practice it is often desirable to reduce the number of linear inequalities that must be evaluated by the on-line reference signal governor. This is accomplished by thinning the set of vertices,  $V$ . A set,  $V' \subset V$ , is obtained by including only those elements of  $V$  that are a prespecified Euclidean distance,  $d$ , from each other. The set,  $X_s^\delta \subset X_s^\varepsilon$ , is then obtained by performing the recursive algorithm on  $V'$ . The number inequalities required to characterize  $X_s^\delta$  is fewer than that required to characterize  $X_s^\varepsilon$ . Of course, the volume of  $X_s^\delta$  is also less than that of  $X_s^\varepsilon$ . However, the number of vertices in  $V'$ , and the number of inequalities required to characterize  $X_s^\delta$ , can generally be reduced substantially with little loss in volume. Finally, it is important to note that the recursive convex hull algorithm is performed off-line, and only the resultant linear inequalities of eq. (52), or eq. (53), are used by the on-line constraint mitigation strategy.

### 3.3.1 Numerical Precision Considerations

Due to finite word length limitations of digital machines, it may be necessary to account for computer round-off errors when determining valid and failed linear inequalities using eq. (59). Computer round-off errors may creep in when generating the initial linear inequalities that characterize  $O_\infty^\varepsilon$ , when obtaining the vertices of  $X_s^\varepsilon$ , and when generating the linear inequalities that characterize  $X_k^\varepsilon$ . This problem can be eliminated by sufficiently reducing the set of vertices.

Another motivation for reducing the set of vertices is to obtain a set of vertices that lie in general position. If the elements of  $V$  are not in general position, then  $X_s^\varepsilon$  is not simplicial. That is, some facets of  $X_s^\varepsilon$  will contain more than  $n$  vertices. Since the above convex hull algorithm constructs  $X_s^\varepsilon$  in a simplicial manner, facets that are not simplexes are partitioned into two or more simplexes that have the same affine hull. As noted earlier, this results in redundant inequalities in eq. (52). Figure 13 shows a three dimensional example where facets  $f_4$ ,  $f_5$ , and  $f_6$  are all contained in the same hyperplane. In this case elimination of  $v_3$  yields a simplicial polytope with no loss of volume. Figure 14 shows another third-order example. In this case facets  $f_4$  and  $f_5$  are contained in the same hyperplane. Unlike the example of Fig. 13, deletion of any of the involved vertices results in some loss of volume.

In theory, as the recursive convex hull algorithm incorporates new vertices, the above situations can be detected and appropriate action taken. However, computer round-off errors must be accounted for when determining if a set of vertices lie in general position. One method, is to declare that  $v_{k+1}$  is contained in the  $i^{th}$  hyperplane,  $h_i$ , whenever

$$|\gamma_{ki}^T v_{k+1} - 1| < \mu$$

where,  $\mu$  is a small positive tolerance. In this case the vertices do not lie in general position. The appropriate value of  $\mu$  is dependant on the numerical precision of the host computer system. Another

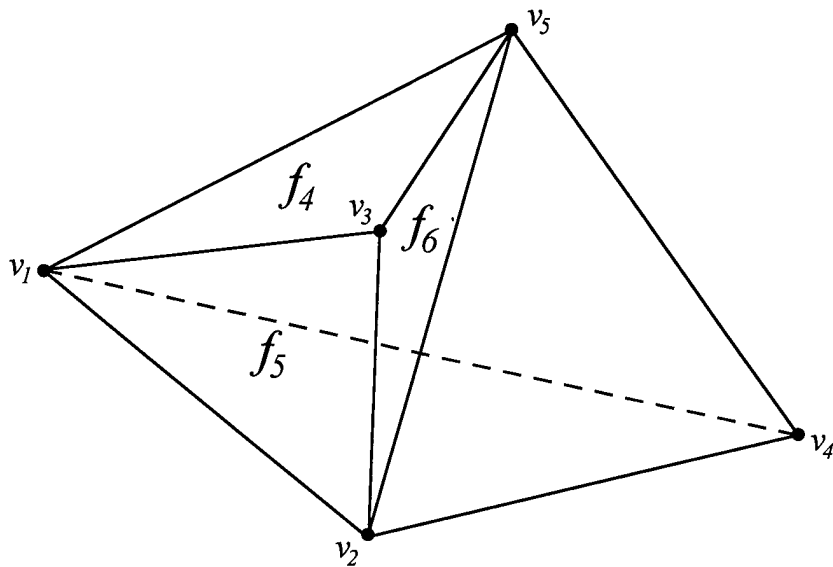


Figure 13. A set of points that are not in general position in  $E^3$

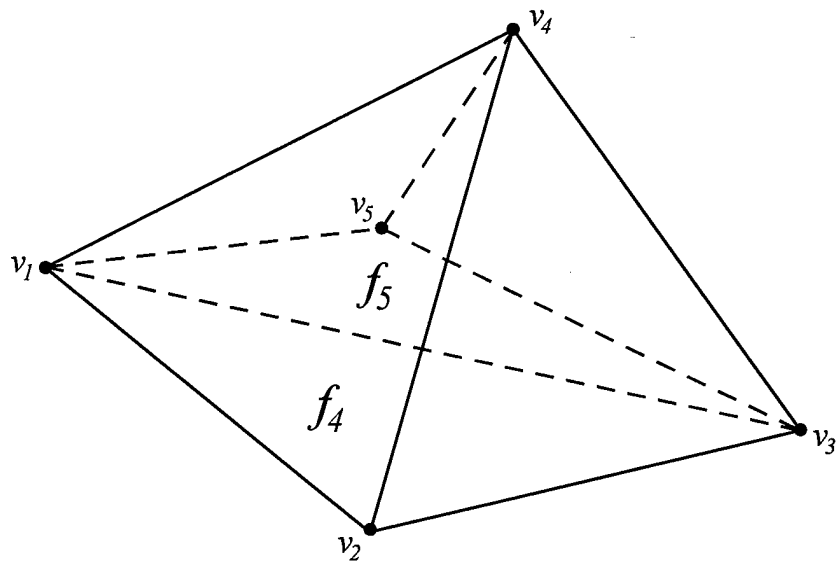


Figure 14. A nonsimplicial polyhedron in  $E^3$

method is to compute the condition numbers of the matrices

$$M_{k_i} = \left[ \frac{v_{i_1} - v_{i_2}}{\|v_{i_1} - v_{i_2}\|_2}, \frac{v_{i_1} - v_{i_3}}{\|v_{i_1} - v_{i_3}\|_2}, \dots, \frac{v_{i_1} - v_{i_n}}{\|v_{i_1} - v_{i_n}\|_2}, \frac{v_{i_1} - v_{k+1}}{\|v_{i_1} - v_{k+1}\|_2} \right], \quad i = 1 \dots p$$

where  $v_{i_j}$ ,  $j = 1 \dots n$ , are the  $n$  vertices contained in the  $i^{\text{th}}$  facet / hyperplane of  $X_k^\varepsilon$ . Notice that if  $v_{k+1}$  is contained in  $h_i$  then the elements of  $M_{k_i}$  define a hyperplane that passes through the origin. In this case one eigenvalue of  $M_{k_i}$ ,  $\lambda_{\min_i} = \min_i |\lambda(M_{k_i})|$ , is zero,  $\text{Rank}(M_{k_i}) = n - 1$ , and  $\text{Cond}(M_{k_i}) = \text{undefined}$ . Now, due to computer round-off errors, it generally transpires that  $\lambda_{\min_i}$  is non-zero and  $\text{Cond}(M_{k_i})$  is a large, but defined, real number. Thus, an appropriate threshold,  $\xi$ , for  $\text{Cond}(M_{k_i})$  must be determined. Then, after the recursive convex hull algorithm has terminated, if

$$\max_i (\text{Cond}(M_{k_i})) > \xi$$

on any pass of the algorithm, the elements of  $V$ , or  $V'$ , are not in general position. Determination of an appropriate hyperplane failure tolerance,  $\mu$ , and matrix condition number threshold,  $\xi$ , requires some analysis.

This analysis is demonstrated with the following example in  $E^3$ . Let the  $i^{\text{th}}$  hyperplane,  $h_i$ , be specified by the linear functional,

$$\gamma_{k_i}^T = [-1, 2, -4], \text{ and scalar, } b_i = 1$$

It is easy to verify that the vertices,

$$v_{i_1} = \begin{bmatrix} 1 \\ 2 \\ 0.5 \end{bmatrix}, \quad v_{i_2} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}, \quad v_{i_3} = \begin{bmatrix} 0.5 \\ -3 \\ -1.875 \end{bmatrix}, \quad \text{and } v_{k+1} = \begin{bmatrix} -2 \\ 5 \\ 2.75 \end{bmatrix}$$

are each contained in  $h_i$ . A perturbed vertex,  $\tilde{v}_{k+1}$ , is generated by perturbing the third element of  $v_{k+1}$  by a small amount, viz.,

$$\tilde{v}_{k+1} = \begin{bmatrix} -2 \\ 5 \\ 2.75 + \delta \end{bmatrix}.$$

Now, compute  $\zeta_{k_i} = \gamma_{k_i}^T \tilde{v}_{k+1} - 1$ ,  $Cond(M_{k_i})$ ,  $Rank(M_{k_i})$ , and  $\lambda_{\min_i}$  for several values of  $\delta$ .

Here,

$$M_{k_i} = \left[ \frac{v_{i_1} - v_{i_2}}{\|v_{i_1} - v_{i_2}\|_2}, \frac{v_{i_1} - v_{i_3}}{\|v_{i_1} - v_{i_3}\|_2}, \dots, \frac{v_{i_1} - v_{i_n}}{\|v_{i_1} - v_{i_n}\|_2}, \frac{v_{i_1} - \tilde{v}_{k+1}}{\|v_{i_1} - \tilde{v}_{k+1}\|_2} \right].$$

Table 1 shows the results for several values of  $\delta$ .

The results of Table 1 were obtained using Matlab<sup>®</sup> version 4.2, hosted on a Pentium 120 MHz personal computer. This system's machine epsilon, the distance from 1 to the next largest floating point number, is  $2.22 \times 10^{-16}$ . From Table 1 it is clear that  $h_i$  should be declared invalid if  $\zeta_{k_i} > 3.2 \times 10^{-14}$ . Also, if  $Cond(M_{k_i}) > 1.4 \times 10^{15}$ , then the elements of  $V$ , or  $V'$ , are not in general position. Thus, in this case, reasonable values for  $\mu$  and  $\xi$  are  $3 \times 10^{-14}$  and  $1 \times 10^{15}$ , respectively. It is important to note that evaluating the condition numbers of the matrices,  $M_{k_i}$ , provides much more information than simply testing the rank of these matrices. Evaluating the above condition numbers lets the designer know the relative degree to which the elements of  $V$ , or  $V'$ , are in general position.

### 3.4 Two Dimensional Problem

The following two dimensional problem provides an illuminating demonstration of the methods described in this chapter. Moreover, the two dimensional problem allows us to make use of graphical interpretations of the concepts presented here. First, the maximal statically admissible set,  $X_s$ , is constructed for the continuous-time system. Then, the discrete-time methods of Sections 3.2 and 3.4 are used to develop a finitely determined approximation to  $X_s$ .

Consider the two dimensional continuous time system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t) \\ y(t) &= Cx(t) = [1 \ 0] x(t) \end{aligned} \tag{60}$$

Table 1. Numerical precision analysis

$\delta$	$\zeta_{k_i} - 1$	$Cond(M_{k_i})$	$Rank(M_{k_i})$	$\lambda_{\min_i}$
-0.05	0.2	219	3	$2.28 \times 10^{-2}$
-0.001	$4.00 \times 10^{-3}$	$1.10 \times 10^4$	3	$4.38 \times 10^{-4}$
$-1 \times 10^{-12}$	$4.00 \times 10^{-12}$	$1.10 \times 10^{13}$	3	$4.37 \times 10^{-13}$
$-1 \times 10^{-13}$	$4.00 \times 10^{-13}$	$1.10 \times 10^{14}$	3	$4.39 \times 10^{-14}$
$-1 \times 10^{-14}$	$4.09 \times 10^{-14}$	$1.04 \times 10^{15}$	3	$4.81 \times 10^{-15}$
$-8 \times 10^{-15}$	$3.20 \times 10^{-14}$	$1.41 \times 10^{15}$	3	$3.30 \times 10^{-15}$
$-7 \times 10^{-15}$	$2.84 \times 10^{-14}$	$1.53 \times 10^{15}$	2	$2.90 \times 10^{-15}$
$8 \times 10^{-15}$	$-3.20 \times 10^{-14}$	$1.42 \times 10^{15}$	3	$-3.17 \times 10^{-15}$
$7 \times 10^{-15}$	$-2.84 \times 10^{-14}$	$1.59 \times 10^{15}$	2	$-2.94 \times 10^{-15}$

where the constrained control signal,  $u(t)$ , is given by

$$-1 \leq u(t) = k_x x(t) + k_r r'(t) \leq 1 \quad (61)$$

and, the inner-loop control system's gains

$$k_x = \begin{bmatrix} -9 & -2.5 \end{bmatrix}, \text{ and } k_r = 8 \quad (62)$$

have been chosen to provide good small signal tracking performance. Note, that  $k_r$  has been chosen so that the system output,  $y = x_1$ , tracks  $r'$  asymptotically. Notice also that the open-loop system dynamics matrix is unstable, with eigenvalues at  $-1$  and  $2$ . Furthermore, statically admissible values of  $r'$  satisfy

$$-r_s \leq r' \leq r_s$$

where, in this example

$$r_s = \frac{CA_{cl}^{-1}B}{1 - k_x A_{cl}^{-1}B} = 1$$

and  $A_{cl} = A + Bk_x$ .

Let  $X_s(r_o)$  denote the set of statically admissible states for a constant  $r = r_o$ . Then the statically admissible set,  $X_s$ , may be manually constructed for the second-order system of eqs. (60) and (61) by obtaining  $X_s(r_o)$  for constant statically admissible reference signals that satisfy  $-r_s \leq r_o \leq r_s$ , and then noting

$$X_s = \bigcup_{-r_s \leq r_o \leq r_s} X_s(r_o)$$

Also, for each  $-r_s \leq r_o \leq r_s$ , statically admissible values of the state vector,  $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ , lie in the intersection of the two half-spaces,

$$\begin{aligned} k_x x - (1 - k_r r_o) &\leq 0 \\ -k_x x - (1 + k_r r_o) &\leq 0. \end{aligned} \quad (63)$$

Thus, for a particular constant, statically admissible reference signal,  $r_o$ ,  $X_s(r_o)$  is the set of all initial states,  $x(0)$ , such that the ensuing trajectory does not violate eq. (63).



Figure 15 shows the hyperplanes and extremal trajectories for  $r_o = 0$  and  $r_o = 0.9$ .  $X_s(0)$  is the region bounded by the two hyperplanes and the two extremal trajectories, while  $X_s(0.9)$  is the region bounded by the right hyperplane and the single extremal trajectory. A family of extremal trajectories for several statically admissible values of  $r_o$  is shown in Figure 16. Notice that as  $r_o$  approaches  $\pm r_s = \pm 1$ , the size of the set of statically admissible states,  $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ , decreases. Moreover, at  $r_o = \pm r_s = \pm 1$ , the statically admissible set reduces to a single point, viz.,  $X_s(\pm 1) = \left\{ \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix} \right\}$ . Moreover, the states  $x = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}$  are not elements of  $X_s(r_o)$  for  $r_o \neq \pm 1$ . Thus, should the system ever reach the valid equilibrium points  $x = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}$  it becomes “stuck”. This phenomenon is a common characteristic of control systems where the open-loop system is unstable and subject to control constraints.

Figure 17 shows  $X_s(0)$  inscribed within  $X_s$ . The set,  $X_s$ , was obtained from the union of all  $X_s(r_o)$  for  $-r_s \leq r_o \leq r_s$ . Notice that while  $X_s$  is convex, it is not finitely determined, as is demonstrated by the two curved segments of its boundary. Finitely determined maximal statically admissible sets are an artifact of discrete time systems.

The maximal positively invariant set,  $X_I$ , is now constructed to illustrate the relationship between it and the maximal statically admissible set,  $X_s$ . The set,  $X_I$ , is obtained by finding all initial states,  $x(0)$ , for which there exists a feasible reference signal trajectory such that the resulting state trajectory,  $x(t)$ , remains in  $X_I$  for all  $t \in Z^+$ . Actually, we must have that  $x(t)$  converges to a statically admissible state,  $x_{ss} \in X_s$ . From eq. (61) it is clear that for a particular state,  $x(t)$ ,  $r(t)$  must satisfy

$$r_f(t)_{\min} \leq r(t) \leq r_f(t)_{\max},$$

where

$$r_f(t)_{\min} = \frac{-1 - k_x x(t)}{k_r}$$

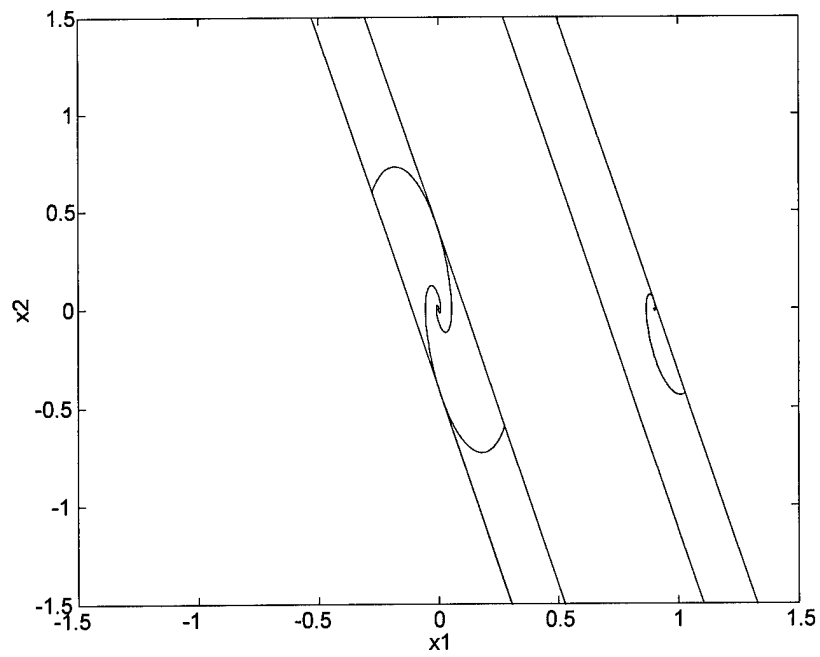


Figure 15. Statically admissible sets  $X_s(0)$  and  $X_s(0.9)$  for the 2D problem.

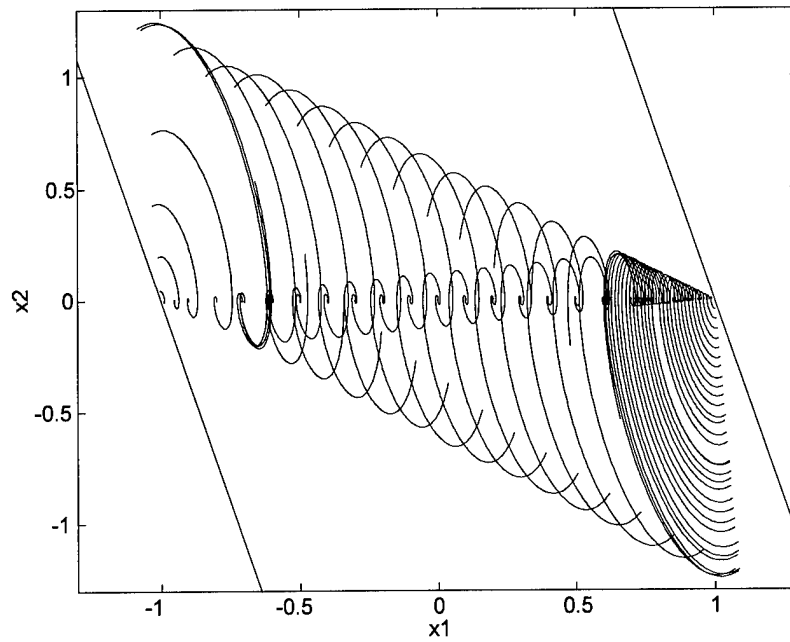


Figure 16. Extremal trajectories for several values of  $r_0$  for the 2D problem.

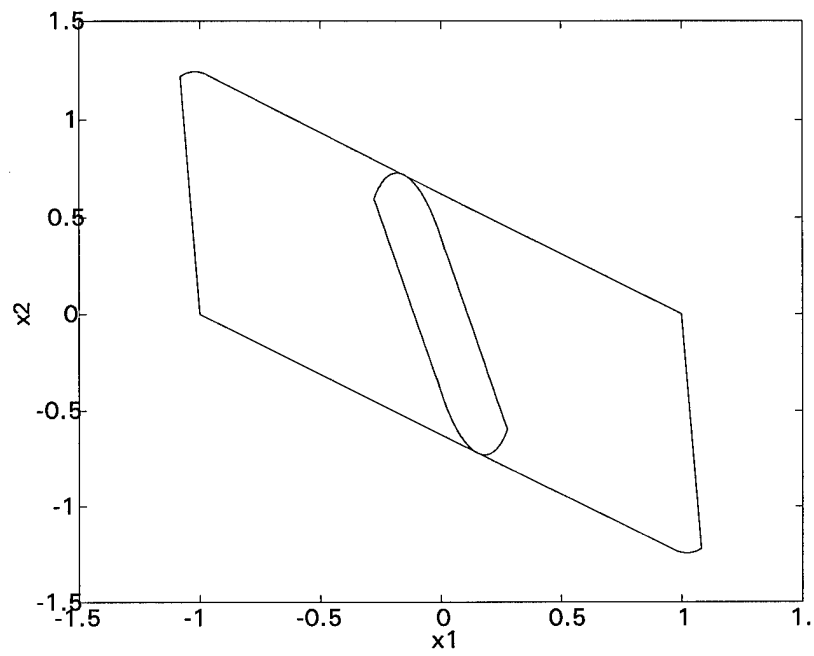


Figure 17. Convex hulls of  $X_s$  and  $X_s(0)$  for the 2D problem.

and

$$r_f(t)_{\max} = \frac{1 - k_x x(t)}{k_r}.$$

Thus, if  $x(0) \in X_I$ , and at each time increment,  $t \in Z^+$ ,  $r(t)$  is chosen to be the solution of the problem

$$\min_{r \in \mathfrak{R}} |r| \text{ subject to } r_f(t)_{\min} \leq r \leq r_f(t)_{\max} \quad (64)$$

then the state trajectory must converge to a statically admissible equilibrium point.

Figure 18 is a plot of  $X_I$ . Initial states in the two regions bounded by dashed lines are elements of  $X_I \cap X_s^c$ . With the exception of those initial states that lie on the dashed boundary segments, choosing  $r(t)$  in accordance with eq. (64) causes the state trajectory to converge to the origin for all initial states in  $X_I$ . State trajectories resulting from initial conditions that lie on the dashed boundary segments converge to the "sticking" points,  $x_{ss} = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}$ . Finally, notice that  $X_s \subset X_I$ , and that  $X_I$  is not convex.

### 3.4.1 The Maximal Statically Admissible Set

Application of the methods described here to obtain a finitely determined approximation to  $X_s$  requires consideration of an equivalent discrete-time system. Using a sampling rate of 100 Hz, i.e.,  $T_s = 0.01$  sec, the equivalent discrete-time system is given by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1.0001 & 1.0051 \times 10^{-2} \\ 2.0101 \times 10^{-2} & 1.0102 \end{bmatrix} x(k) + \begin{bmatrix} 1.0034 \times 10^{-4} \\ 2.0101 \times 10^{-2} \end{bmatrix} u(k) \\ &= A_d x(k) + B_d u(k) \end{aligned} \quad (65)$$

$$y(k) = Cx(k) = [1 \ 0] x(k) \quad (66)$$

$$-1 \leq u(k) = k_x x(k) + k_r r(k) = [-9 \ -2.5] x(k) + 8r'(k) \leq 1. \quad (67)$$

A finite set of inequalities which characterize  $O_\infty^\varepsilon$  may be obtained using the methods of Chapter 2. The augmented system and output constraints are given by eq. (31) with

$$A_g = \begin{bmatrix} 1 & 0 \\ B_{cl} & A_{cl} \end{bmatrix}, B_{cl} = B_d k_r, A_{cl} = A_d + B_d k_x$$

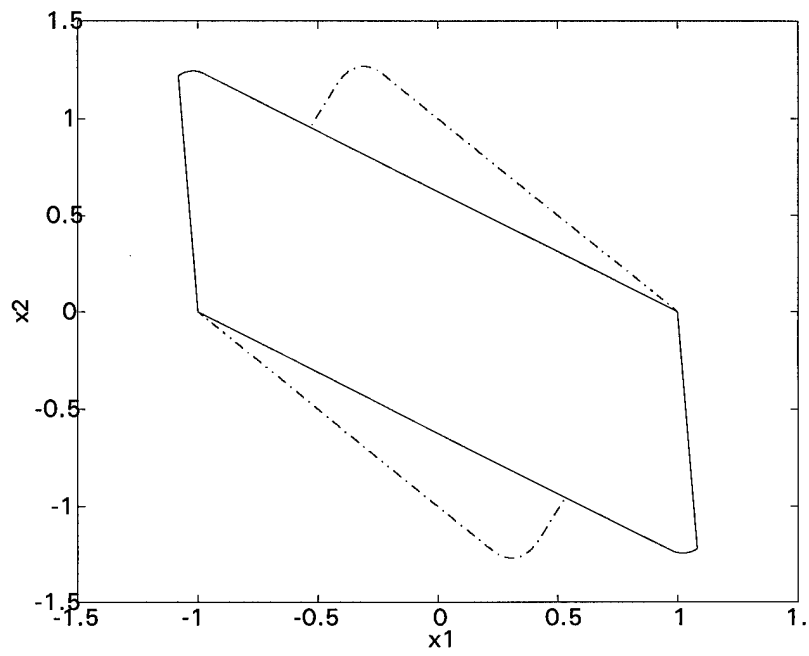


Figure 18. Maximal positively invariant set for the 2D problem

$$B_g = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \text{ and } C_g = [k_r \quad k_x] = [8 \quad -9 \quad -2.5].$$

In this case the constrained output,  $y_c$ , is the constrained control signal,  $u$ . Also, the constraint set,  $Y$ , contains the origin, is bounded, and is given by

$$Y = \{y_c \in \mathfrak{R} : f_i(y_c) \leq 0, \quad i = 1, 2\}$$

where  $f_1(y_c) = y_c - 1$ , and  $f_2(y_c) = -y_c - 1$ . Moreover, the pair  $(A_{cl}, k_x)$  is observable. Thus,  $O_\infty^\varepsilon$  is given by eq. (34) where

$$H_o = k_x (I - A_{cl})^{-1} B_{cl} + k_r = -1.$$

With  $\varepsilon = 0.05$ , the algorithm of Section 2.3 gives  $t_1^* = t_2^* = 121$ . Thus,  $O_\infty^\varepsilon$  is given by

$$O_\infty^\varepsilon = \left\{ x_g \in \mathfrak{R}^3 : f_i(C_g A_g^t x_g) \leq 0, t = 0, \dots, 121, i = 1, 2 \right. \\ \left. \text{and } -0.95 \leq r' \leq 0.95 \right\}. \quad (68)$$

Recall, that eq. (35) is an equivalent expression for  $O_\infty^\varepsilon$ . In this case we have  $\Gamma_g \in \mathfrak{R}^{246 \times 3}$  and  $\beta_g \in \mathfrak{R}^{246}$ . Also,  $\Gamma_g$  and  $\beta_g$  are available from the computations performed to determine  $t_i^*$ .

The set of vertices,  $V$ , of the polygon,  $X_s^\varepsilon$ , are obtained from the Linear Program (54). In this problem, two rows of  $\Gamma_g$  result in  $c^T = 0$ , and are neglected. The Linear Program of eq. (54) generates 244 vertices. However, only 86 of these vertices are unique. Thus,  $V$  has 86 elements. Figure 19 is a plot of all 86 vertices.

Notice that the vertices are clustered in two large and four smaller groups. As noted earlier, this suggests that, by eliminating closely spaced vertices to obtain a set,  $V' \subset V$ , we may generate a characterization of a set,  $X_s^\delta \subset X_s^\varepsilon$ , with substantially fewer linear inequality constraints, and with little loss of volume.

The recursive convex hull algorithm of Section 3 is used to generate the set of linear inequalities (52) that characterize  $X_s^\varepsilon$ . In this two dimensional case, the 86 vertices in  $V$  result in 86 linear inequality constraints. Figure 20 is a plot of the 86 supporting hyperplanes of  $X_s^\varepsilon$ . Comparing

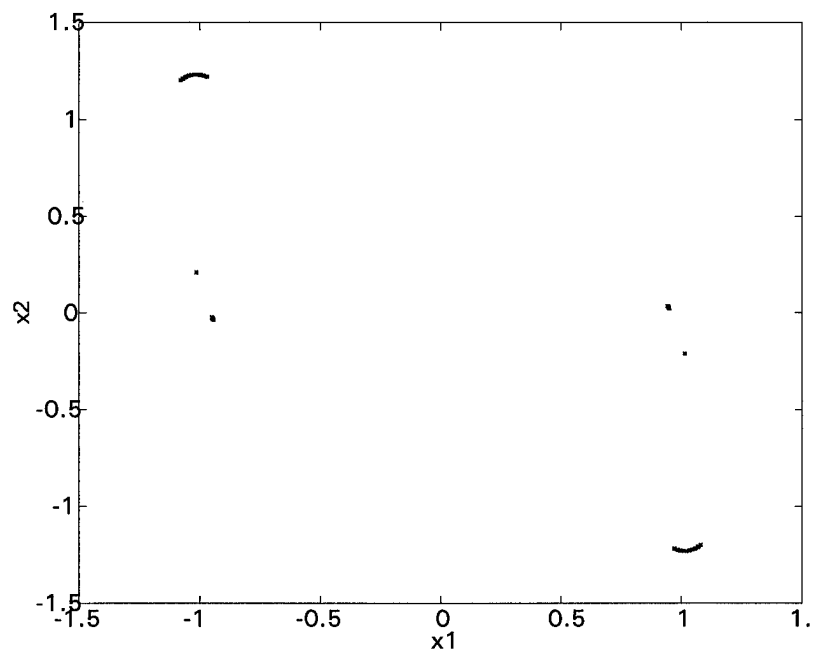


Figure 19. Vertices of  $X_s^\epsilon$  for the 2D problem.

Figures 20 and 17 shows that the most noticeable difference is that two small areas, one at the lower left corner and another at the upper right corner of  $X_s$ , have been shaved off. In fact,  $X_s^\varepsilon$  limits  $x_1$  such that  $-0.9556 \leq x_1 \leq 0.9556$ , and does not contain the states  $x = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}$ . This is due to the  $\varepsilon$  approximation used to obtain  $O_\infty^\varepsilon$ , and is desirable in light of the above mentioned “sticking” phenomenon.

To illustrate the effects of eliminating closely spaced vertices, two different subsets of  $X_s^\varepsilon$  ( $X_s^{\delta_1}$  and  $X_s^{\delta_2}$ ) are characterized using two different subsets of  $V$  ( $V_1'$  and  $V_2'$ ). Each subset is generated by requiring that all elements be a specified Euclidean distance,  $d$ , from each other. The set  $V_1'$  contains 26 vertices, and is obtained with  $d = 0.009$ , and  $V_2'$  contains 10 vertices obtained with  $d = 0.05$ . Given  $V_1'$  and  $V_2'$ , the recursive convex hull algorithm generates the linear inequality constraints of eq. (53). The sets  $X_s^{\delta_1}$  and  $X_s^{\delta_2}$  are characterized with 26 and 10 linear inequality constraints respectively.

Figure 21 shows the 26 supporting hyperplanes associated with  $X_s^{\delta_1}$ . Notice that the loss of volume is minimal. Moreover, the set of statically admissible equilibrium points allowed by  $X_s^{\delta_1}$  is identical to that of  $X_s^\varepsilon$ , viz., the line segment  $-0.9556 \leq x_1 \leq 0.9556$ . However, there are only 26 inequalities required to characterize  $X_s^{\delta_1}$ , versus the 86 inequalities of  $X_s^\varepsilon$ , or the 246 inequalities required to characterize  $O_\infty^\varepsilon$ . Figure 22 shows the 10 supporting hyperplanes associated with  $X_s^{\delta_2}$ . While Figure 22 shows a slight reduction in volume, this may be acceptable in light of the reduced computational burden.

Notice that for this problem 86 inequalities are required to characterize  $X_s^\varepsilon$ , but a very good approximation is obtained using only 10 inequalities. Thus, while the method of [9] requires evaluation of 246 inequalities at each time step to insure  $x(k+1) \in O_\infty$ , only 10 inequalities must be evaluated to insure  $x(k+1) \in X_s^\varepsilon$ . Not only does this method provide improved tracking performance, but the computational burden is reduced.



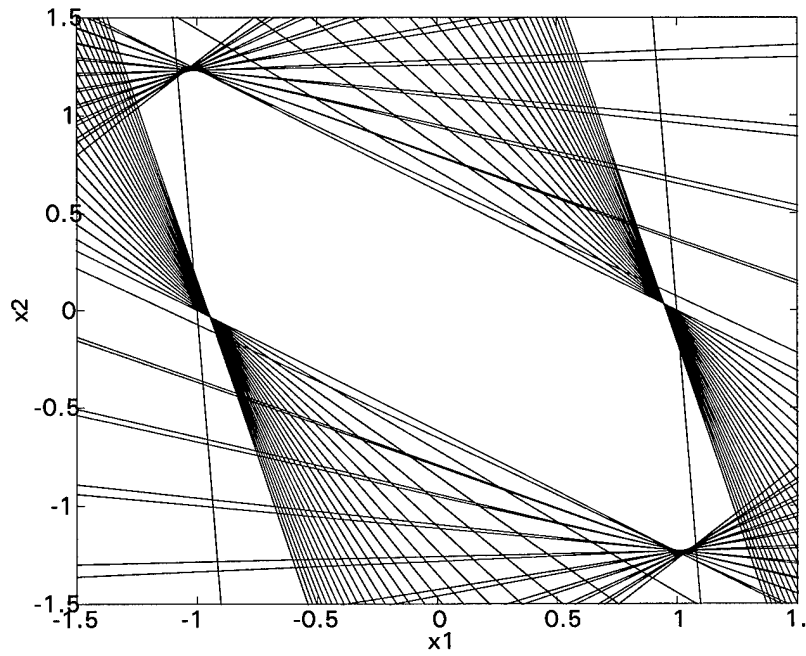


Figure 20. Supporting hyperplanes for  $X_s^\epsilon$  for the 2D problem.

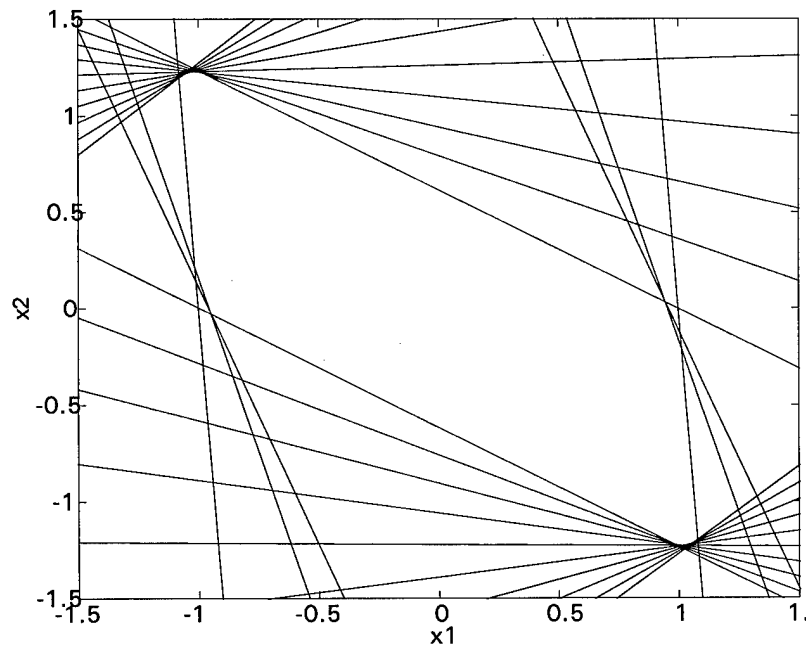


Figure 21. Supporting hyperplanes for  $X_s^{\delta_1}$  for the 2D problem.

The trade-off between computational burden and steady state responses allowed by  $X_s^\delta$ , for various subsets of  $V$ , may be computed from the resultant linear inequality constraints of eq. (53).

In steady state  $x(t+1) = x(t) = x_{ss}$  and  $r'_{ss} = y_{ss}$ . Then,  $x_{ss} \in X_s^\delta$  requires that

$$\Gamma_\delta (I - A_{cl})^{-1} B_{cl} r'_{ss} \leq \beta_\delta. \quad (69)$$

Let  $\Gamma_\delta$  be an  $N \times n$  matrix and define

$$\Psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_N \end{bmatrix} = \Gamma_\delta (I - A_{cl})^{-1} B_{cl}.$$

Since each element of  $\beta_\delta$  is 1, and the origin is in the interior of  $X_s^\delta$ , then

$$r'_{ss_{\max}} = \frac{1}{\max(\psi_i)}$$

and

$$r'_{ss_{\min}} = \frac{1}{\min(\psi_i)}.$$

This provides a means to determine if a particular set,  $X_s^\delta$ , supports a desired set of equilibria points.

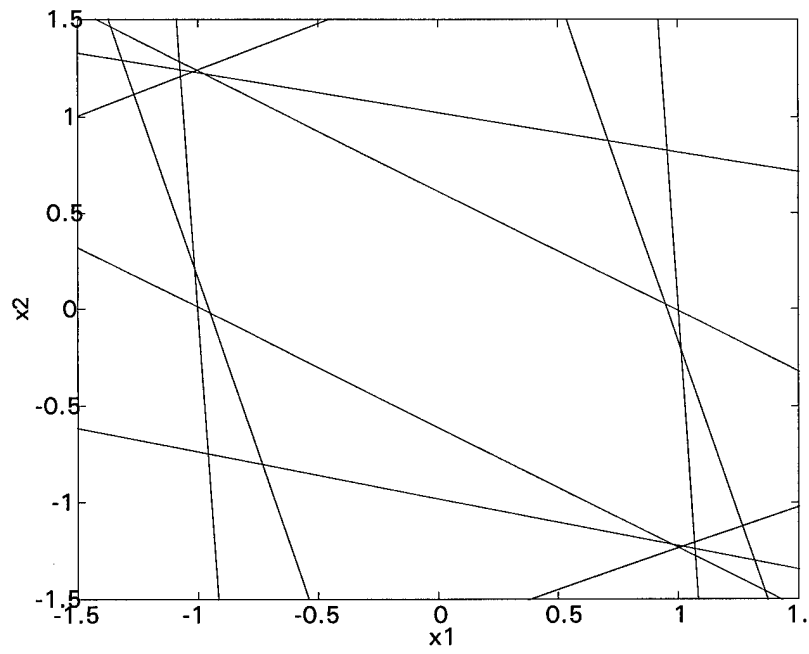


Figure 22. Supporting hyperplanes for  $X_s^{\delta_2}$  for the 2D problem.

### 3.5 Summary

In this Chapter a procedure was developed to project the maximal output admissible set,  $O_{\infty}^{\varepsilon} \subset \mathfrak{R}^{n+1}$ , onto the  $\mathfrak{R}^n$  subspace to obtain an approximation of the maximal statically admissible set,  $X_s^{\varepsilon} \subset \mathfrak{R}^n$ . It was also noted that the boundary of  $X_s^{\varepsilon}$  generally contains several segments with closely spaced vertices. By thinning these clusters of vertices, the subsets,  $X_s^{\delta} \subset X_s^{\varepsilon}$ , were obtained. Since  $X_s^{\delta}$  contains fewer vertices, it is characterized with fewer linear inequalities. Also, since only closely spaced vertices are eliminated, the volume of  $X_s^{\delta}$  is similar to that of  $X_s^{\varepsilon}$ . Thus, a substantial reduction in on-line computational burden may be realized with little loss in closed-loop system performance. The concepts and methodology developed here were illustrated with a second order example.

A recursive convex hull algorithm was developed to generate the linear inequalities that characterize  $X_s^{\varepsilon}$ . Unlike existing convex hull algorithms the algorithm developed here does not generate the complete facial graph, but only the list of vertices contained in each facet, as well as the desired linear inequalities. Thus, this algorithm may be more efficient than existing algorithms. While not required for the problem addressed here, the recursive nature of this algorithm also makes it amenable to on-line operation. Moreover, this algorithm is applicable to problems with interior points, and to problems where the points are not in general position.

## Chapter 4 - Constraint Avoidance Methods

### 4.1 Overview

In the previous chapter the set  $X_s^\varepsilon \subset \mathfrak{R}^n$  was characterized with a set of linear inequalities by projecting the polytope,  $O_\infty^\varepsilon \subset \mathfrak{R}^{n+1}$ , onto the  $\mathfrak{R}^n$  subspace. The set  $X_s^\delta \subset X_s^\varepsilon$  was also obtained by eliminating closely spaced vertices of  $X_s^\varepsilon$ . Since it contains fewer vertices the number of inequalities required to characterize  $X_s^\delta$  is fewer than those required to characterize  $X_s^\varepsilon$ .

In this chapter specific reference signal governor (RSG) algorithms are developed. The constraint avoidance strategy employed here is to choose  $r'(t)$  at each time step so that  $x(t+1) \in X_s^\varepsilon$  and  $y_c(t) \in Y$ . This allows the greatest flexibility in selecting  $r'$  at each time step, while also providing a BIBO stable closed-loop system. In particular,  $r'$  is not restricted to statically admissible values at each time step. Examples are used to illustrate the potential improvement in tracking performance this can provide. Constraint avoidance methods based on polyhedral and elliptical subsets of the set  $X_s^\varepsilon$  are also presented. These methods can result in a substantial reduction in the RSG on-line computational burden.

### 4.2 Constraint Avoidance Strategy

The definitions concerning admissible and feasible reference signals first presented in Chapter 2 play an important role in the proposed constraint avoidance strategy, and are reviewed here. Given  $x(t) \in X_s^\varepsilon$ , if  $r'(t)$  is chosen such that  $x(t+1) \in X_s^\varepsilon$ , then  $r'(t)$  is said to be admissible with respect to the set  $X_s^\varepsilon$ . Also, if  $r'(t)$  is chosen such that  $y_c(t) \in Y$  then  $r'(t)$  is said to be feasible. Given  $x(t) \in X_s^\varepsilon$  it may be possible to choose an  $r'(t)$  that is admissible, but not feasible. In this case the hard constraints will be encountered and saturation will ensue. On the other hand it may be possible to choose an  $r'(t)$  that is feasible but not admissible. In this case the hard constraints are not immediately violated, but  $x(t+1) \notin X_s^\varepsilon$  will transpire and an unstable system response

may ensue. Thus, both admissibility and feasibility constraints must be enforced at each time step to obtain a guaranteed BIBO stable closed-loop system. This requires that the admissibility and feasibility constraints be consistent for any  $x \in X_s^\varepsilon$ . That is, the set of reference signals that are both admissible and feasible must be non-empty for any  $x \in X_s^\varepsilon$ . The following theorem addresses this issue.

**Theorem 1.** *For any  $x(t) \in X_s^\varepsilon$  there exists an  $r'(t)$  such that  $x(t+1) \in X_s^\varepsilon$  and  $y_c(t) \in Y$ .*

Proof:

Let  $x_o \in X_s^\varepsilon$  be arbitrary but fixed. Because  $X_s^\varepsilon$  is the projection of  $O_\infty^\varepsilon$  onto the  $\mathbb{R}^n$  subspace, there exists at least one statically admissible constant reference signal,  $r'_o \in R_s^\varepsilon$ , such that  $x_{g_o} = \begin{bmatrix} r'_o \\ x_o \end{bmatrix} \in O_\infty^\varepsilon$ . Also, from the definition of the set  $O_\infty^\varepsilon$ , if  $x_g = \begin{bmatrix} r' \\ x \end{bmatrix} \in O_\infty^\varepsilon$ , then  $x \in X_s^\varepsilon$  and  $y_c \in Y$ . In addition, if  $x_g(0) = \begin{bmatrix} r'_o \\ x(0) \end{bmatrix} \in O_\infty^\varepsilon$ , then  $x_g(t) = \begin{bmatrix} r'_o \\ x(t) \end{bmatrix} \in O_\infty^\varepsilon$  for all  $t \in Z^+$ . Thus, given  $x(t) = x_o$ , let  $r'(t) = r'_o$ . Then,  $x(t+1) \in X_s^\varepsilon$  and  $y_c(t) \in Y$ .  $\square$

It follows that if  $x(0) \in X_s^\varepsilon$  and at each time increment the RSG chooses  $r'(t)$  such that  $x(t+1) \in X_s^\varepsilon$  and  $y_c(t) \in Y$ , then  $x(t) \in X_s^\varepsilon$  and  $y_c(t) \in Y$  for all  $t \in Z^+$ , and the closed-loop system will be BIBO stable. In the future, a reference signal that is both admissible and feasible is said to be allowable.

Unlike the DTRG, the RSG relies on the existence of a time varying allowable reference signal to insure  $x(t) \in X_s^\varepsilon$  and  $y_c(t) \in Y$  for all  $t \in Z^+$ , and does not require that at any time increment  $r'(t)$  be such that it is allowable for all  $\tau > t$ . Also, the RSG does not restrict the modified reference signal to the set of statically admissible reference signals, vis.,  $R_s^\varepsilon$ . This additional flexibility in selecting  $r'(t)$  may be used to improve tracking performance over that obtained with the DTRG system. At the same time the RSG constraint avoidance strategy results in a BIBO stable closed-loop system.

### 4.3 Determination of the Modified Reference Signal

In this section explicit algorithms are developed which compute  $r'(t)$  at each time increment. The single input problem is exclusively considered. First the set of allowable reference signals is determined based on the admissibility and feasibility constraints. Then two control concepts are specified which choose the best allowable reference signal based on two different metrics or criteria.

Admissibility of  $r'(t)$  with respect to the set  $X_s^\varepsilon$  requires that  $x(t+1) \in X_s^\varepsilon$ . In terms of the linear inequalities that characterize  $X_s^\varepsilon$  this translates into

$$\Gamma_\varepsilon x(t+1) \leq \beta_\varepsilon.$$

Since  $x(t+1) = A_{cl}x(t) + B_{cl}r'(t)$ , the admissibility requirement may also be expressed as

$$\Gamma_\varepsilon B_{cl}r'(t) \leq \beta_\varepsilon - \Gamma_\varepsilon A_{cl}x(t). \quad (70)$$

Upper and lower bounds imposed on  $r'(t)$  by the admissibility constraints are easily determined at each time increment by evaluating the linear inequalities of eq. (70). First, let

$$\Gamma_\varepsilon B_{cl} = \Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} \quad \text{and} \quad \beta_\varepsilon - \Gamma_\varepsilon A_{cl}x(t) = \Phi(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_N(t) \end{bmatrix}. \quad (71)$$

Also, let  $q_1$  denote the set of indices for which  $\theta_i > 0$ , and  $q_2$  denote the set of indices for which  $\theta_i < 0$ . That is, let  $q_1 = \{i \in \{1, 2, \dots, N\} : \theta_i > 0\}$ , and  $q_2 = \{i \in \{1, 2, \dots, N\} : \theta_i < 0\}$ .

Then,  $r'(t)$  is admissible if

$$r_a(t)_{\min} \leq r'(t) \leq r_a(t)_{\max} \quad (72)$$

where  $r_a(t)_{\min}$  and  $r_a(t)_{\max}$  are the minimum and maximum admissible values of  $r'(t)$ , and are given by

$$r_a(t)_{\max} = \min_{i \in q_1} \left\{ \frac{\phi_i(t)}{\theta_i} \right\} \quad (73)$$

and

$$r_a(t)_{\min} = \max_{i \in q_2} \left\{ \frac{\phi_i(t)}{\theta_i} \right\}. \quad (74)$$

Feasibility of  $r'(t)$  requires that  $y_c(t) = C_{c_{cl}}x(t) + D_{c_{cl}}r'(t) \in Y \subset \mathbb{R}^p$ . In the case where the constraint set  $Y$  is a polytope, an equivalent expression is

$$\rho_{\min} \leq C_{c_{cl}}x(t) + D_{c_{cl}}r'(t) \leq \rho_{\max} \quad (75)$$

where  $\rho_{\min}$  and  $\rho_{\max}$  are  $p$ -dimensional vectors. Equation (75) represents  $2p$  inequalities. Now, let

$$D_{c_{cl}} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix}$$

and, define the sets  $q_3 = \{i \in \{1, 2, \dots, p\} : d_i > 0\}$ , and  $q_4 = \{i \in \{1, 2, \dots, p\} : d_i < 0\}$ . Then,  $r'(t)$  is feasible if

$$r_f(t)_{\min} \leq r'(t) \leq r_f(t)_{\max} \quad (76)$$

where  $r_f(t)_{\min}$  and  $r_f(t)_{\max}$  are the minimum and maximum feasible values of  $r'(t)$ , and are given by

$$r_f(t)_{\max} = \min \left\{ \min_{i \in q_3} \left\{ \frac{\rho_{\max} - C_{c_{cl}}x(t)}{d_i} \right\}, \min_{i \in q_4} \left\{ \frac{\rho_{\min} - C_{c_{cl}}x(t)}{d_i} \right\} \right\} \quad (77)$$

and

$$r_f(t)_{\min} = \max \left\{ \max_{i \in q_3} \left\{ \frac{\rho_{\min} - C_{c_{cl}}x(t)}{d_i} \right\}, \max_{i \in q_4} \left\{ \frac{\rho_{\max} - C_{c_{cl}}x(t)}{d_i} \right\} \right\}. \quad (78)$$

The set of all allowable reference signals is the intersection of the sets of admissible and feasible reference signals, and is given by

$$r'(t)_{\min} \leq r'(t) \leq r'(t)_{\max} \quad (79)$$

where

$$r'(t)_{\max} = \min \{r_f(t)_{\max}, r_a(t)_{\max}\}$$

and

$$r'(t)_{\min} = \max \{r_f(t)_{\min}, r_a(t)_{\min}\}.$$

Also, since the set of allowable reference signals must be non-empty when  $x(t) \in X_s^\varepsilon$ , then  $r'(t)_{\min} \leq r'(t)_{\max}$  is guaranteed. Now that the set of allowable reference signals has been determined, the reference signal governor selects the best allowable reference signal based on the specified metric.



### 4.3.1 Control Concepts

In addition to insuring  $r'(t)$  is allowable at each time increment we must also define a metric to determine the "optimal" modified reference signal from the set of all allowable reference signals. Two possible metrics, which are consistent with those used in the scalar example of Chapters 1 and 2, are presented next.

Let  $R_a(t)$  denote the set of all allowable reference signals at time increment  $t$ . A reasonable control concept is to minimize the distance between the exogenous and modified reference inputs, subject to the constraints. In this case  $r'(t)$  is chosen so as to solve the minimization problem

$$\min_{r'(t) \in R_a(t)} \|r(t) - r'(t)\| \quad (80)$$

This is referred to as the minimum distance reference signal governor (MDRSG) and results in

$$r'(t) = \begin{cases} r'(t)_{\max}, & r(t) \geq r'(t)_{\max} \\ r(t), & r'(t)_{\min} \leq r(t) \leq r'(t)_{\max} \\ r'(t)_{\min}, & r(t) \leq r'(t)_{\min} \end{cases} \quad (81)$$

The second control concept attempts to drive the tracking error,  $e = r - y$ , to zero as quickly as possible. In this case, if  $|e(t)| > \tau$ , then  $r'(t)$  is chosen so as to minimize  $|e(t+1)|$  subject to the constraints. Here,  $\tau > 0$  is a predetermined tracking error tolerance which prevents chattering and oscillatory responses. When  $|e(t)| \leq \tau$  we revert to eq. (81). Specifically, the control concept is

$$\begin{aligned} \text{If } e > \tau \text{ then } & \max_{r'(t) \in R_a(t)} (r') \\ \text{If } e < -\tau \text{ then } & \min_{r'(t) \in R_a(t)} (r') \\ \text{If } |e| \leq \tau \text{ then } & \min_{r'(t) \in R_a(t)} (r(t) - r'(t)) \end{aligned} \quad (82)$$

This is referred to as the saturating reference signal governor (SRSG), and results in

$$r'(t) = \begin{cases} r'(t)_{\max}, & r(t) - y(t) \geq \tau \\ r'(t)_{\min}, & r(t) - y(t) \leq -\tau \\ r(t), & |r(t) - y(t)| < \tau \text{ and } r'(t)_{\min} \leq r(t) \leq r'(t)_{\max} \\ r'(t)_{\max}, & |r(t) - y(t)| < \tau \text{ and } r(t) \geq r'(t)_{\max} \\ r'(t)_{\min}, & |r(t) - y(t)| < \tau \text{ and } r(t) \leq r'(t)_{\min} \end{cases} \quad (83)$$

The MDRSG control concept is similar to that of the DTRG in that both control concepts attempt to minimize the difference between the exogenous and modified reference signals. However,

the MDRSG constraints are less restrictive in that at any particular time step,  $t$ ,  $r'(t)$  is not required to be statically admissible. Thus, the MDRSG may provide improved tracking performance in some cases. The SRGS is the most aggressive of the three control concepts in that it attempts to drive the tracking error to within a prespecified tolerance as fast as possible. As the following examples show, of the three control concepts, the SRSG has the potential to provide the best tracking performance. While the MDRSG preserves small signal performance, the SRSG modifies small signal performance. Both the MDRSG and SRSG control concepts result in BIBO stable closed-loop systems, and the computational burden imposed by either control concept is a function of the number of inequality constraints required to characterize  $X_s^\varepsilon$ .

Other control concepts are also possible. Specifically, a BIBO stable LQT controller could be developed using one step ahead constraint enforcement. In this case, any reference signal prediction scheme can be used, and the constraints of [17] are replaced with those developed here to obtain a BIBO stable closed-loop system.

#### 4.4 Simulation Results

Three examples are presented here based on the two dimensional system of Section 3.4. Along with the controlled process of Section 3.4, two additional examples are developed in which the open-loop plant of eq. (65) is combined with different state feedback matrices to obtain two different overdamped controlled processes. The trade-off between achievable tracking performance and the size of  $X_s^\varepsilon$  is then examined. The specific dual-loop controller architecture for this problem is depicted in Fig. 23. Simulations results are presented for the MDRSG, SRSG, and DTRG control concepts. System responses to both statically admissible and statically inadmissible reference signals are also presented.

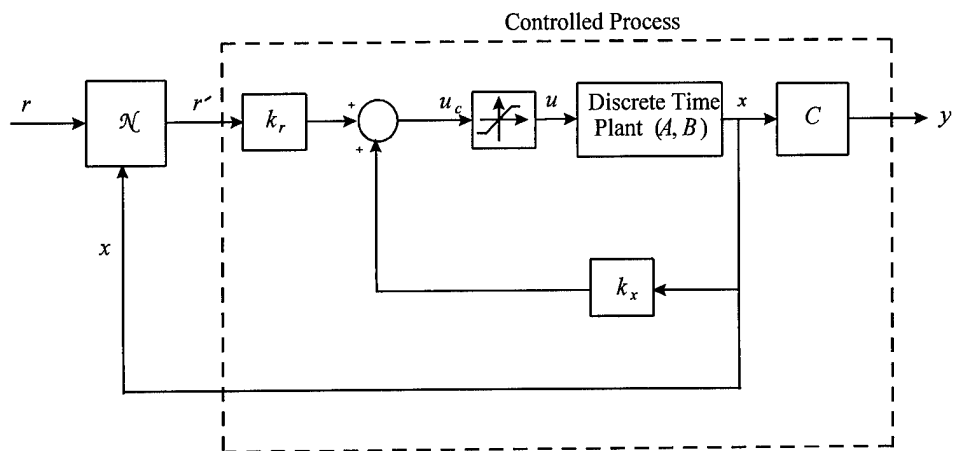


Figure 23. Nonlinear control system.

#### 4.4.1 Example One

The discrete-time model for the controlled process of Section 3.4 is

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 0.999197 & 9.799662 \times 10^{-3} \\ -0.160808 & 0.959898 \end{bmatrix} x(t) + \begin{bmatrix} 8.026867 \times 10^{-4} \\ 1.608080 \times 10^{-1} \end{bmatrix} r'(t) \\ &= A_{cl}x(t) + B_{cl}r'(t) \\ y(t) &= Cx(t) = [1 \ 0] x(t). \end{aligned} \quad (84)$$

The control signal is subject to hard constraints and is given by

$$-1 \leq u(t) = k_x x(t) + k_r r'(t) \leq 1 \quad (85)$$

where

$$k_x = [ -9 \quad -2.5 ], \text{ and } k_r = 8.$$

In terms of eq. (75) the output constraint is

$$-1 \leq y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \leq 1$$

where  $C_{ccl} = k_x$  and  $D_{ccl} = k_r$ .

Recall that for this system  $X_s^\varepsilon$  is characterized with 86 linear inequalities. Thus,  $\Gamma_\varepsilon$  is an  $86 \times 2$  matrix, and  $\beta_\varepsilon$  is an 86 element vector. At each time increment, the minimum and maximum admissible reference signal bounds,  $r_a(t)_{\min}$  and  $r_a(t)_{\max}$ , are determined by evaluating the 86 inequalities in accordance with eqs. (70)-(74). Also, the minimum and maximum feasible reference signal bounds are given by

$$r_f(t)_{\min} = \frac{-1 - k_x x(t)}{k_r} \quad (86)$$

and

$$r_f(t)_{\max} = \frac{1 - k_x x(t)}{k_r}. \quad (87)$$

### Example 1a.

System responses to a statically admissible input are investigated first. The initial condition and exogenous reference signal are

$$x(0) = [0.5, 0]^T$$

and

$$r(t) = \begin{cases} 0.5, & 0 \leq t \leq 10 \\ -0.5, & 11 \leq t \leq 500 \end{cases} .$$

The desired final state is  $x(500) = [-0.5, 0]^T$ . The state trajectory and control signal responses obtained for the system with no reference signal governor are shown in Figures 24 and 25. Figure 24 shows the state trajectory superimposed on a plot of the boundary of  $X_s^\varepsilon$ , denoted by  $\partial X_s^\varepsilon$ . Although the initial condition, final state, and exogenous reference signal are all statically admissible in this example, in the absence of saturation effects mitigation measures, the control signal hard constraints are encountered, and an unstable system response ensues.

Figures 26-29 show the MDRSG system responses to the same initial condition and exogenous reference signal. Figure 26 is a plot of the state trajectory superimposed on  $\partial X_s^\varepsilon$ . As designed, the MDRSG restricts the state trajectory to the invariant set  $X_s^\varepsilon$ . In deed, the state trajectory traces a segment of  $\partial X_s^\varepsilon$  enroute to the final state. An important point is that when the state trajectory encounters a segment of  $\partial X_s^\varepsilon$  the linear inequalities associated with that segment of  $\partial X_s^\varepsilon$  become the active admissibility constraints.

The control signal response is shown in Figure 27. The minimum and maximum control signals generated for this example were  $u_{\min} = -1.0$ , and  $u_{\max} = 0.9999934$ . Hard limits were not imposed on the control signal in this case. Thus, the control constraints are respected by the MDRSG system. Limits imposed on the modified reference signal by the admissibility and feasibility constraints are plotted in Figure 28. Clearly, the constraints are consistent at all times. The modified reference signal is shown in Figure 29 along with an expanded view of the constraints. Figures 27

and 29 show that the feasibility constraints are active between 0.1 and 0.6 seconds. At about 0.6 seconds the state trajectory encounters  $\partial X_s^\varepsilon$  and the admissibility constraints become active. At about 2 seconds the state trajectory leaves  $\partial X_s^\varepsilon$ , both the feasibility and admissibility constraints become inactive, and  $r'(t) = r(t) = -0.5$ . After 2 seconds, when all constraints are inactive, the reference governor becomes transparent, and a linear system response ensues. Thus, small signal performance is preserved. Figures 29 and 27 also demonstrate the nonlinear nature of the reference signal governor.

System responses obtained with the SRSG were very similar to those obtained with the MDRSG for this example. Recall that the SRSG generates the largest magnitude allowable modified reference signal whenever the tracking error is greater than some prespecified tolerance. A tracking error tolerance of  $\tau = 0.05$  was used in this case. The only noticeable difference is in the control signal response, Figure 30. Notice that, unlike the MDRSG control signal response, the control signal is driven to its upper limit in the interval between 2.1 and 2.4 seconds. However, this has a negligible impact on the state trajectory. The lightly damped nature of the controlled process in this example,  $\zeta = 0.5$ , precludes use of a tracking error tolerance much less than  $\tau = 0.05$ . Also, the admissibility constraints are active much of the time the tracking error is greater than  $\tau = 0.05$ . The SRSG control concept is clearly better suited to highly damped systems, and situations where the state trajectory does not encounter  $\partial X_s^\varepsilon$  for an extended period of time. This will be demonstrated with the next two examples.

The DTRG tracking performance is also similar to that of the MDRSG, Fig. 31. However, Fig. 32 shows that when the state trajectory is on  $\partial X_s^\varepsilon$  the DTRG control signal response exhibits chatter. This is a common phenomenon associated with the DTRG.

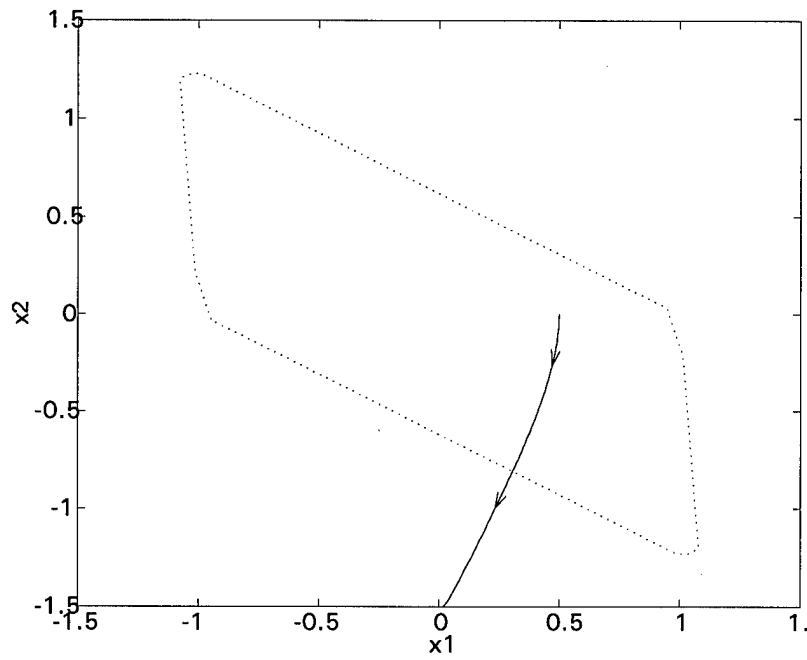


Figure 24. System response for Example 1a with no constraint avoidance measures.

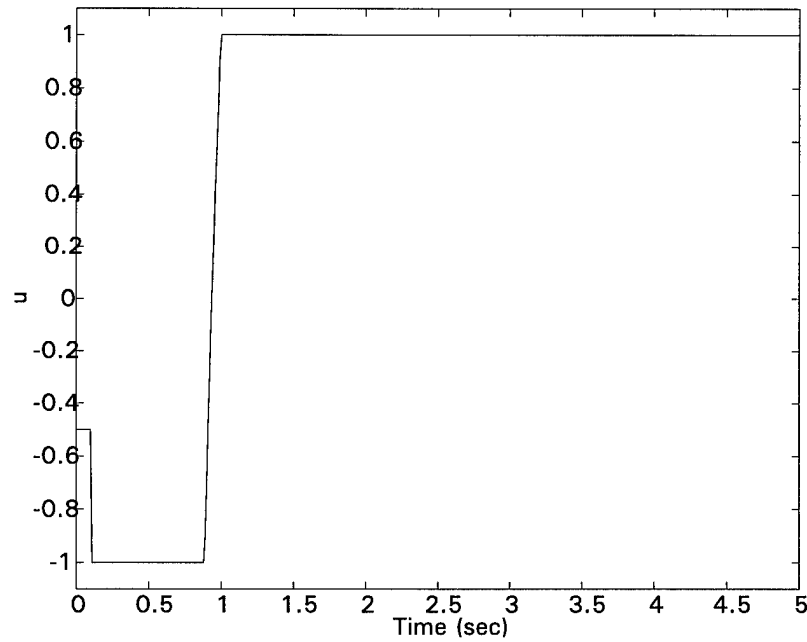


Figure 25. Control signal response for Example 1a with no constraint avoidance measures.

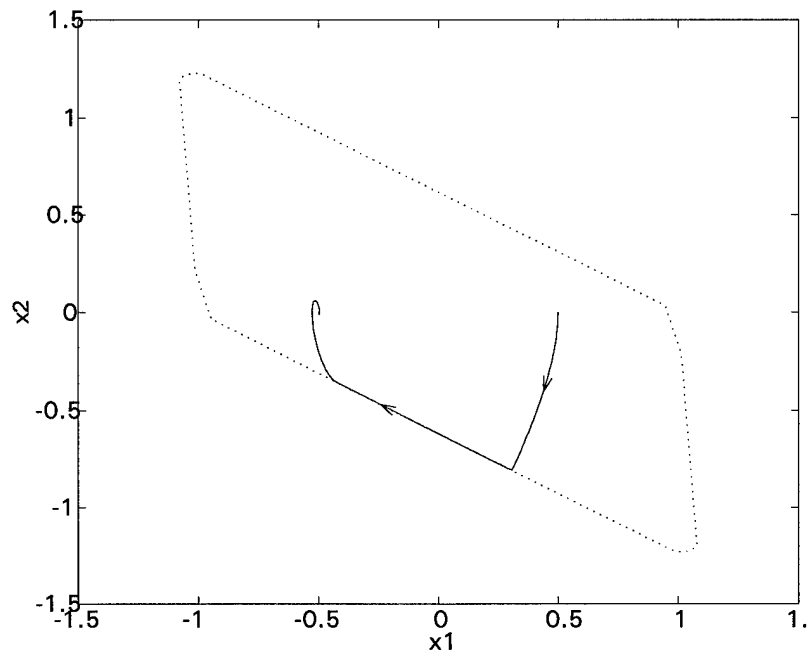


Figure 26. MDRSG state trajectory for Example 1a.

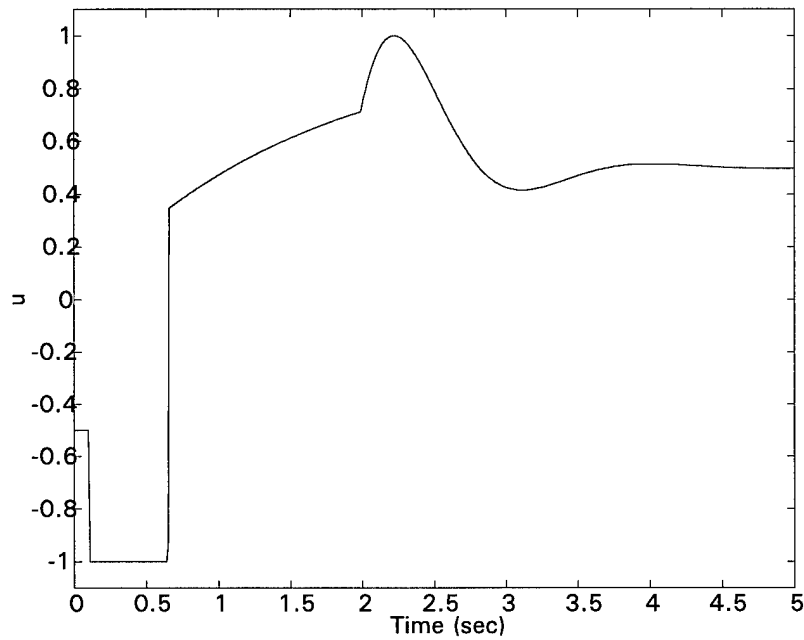


Figure 27. MDRSG control signal response for Example 1a.



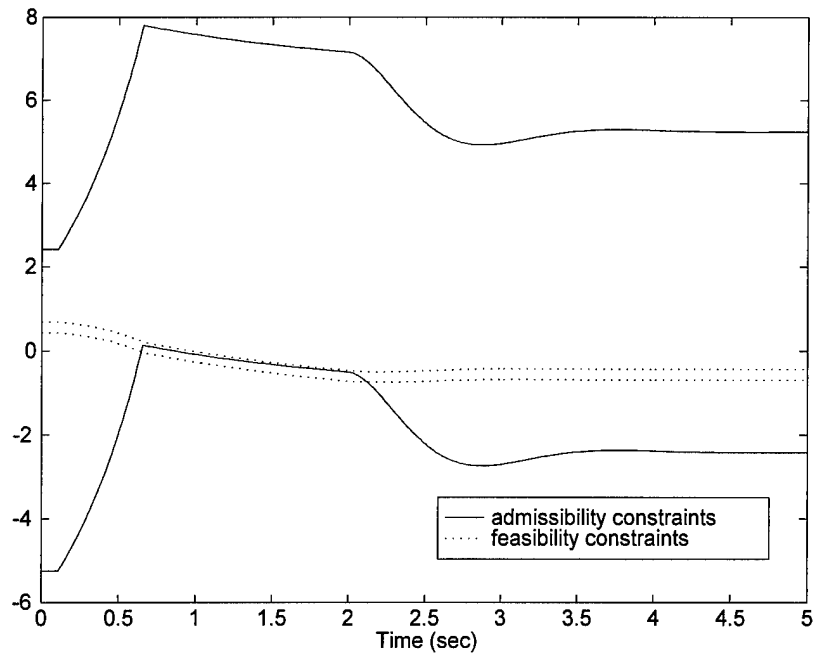


Figure 28. MDRSG reference signal constraints for Example 1a

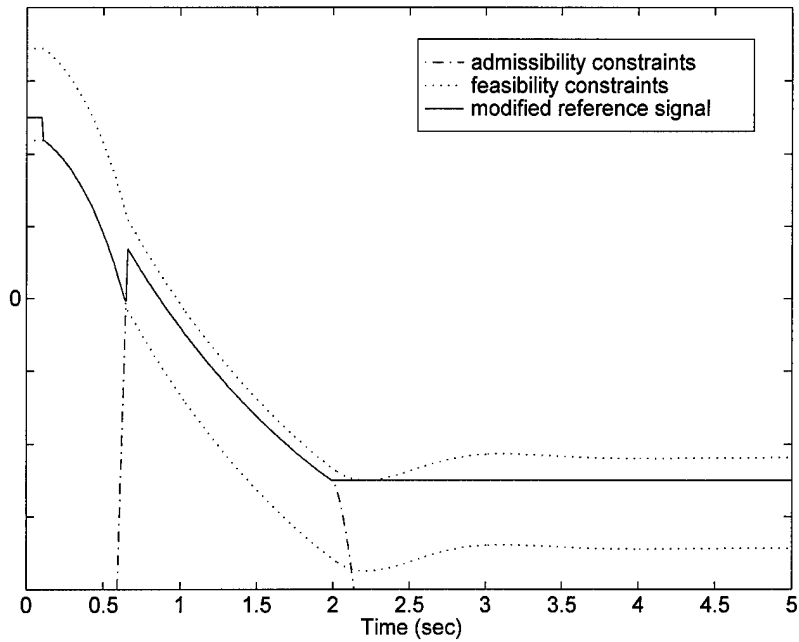


Figure 29. Modified reference signal response for Example 1a.

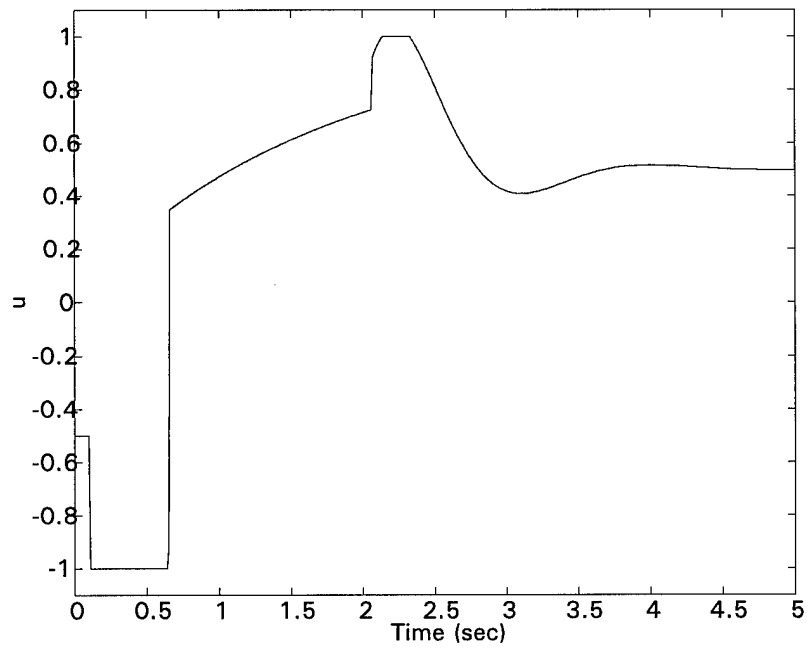


Figure 30. SRS system control signal response for Example 1a.

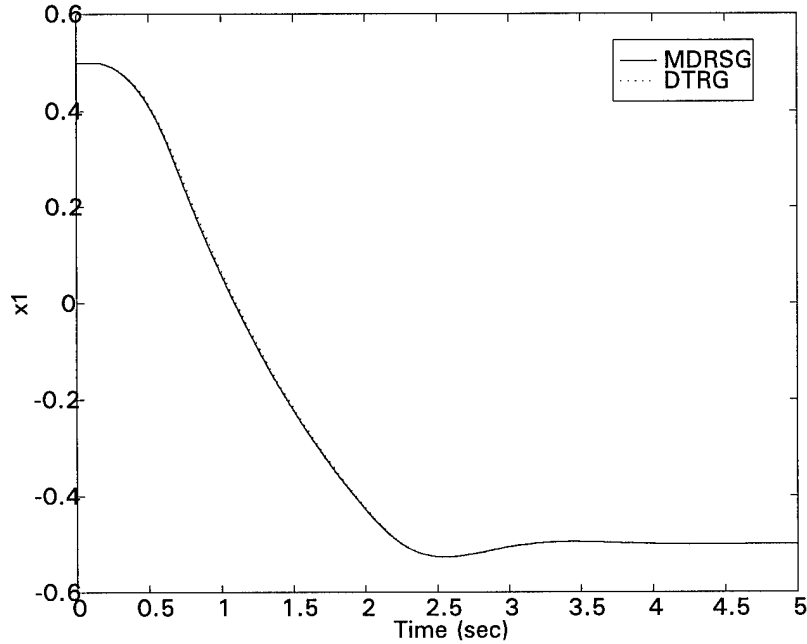


Figure 31. Comparison of DTRG and MDRSG tracking performance for Example 1a.

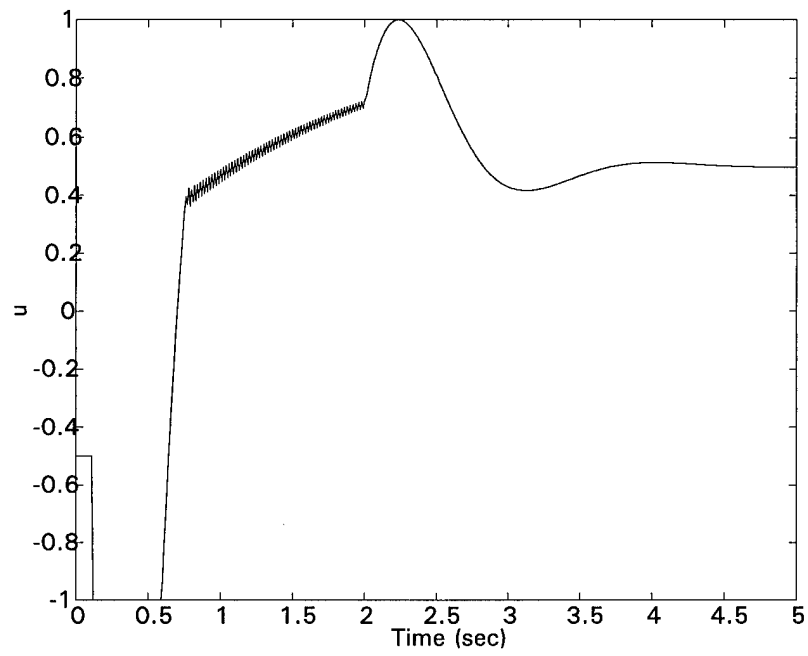


Figure 32. DTRG control signal response for Example 1a.

### Example 1b.

System responses to statically inadmissible exogenous reference signals are now investigated.

The initial condition and exogenous reference signal are

$$x(0) = [ 0.9, 0 ]^T$$

and

$$r(t) = \begin{cases} 0.9, & 0 \leq t \leq 50 \\ 2.0, & 51 \leq t \leq 300 \\ -3.0, & 301 \leq t \leq 2000 \end{cases} .$$

For this system, statically admissible inputs satisfy  $|r| \leq 0.95556$ . Thus,  $r = 2$  and  $r = -3$  are both statically inadmissible inputs.

Figures 33-36 show system responses obtained with the MDRSG constraint avoidance control concept. Regardless of the statically inadmissible input, the system converges to a statically admissible equilibrium point, vis.,  $r'(2000) = -0.95556$  and  $x(2000) = \begin{bmatrix} -0.95556 \\ 0 \end{bmatrix}$ . The minimum and maximum control signals generated for this case are  $u = -1.0$  and  $u = 1.0$  respectively. Also,

$$\max_t (r_f(t)_{\min} - r_a(t)_{\max}) = -1.964 \times 10^{-6}, \text{ at } t = 165$$

and

$$\max_t (r_a(t)_{\min} - r_f(t)_{\max}) = -2.496 \times 10^{-6}, \text{ at } t = 1028.$$

Thus, the control signal constraints are not violated, and the admissibility and feasibility constraints are consistent at all times. Finally, the controlled process state coincides with vertices of  $X_s^e$  at both  $t = 165$  and  $t = 1028$ .

A detailed analysis of the MDRSG performance is facilitated by Fig. 37 which shows three expanded views of Fig. 36. Note that for  $0 \leq t \leq 50$  ( $t = 50$  corresponds to 0.5 seconds) no constraints are active. This is expected since the system is initialized at an equilibrium point. Then, at  $t = 51$  the exogenous reference input changes from 0.9 to 2.0, and the modified reference input is limited by the upper feasibility constraint which becomes active at  $t = 51$ . The state trajectory initially encounters the boundary of  $X_s^e$  at  $t = 52$ , at which time the admissibility constraints be-

come active. When the exogenous reference signal changes from 2.0 to -3.0 at  $t = 301$  the lower feasibility constraint becomes active, and  $x(t) \in \text{int}(X_s^\varepsilon)$  for  $301 \leq t \leq 473$ . The state trajectory encounters the boundary of  $X_s^\varepsilon$  again at  $t = 474$ , and remains on the boundary for the remainder of the simulation.

Again, SRSG and DTRG tracking performances are similar to that obtained with the MDRSG for this example. However, the DTRG control signal chatter is again present when the state trajectory is on  $\partial X_s^\varepsilon$ .

The simulation results presented here clearly demonstrate the effectiveness of the proposed MDRSG constraint avoidance strategy. Based on the current controlled process state, the MDRSG generates a modified reference input that ensures the system constraints are avoided, and that the controlled process state is constrained to the invariant set  $X_s^\varepsilon$ . Moreover, a stable closed-loop system response is obtained regardless of the exogenous reference input. Due to the lightly damped nature of the controlled process, the SRSG control concept does not improve tracking performance substantially.

#### 4.4.2 Example Two

This example is included to demonstrate the potential improvement in tracking performance that can be realized with the SRSG control concept. The open-loop plant of eq. (65) is combined with a different state feedback matrix to obtain an overdamped controlled process. The new state feedback matrix is

$$k_x = [ -33 \quad -10.5 ]$$

To obtain perfect tracking  $k_r$  is chosen so that

$$k_r = -\frac{1}{CA_{cl}^{-1}B} = 32$$

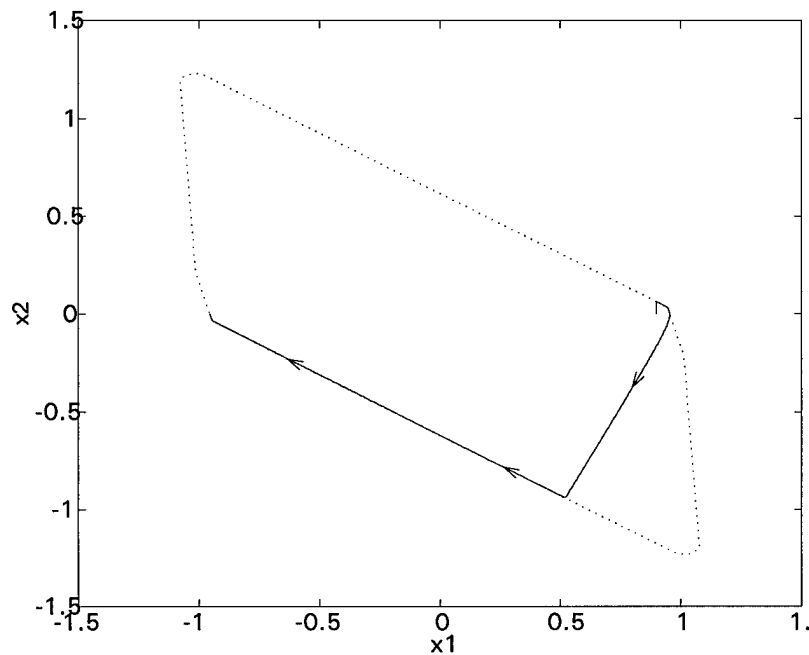


Figure 33. MDRSG state trajectory for Example 1b.

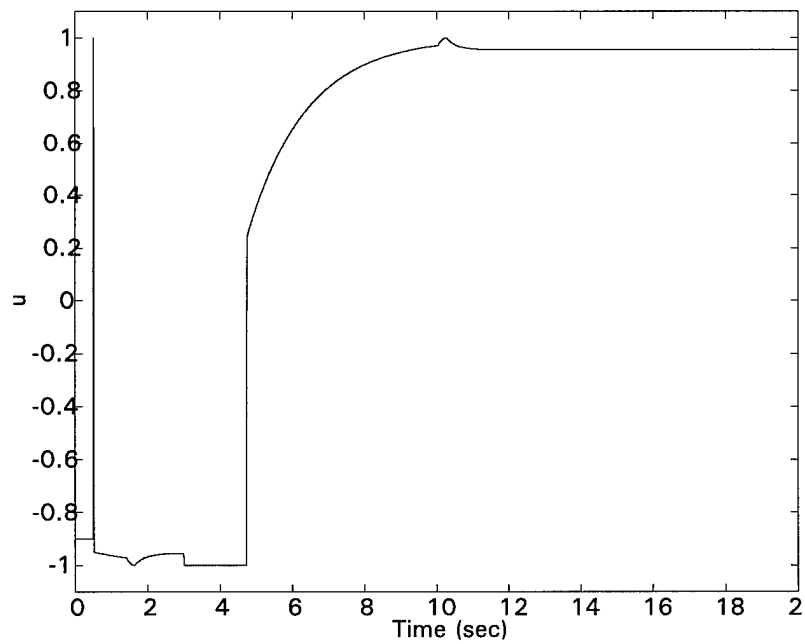


Figure 34. MDRSG control signal response, statically inadmissible input for Example 1b.

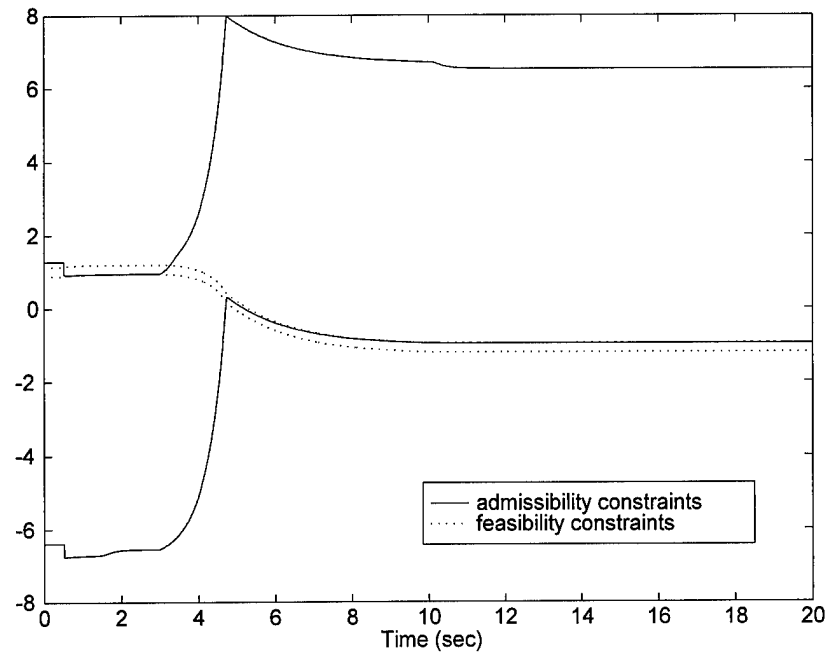


Figure 35. MDRSG reference signal constraints, statically inadmissible input

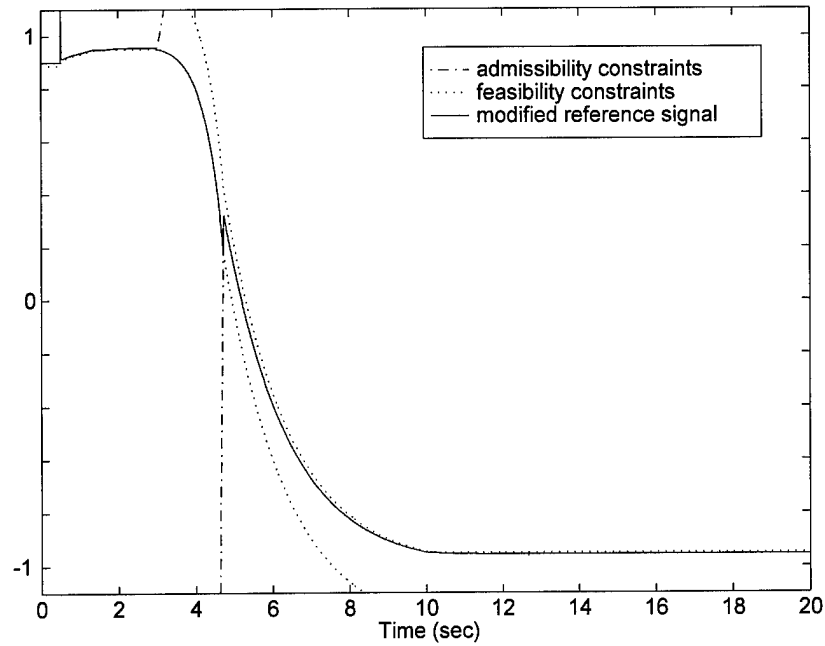


Figure 36. Modified reference signal response for Example 1b.

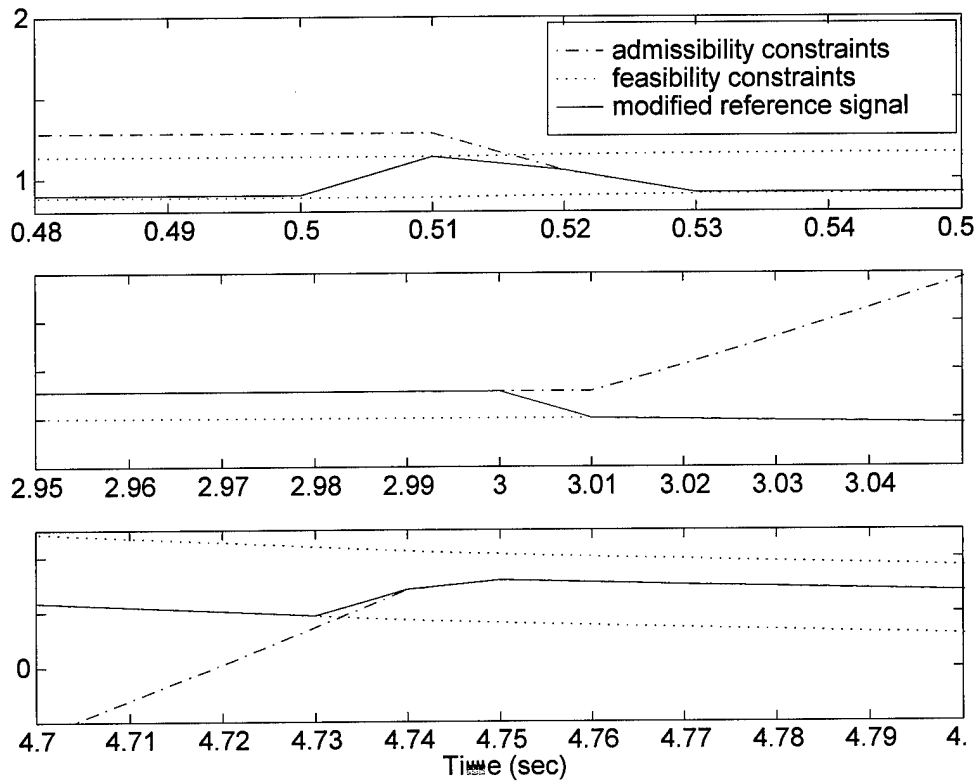


Figure 37. Expanded views of modified reference signal response for Example 1b.



With these values of  $k_x$  and  $k_r$  the discrete time controlled process dynamics and input matrices are

$$A_{cl} = \begin{bmatrix} 9.967893 \times 10^{-1} & 8.9970 \times 10^{-3} \\ -6.432321 \times 10^{-1} & 7.990903 \times 10^{-1} \end{bmatrix} \text{ and } B_{cl} = \begin{bmatrix} 3.2107 \times 10^{-3} \\ 6.432321 \times 10^{-1} \end{bmatrix},$$

The eigenvalues of  $A_{cl}$  are  $\lambda_1 = 9.610593 \times 10^{-1}$  and  $\lambda_2 = 8.348202 \times 10^{-1}$ . The unconstrained system response to a unit step input is shown in Fig. 38.

As before, the controlled process constraints are given by

$$-1 \leq u(t) = k_x x(t) + k_r r'(t) \leq 1 \quad (88)$$

In this case  $O_\infty^\varepsilon$ , with  $\varepsilon = 0.05$ , is characterized with 60 linear inequalities, and the Linear Program (54) generates 14 unique vertices. Thus,  $X_s^\varepsilon$  is characterized with 14 linear inequality constraints, Fig. 39. Figure 40 shows a comparison of the sets,  $X_s^\varepsilon$ , obtained for Examples 1 and 2.

Figure 41 is a comparison of the DTRG, MDRSG, and SRSG responses to a 0.1 amplitude step input. As expected, the DTRG and MDRSG tracking performances are very similar. However, the SRSG control concept provides a substantial improvement in tracking performance. As shown in Fig. 42, the SRSG drives the control signal to its limits, similar to a Bang - Bang controller, whenever  $r - y > \tau$ . In this case  $\tau = 0.065$ . In the absence of constraint avoidance measures saturation does occur, viz., a peak control signal command of 3.2 is generated, but the constrained system response is stable in this case. In fact, enforcing hard limits on the control signal results in a response that is similar to that of the MDRSG system. This is because the state trajectory does not encounter  $\partial(X_s^\varepsilon)$  in this case.

An excellent example of the SRSG control concept's potential for improving tracking performance is shown in Fig. 43. Here, the tracking performance of the unconstrained LTI controlled process is compared to that of the DTRG and SRSG systems for a 0.1 amplitude step input. As expected the DTRG system response is slightly slower. However, the SRSG system response is actually much faster than that of the LTI system. This is inspite of the fact that the SRSG system only

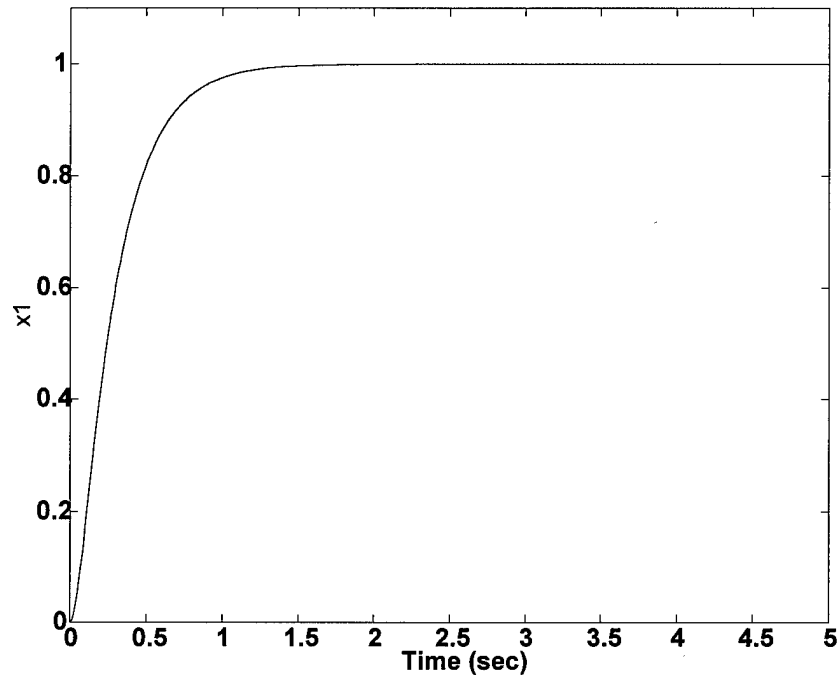


Figure 38. Unconstrained system response to a unit step input for Example 2.

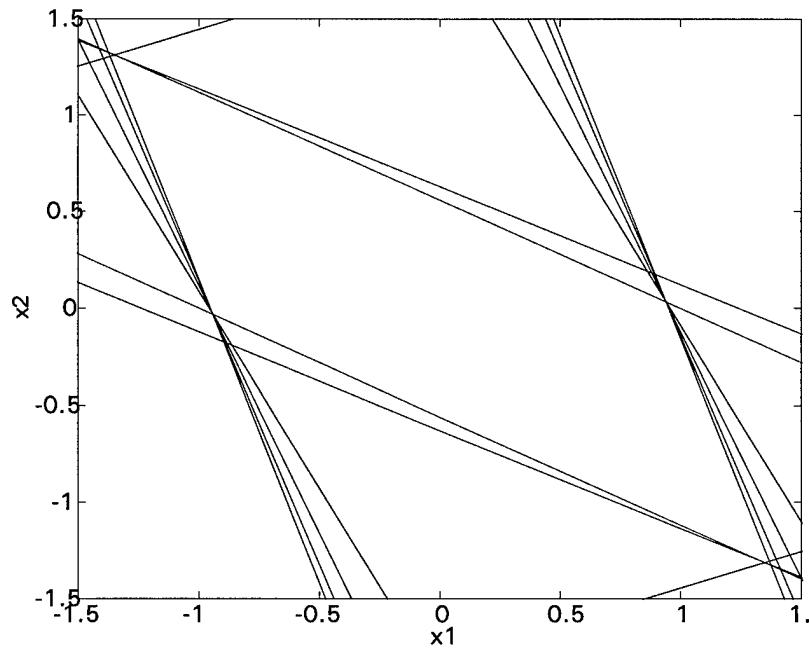


Figure 39. Supporting hyperplanes of  $X_s^\epsilon$  for Example 2.

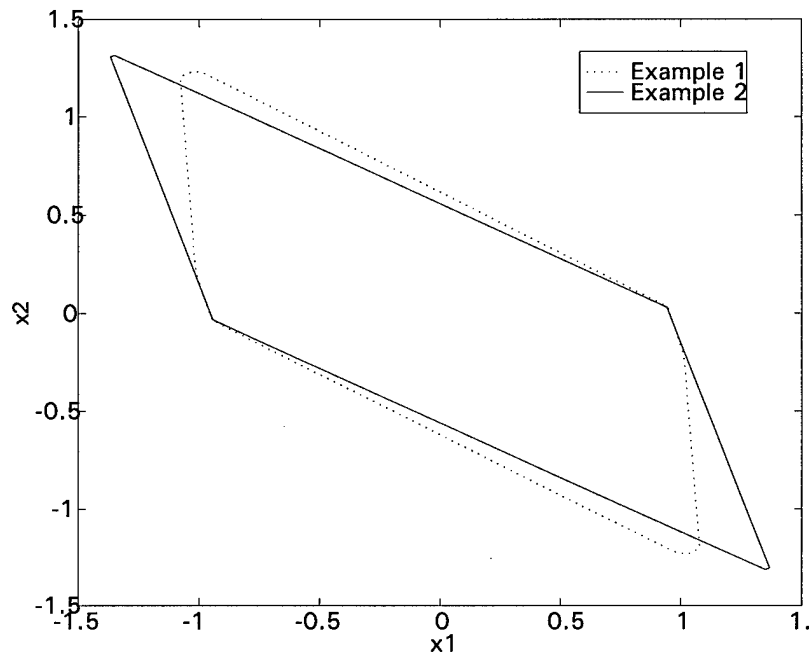


Figure 40. Comparison of  $X_s^\epsilon$  obtained for Examples 1 and 2.

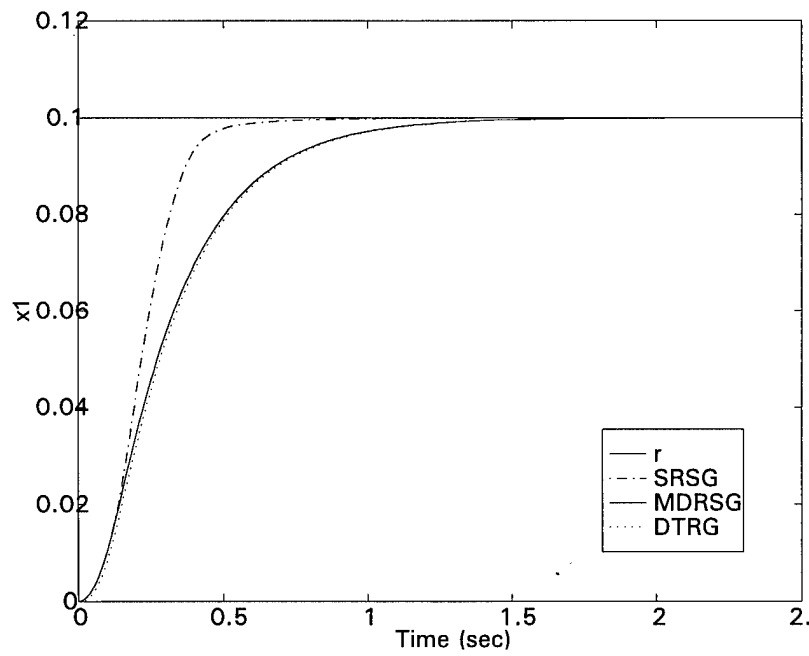


Figure 41. Small signal tracking performance comparison for Example 2.

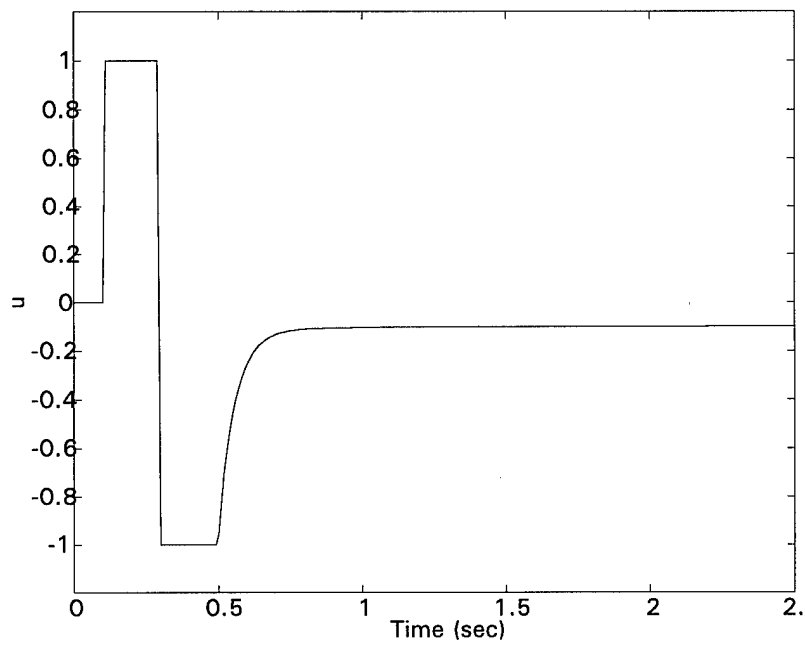


Figure 42. SRS control signal response to a 0.1 amplitude step input for Example 2.

generates allowable control signal increments, whereas the LTI system generates a peak control signal magnitude that is more than three times the allowable limit in this case.

An example of tracking performance during a slewing maneuver is presented next. The initial condition and exogenous reference signal are

$$x(0) = [0.3, 0]^T$$

and

$$r(t) = \begin{cases} 0.3, & 0 \leq t \leq 10 \\ -0.3, & 11 \leq t \leq 500 \end{cases}$$

Figure 44 shows that without constraint avoidance measures, the constrained system response is unstable. Tracking performances of the DTRG, MDRSG, and SRSG systems are shown in Fig. 45. The SRSG control concept clearly outperforms the DTRG and MDRSG control concepts. However, the improvement is not as great as that obtained in the earlier case, Fig. 41. The SRSG control signal response is shown in Fig. 46. From Fig. 46 it is clear that the state trajectory coincides with the boundary of  $X_s^\varepsilon$  from about 0.5 seconds to about 1.25 seconds. As noted earlier, any time the state trajectory encounters  $\partial X_s^\varepsilon$  the admissibility constraints are active, and SRSG tracking performance improvements are limited. Hence, the greatest improvements in tracking performance are realized with the SRSG control concept in situations where  $\partial X_s^\varepsilon$  is not encountered. This includes small signal operation and small to moderate slewing maneuvers.

#### 4.4.3 Example Three

This example is included to demonstrate the trade-offs that exist between system performance and the size of  $X_s^\varepsilon$ . In this case the open-loop plant of eq. (65) is again combined with a different state feedback matrix to obtain an overdamped controlled process that is substantially slower than that of Example 2. The state feedback matrix is

$$k_x = [ -5 \quad -3.5 ] .$$

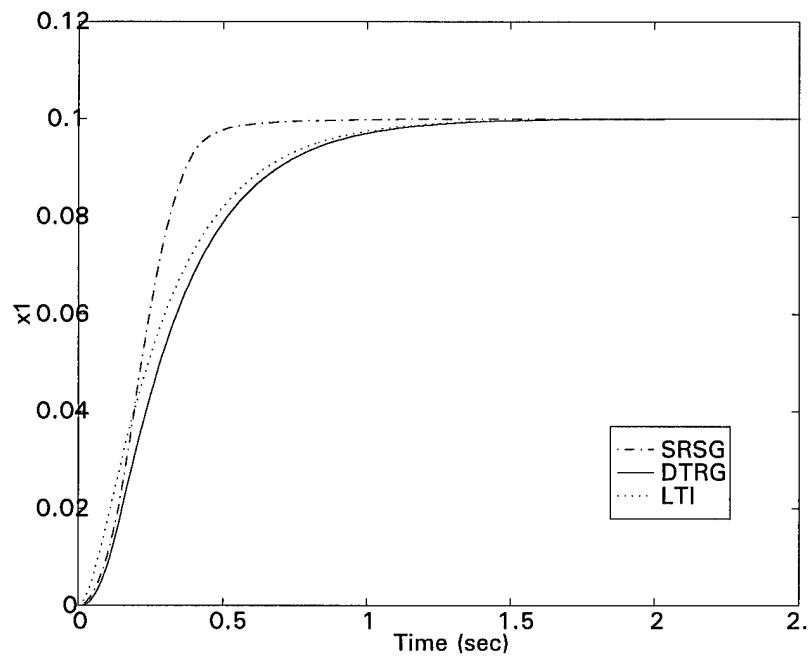


Figure 43. Comparison of SRS, DTRG, and LTI system tracking performances for Example 2.

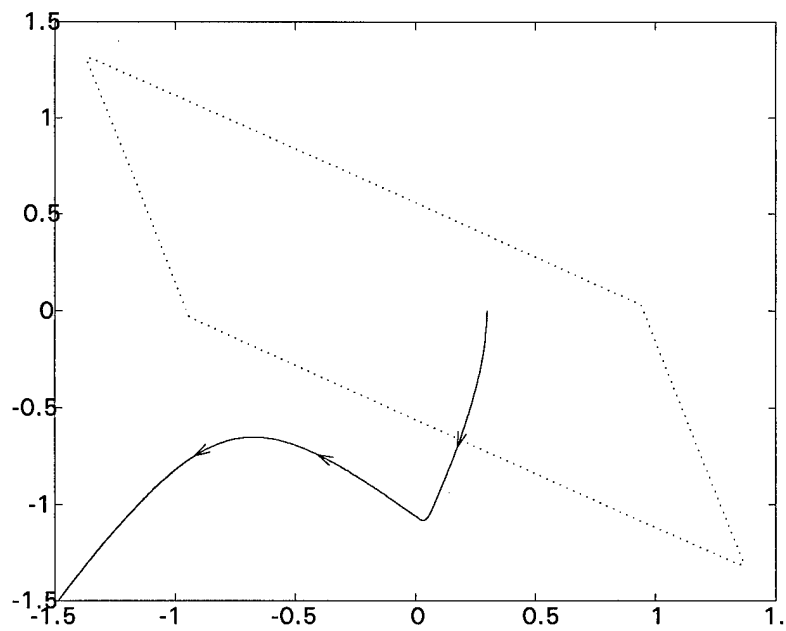


Figure 44. System response to a slewing maneuver without constraint avoidance for Example 2.

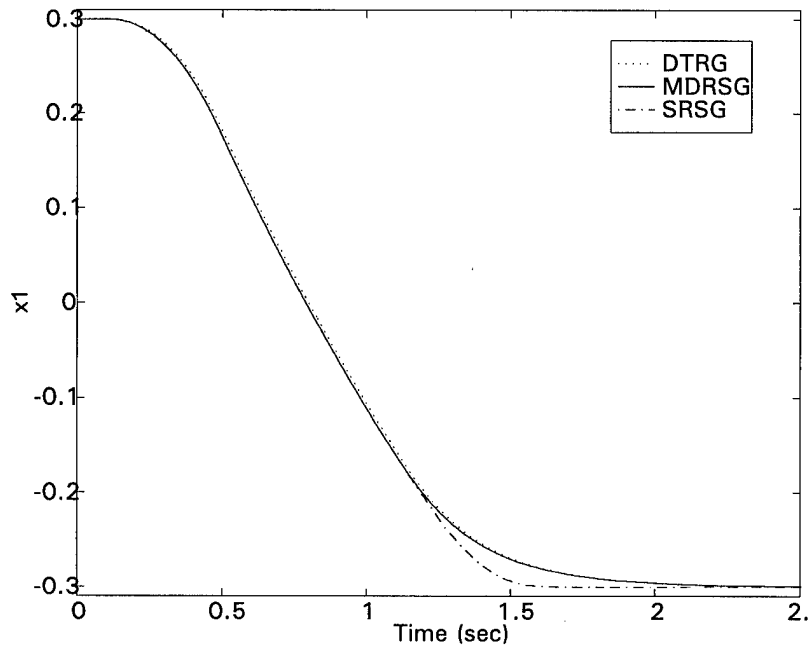


Figure 45. Comparison of DTRG, MDRSG, and SRSG responses to a slewing maneuver for Example 2.

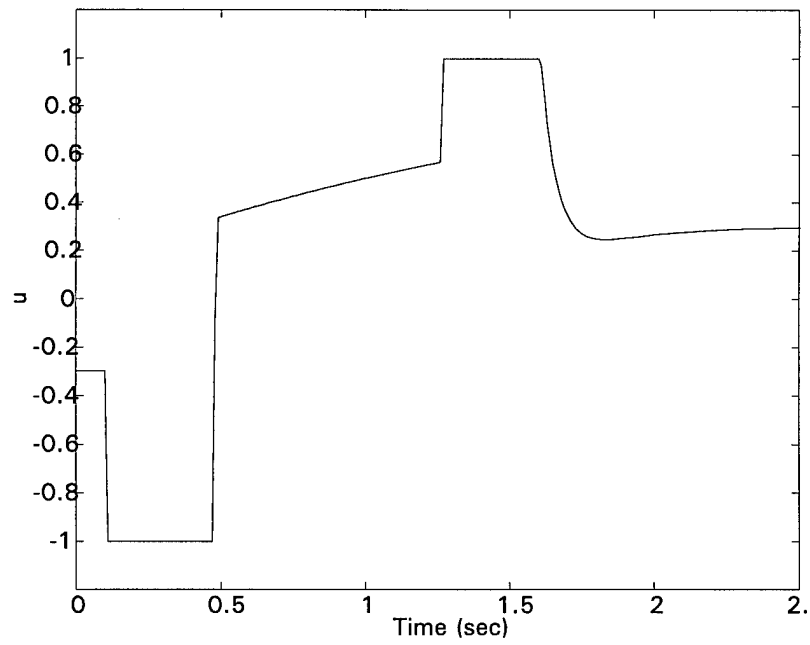


Figure 46. SRSG control signal response to a slewing maneuver for Example 2.

$k_r$  is again chosen to obtain perfect tracking, viz.,

$$k_r = -\frac{1}{CA_{cl}^{-1}B} = 4.$$

With these values of  $k_x$  and  $k_r$  the discrete time controlled process dynamics and input matrices are

$$A_{cl} = \begin{bmatrix} 9.995987 \times 10^{-1} & 9.6993 \times 10^{-3} \\ -8.04040 \times 10^{-2} & 9.397973 \times 10^{-1} \end{bmatrix} \text{ and } B_{cl} = \begin{bmatrix} 4.0134 \times 10^{-4} \\ 8.04040 \times 10^{-2} \end{bmatrix}.$$

The eigenvalues of  $A_{cl}$  are  $\lambda_1 = 9.803837 \times 10^{-1}$  and  $\lambda_2 = 9.590122 \times 10^{-1}$ . The system constraints are again given by eq. (88). Figure 47 compares the response of the unconstrained system to that of the controlled process of Example 2 for a unit step input. As Fig. 47 shows, this system is much slower than the system of Example 2.

In this case  $O_\infty^\varepsilon$ , with  $\varepsilon = 0.05$ , is characterized with 198 linear inequalities, and the Linear Program (54) generates 70 unique vertices. Thus,  $X_s^\varepsilon$  is characterized with 70 linear inequality constraints, Fig. 48. Figure 49 shows a comparison of the sets,  $X_s^\varepsilon$ , obtained for Examples 1, 2, and 3. Notice that all three sets avoid the “sticking” points at  $x = \begin{bmatrix} \pm 1 \\ 0 \end{bmatrix}$ , and are contained in the slab defined by the intersection of the two half-spaces,  $[1 \ 1]x \leq 1$  and  $[-1 \ -1]x \leq 1$ .

As Figs. 47 and 49 show, we have traded tracking performance for a larger statically admissible set. However, from Figs. 50 and 51, it is clear that SRS control concept may be used to improve the tracking performance to the point that the current system performs better than the system of Example 2 with the DTRG control concept. Thus, a larger maximal statically admissible set is obtained without sacrificing tracking performance.

#### 4.5 Constraint Avoidance Using Subsets of $X_s^\varepsilon$

In practice it is usually desirable to base the constraint avoidance strategy on a subset of  $X_s^\varepsilon$  to reduce the on-line computational burden. One method is to use subsets of  $X_s^\varepsilon$  obtained by thinning the set of vertices obtained from the Linear Program (54). The resultant set,  $X_s^\delta \subset X_s^\varepsilon$  is character-



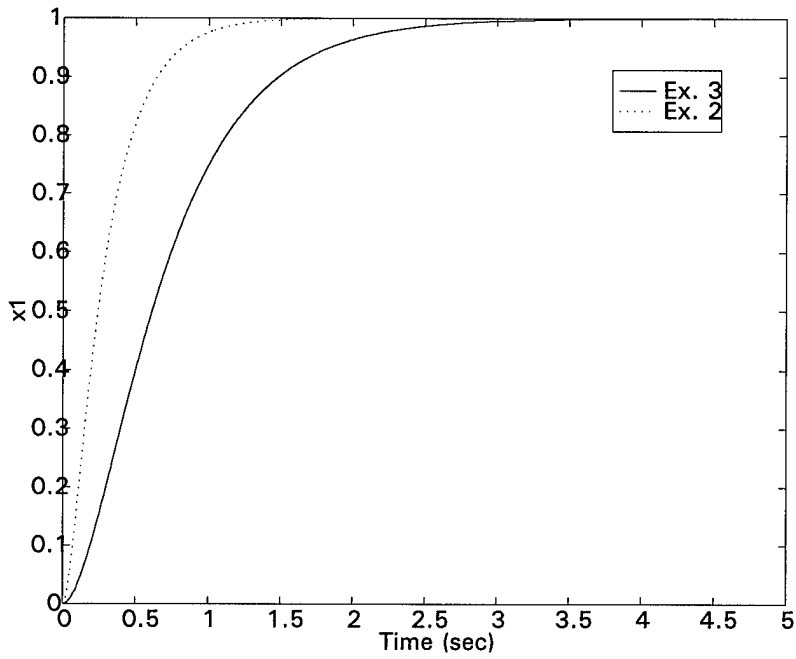


Figure 47. Comparison of the unconstrained system responses to a unit step input for Examples 2 and 3.

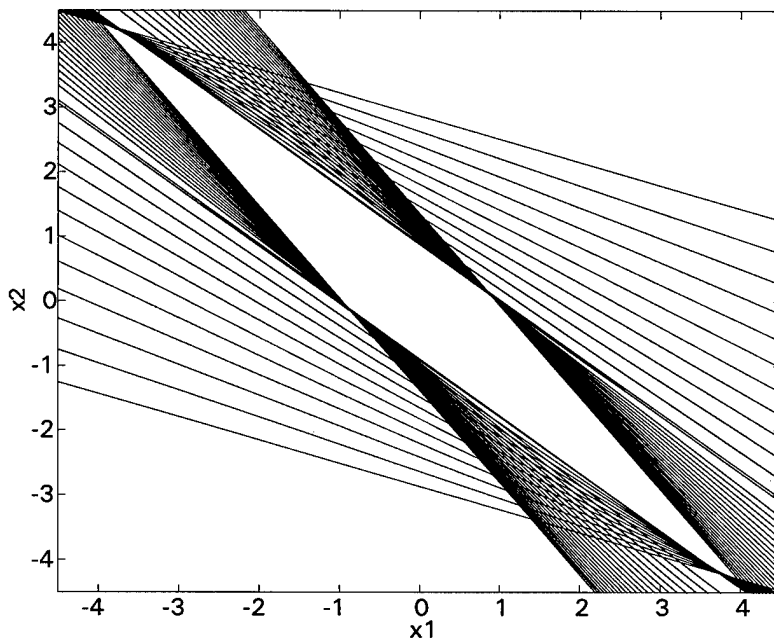


Figure 48. Supporting hyperplanes for  $X_s^\epsilon$  for Example 3.

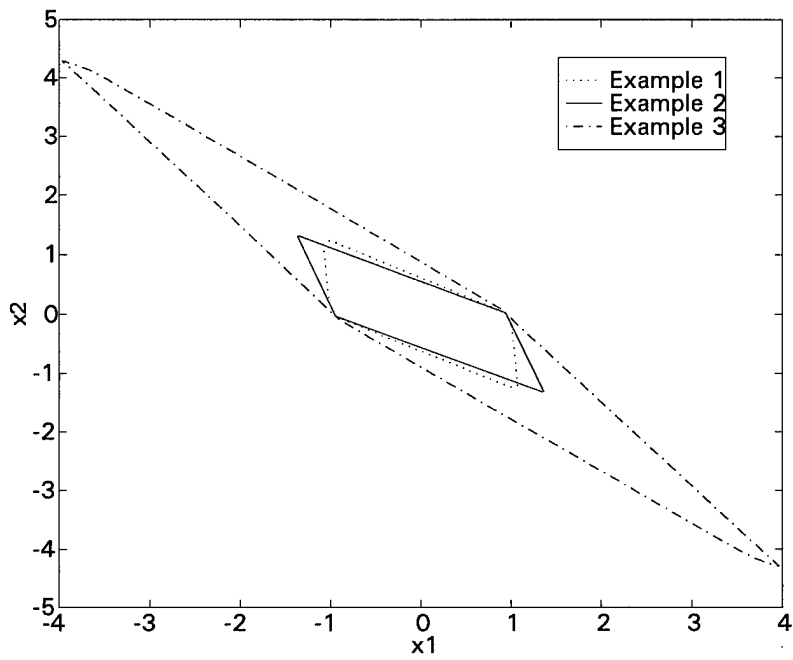


Figure 49. Comparison of  $X_s^e$  for Examples 1, 2, and 3.

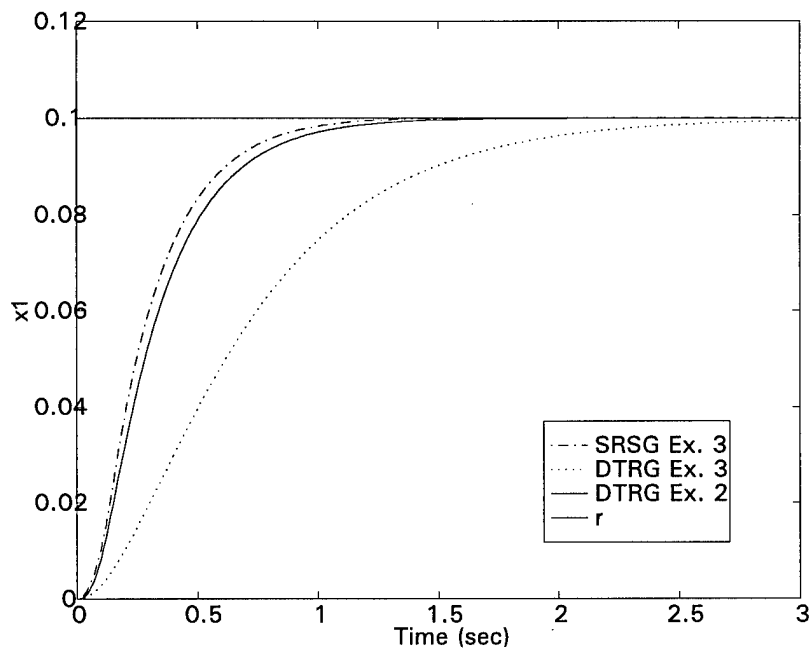


Figure 50. Small signal tracking performance comparison.

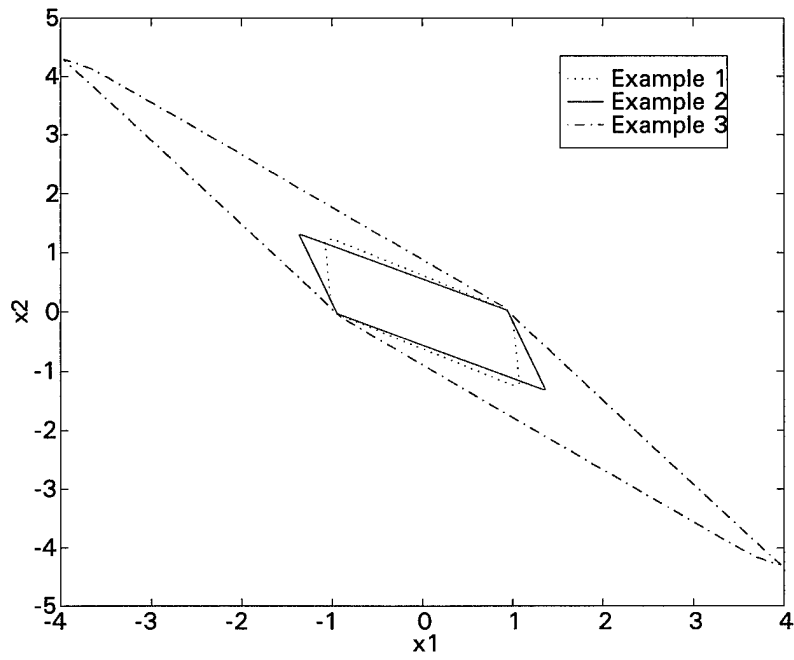


Figure 49. Comparison of  $X_s^\epsilon$  for Examples 1, 2, and 3.

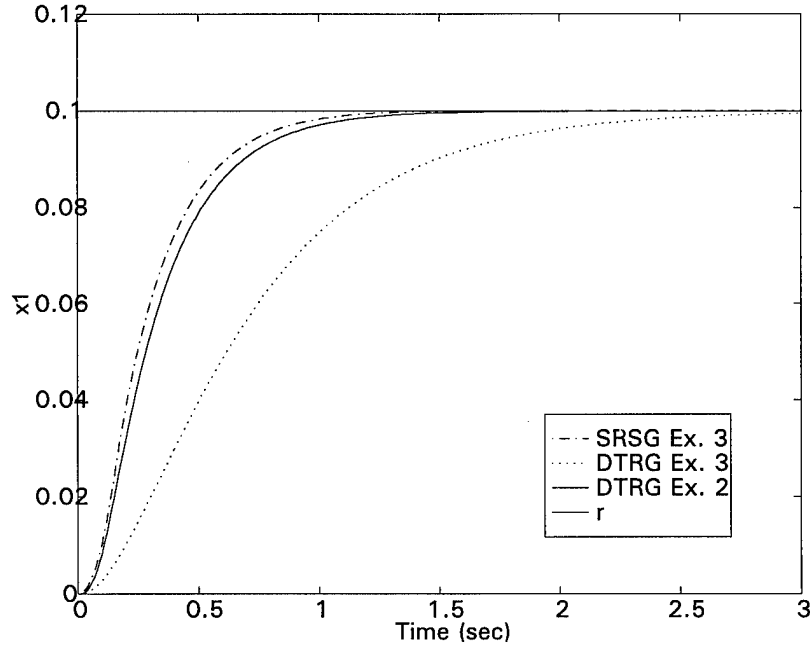


Figure 50. Small signal tracking performance comparison.

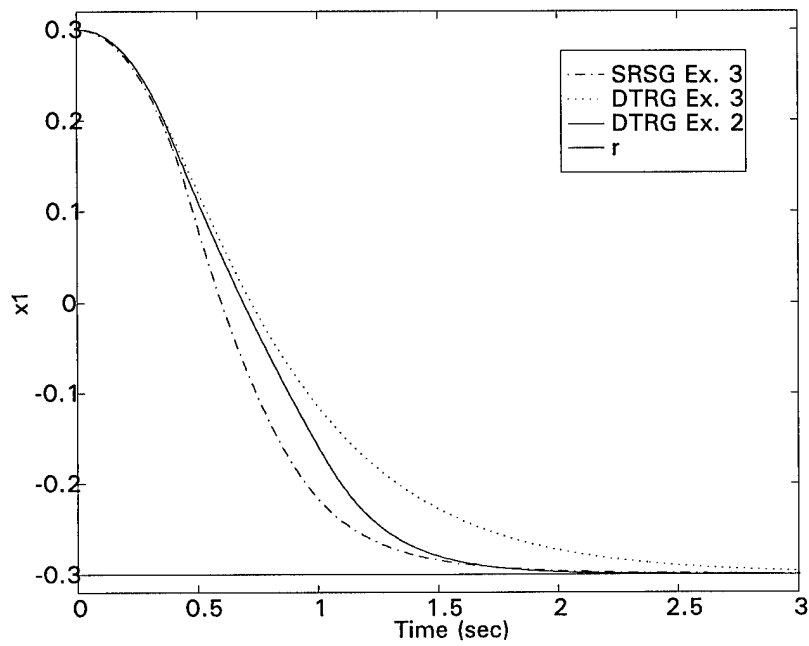


Figure 51. Slewing maneuver tracking performance comparison.

ized by a smaller set of linear inequalities,

$$X_s^\delta = \{x \in \mathfrak{R}^n : \Gamma_\delta x \leq \beta_\delta\}.$$

Thus, fewer inequalities must be evaluated on-line. Of course the volume of the set  $X_s^\delta$  is less than that of the set  $X_s^\varepsilon$ , but generally this method results in a substantial reduction in the number of inequalities with only a small loss of volume, as demonstrated in Section 3.4.1. A potential problem with this method is that unlike the set  $X_s^\varepsilon$ , the set  $X_s^\delta$  is not invariant. However, this problem is easily overcome by taking advantage of the fact that  $X_s^\delta \subset X_s^\varepsilon$ .

Another method to reduce the on-line computational burden is to inscribe an ellipsoid inside either  $X_s^\varepsilon$  or  $X_s^\delta$ . This method leads to even less on-line computational burden, but also results in a greater loss in volume. One benefit of this method is that the resulting set,  $E_s$ , is invariant.

#### 4.5.1 Constraint Avoidance Using Polyhedral Subsets

Here constraint avoidance is based on the set  $X_s^\delta \subset X_s^\varepsilon$ . In this case, given  $x(t) \in X_s^\delta$ ,  $r'(t)$  is admissible with respect to the set  $X_s^\delta$  if  $x(t+1) \in X_s^\delta$ . The feasibility constraint,  $y_c(t) \in Y$ , remains unchanged. Thus, the development of Section 4.3 is modified by replacing  $\Gamma_\varepsilon$  and  $\beta_\varepsilon$  in eqs. (70) and (71) with  $\Gamma_\delta$  and  $\beta_\delta$  respectively. For a given  $x(t) \in X_s^\delta$ ,  $r'(t)$  is now considered allowable if  $x(t+1) \in X_s^\delta$  and  $y_c(t) \in Y$ .

In general  $X_s^\delta$  is not an invariant set for the controlled process, and there exist states,  $x(t) \in X_s^\delta$ , for which no allowable  $r'(t)$  exists. That is, it may transpire that for some  $x(t) \in X_s^\delta$  the admissibility and feasibility constraints are not consistent, and no  $r'(t)$  exists such that both  $x(t+1) \in X_s^\delta$  and  $y_c(t) \in Y$ . This situation can only transpire when  $x(t) \in \partial X_s^\delta$  since the admissibility constraints are not active when  $x(t) \in \text{int}(X_s^\delta)$ .

A solution to this problem lies in the fact that  $X_s^\delta \subset X_s^\varepsilon$ , and  $X_s^\varepsilon$  is an invariant set for the controlled process. That is, for any  $x(t) \in X_s^\delta$  there must exist at least one feasible reference signal

that is admissible with respect to the set  $X_s^\varepsilon$ . Thus, when  $x(t) \in X_s^\delta$ , but no feasible reference signal exists that is admissible with respect to the set  $X_s^\delta$ , one solution is to determine a reference signal that is admissible with respect to the set  $X_s^\varepsilon$ . In this case  $x(t+1) \in X_s^\varepsilon \cap (X_s^\delta)^c$  and  $y_c(t) \in Y$  will transpire. Of course, the desired reference signal must be determined without the benefit of the linear inequalities that characterize the set  $X_s^\varepsilon$ . The following analysis provides the necessary insight to determine an appropriate modified reference signal in this situation.

For a given  $x(t) \in X_s^\delta$  define the sets

$$\begin{aligned} R_a^\delta(t) &= \{r'(t) : r_a^\delta(t)_{\min} \leq r'(t) \leq r_a^\delta(t)_{\max}\}, \\ R_a^\varepsilon(t) &= \{r'(t) : r_a^\varepsilon(t)_{\min} \leq r'(t) \leq r_a^\varepsilon(t)_{\max}\}, \\ R_f(t) &= \{r'(t) : r_f(t)_{\min} \leq r'(t) \leq r_f(t)_{\max}\}. \end{aligned}$$

Where  $R_a^\delta(t)$  denotes the set of reference signals that are admissible with respect to the set  $X_s^\delta$ ,  $R_a^\varepsilon(t)$  the set of reference signals that are admissible with respect to the set  $X_s^\varepsilon$ , and  $R_f(t)$  the set of reference signals that are feasible. For the given state assume  $R_a^\delta(t) \cap R_f(t) = \{\}$ . Since  $x(t) \in X_s^\delta \subset X_s^\varepsilon$ , then  $R_a^\varepsilon(t) \cap R_f(t)$  must be non-empty. Moreover, the sets  $R_a^\delta(t)$ ,  $R_a^\varepsilon(t)$ , and  $R_f(t)$  are all convex, and  $R_a^\delta(t) \subset R_a^\varepsilon(t)$ . Thus, the feasible reference signal that is the minimum distance from the set  $R_a^\delta(t)$  must be contained in  $R_a^\varepsilon(t) \cap R_f(t)$ .

In the single input case, if  $R_a^\delta(t) \cap R_f(t) = \{\}$ , either  $r_a^\delta(t)_{\min} > r_f(t)_{\max}$  or  $r_a^\delta(t)_{\max} < r_f(t)_{\min}$ . If  $r_a^\delta(t)_{\min} > r_f(t)_{\max}$ , then  $r_f(t)_{\max}$  is the feasible reference signal that is the minimum distance from the set  $R_a^\delta(t)$ , and must be contained in  $R_a^\varepsilon(t) \cap R_f(t)$ . In this case choose  $r'(t) = r_f(t)_{\max}$ . Similarly, if  $r_a^\delta(t)_{\max} < r_f(t)_{\min}$ , then choose  $r'(t) = r_f(t)_{\min}$ .

To insure the state trajectory converges to a state  $x(t) \in X_s^\delta$ , the admissibility constraint,  $r'(t) \in R_a^\delta(t)$ , is replaced with the constraint,  $r'(t) \in R_a^\delta(t) \cap R_s^\delta$ , whenever  $x(t)$  is within an epsilon neighborhood of  $\partial X_s^\delta$ . This also avoids chattering and numerical accuracy problems. Here,

$R_s^\delta$  is the set of constant reference signals that are statically admissible with respect to the set  $X_s^\delta$ .

That is,

$$R_s^\delta = \{r_{ss} : \Gamma_\delta x_{ss} \leq \beta_\delta\}$$

where  $x_{ss}$  is the associated equilibrium point for a given statically admissible reference signal,  $r_{ss}$ ,

and is given by

$$x_{ss} = (I - A_{cl})^{-1} B_{cl} r_{ss}.$$

Thus,

$$R_s^\delta = \{r_{ss} : \Gamma_\delta (I - A_{cl})^{-1} B_{cl} r_{ss} \leq \beta_\delta\},$$

and

$$R_a^\delta(t) \cap R_s^\delta = \left\{ r'(t) : \begin{bmatrix} \Gamma_\delta B_{cl} \\ \Gamma_\delta (I - A_{cl})^{-1} B_{cl} \end{bmatrix} r'(t) \leq \begin{bmatrix} \beta_\delta - A_{cl} x(t) \\ \beta_\delta \end{bmatrix} \right\}.$$

Let  $R_{as}^\delta(t) = R_a^\delta(t) \cap R_s^\delta$ , and note that in the single input case an equivalent expression for

the set  $R_s^\delta$  is

$$R_s^\delta = \{r_{ss} : r_{ss\min}^\delta \leq r_{ss} \leq r_{ss\max}^\delta\}$$

where  $r_{ss\min}^\delta$  and  $r_{ss\max}^\delta$  are the minimum and maximum statically admissible reference signals.

Since  $\beta_\delta = [1, \dots, 1]^T$ , and the origin is an interior point of  $R_s^\delta$ , then

$$r_{ss\max}^\delta = \frac{1}{\max(\Gamma_\delta (I - A_{cl})^{-1} B_{cl})}$$

and

$$r_{ss\min}^\delta = \frac{1}{\min(\Gamma_\delta (I - A_{cl})^{-1} B_{cl})}.$$

Thus, the set  $R_{as}^\delta(t)$  is given by

$$R_{as}^\delta(t) = \{r'(t) : r_{as}^\delta(t)_{\min} \leq r'(t) \leq r_{as}^\delta(t)_{\max}\}$$

where,

$$r_{as}^\delta(t)_{\min} = \max(r_a^\delta(t)_{\min}, r_{ss\min}^\delta)$$

and,

$$r_{as}^\delta(t)_{\max} = \min \left( r_a^\delta(t)_{\max}, r_{ss_{\max}}^\delta \right).$$

Now, let  $\varepsilon$  be a predetermined small positive number. Then, when

$$\min (\beta_\delta - \Gamma_\delta x(t)) < \varepsilon, \quad (89)$$

$x(t)$  is within an epsilon neighborhood of  $\partial X_s^\delta$ , and the admissibility constraint becomes  $r'(t) \in R_{as}^\delta(t)$ .

#### 4.5.1.1 Simulation Results

The MDRSG control concept simulations of Section 4.4 are repeated here. Reference signal admissibility constraints are based on a set  $X_s^\delta \subset X_s^\varepsilon$ , which is characterized with 26 linear inequalities. For this set, the minimum and maximum statically admissible reference signals are;  $r_{ss_{\min}}^\delta = -0.95556$  and  $r_{ss_{\max}}^\delta = 0.95556$ . Also,  $\varepsilon = 0.05$  is used in eq. (89) to determine if  $x(t)$  is within an epsilon neighborhood of  $\partial X_s^\delta$ . Results are presented for the same initial conditions and reference signals as those used in Section 4.4.

#### Example 4a.

Figures 52-55 show system responses to a statically admissible reference signal. As before, The initial condition and exogenous reference signal are

$$x(0) = [0.5, 0]^T$$

and

$$r(t) = \begin{cases} 0.5, & 0 \leq t \leq 10 \\ -0.5, & 11 \leq t \leq 500 \end{cases}.$$

With the exception of the reference signal maximum admissibility limit in figure 54, the system responses of Figures 52-55 are nearly identical to those of Figures 26-29. Although the constraint avoidance strategy is based on the set  $X_s^\delta$ , the feasibility and admissibility constraints are consistent at all times for this case, and  $x(t) \in X_s^\delta$  for all time increments. The only noticeable difference is that



the maximum statically admissible reference signal value, 0.95556, is substituted for the maximum admissibility limit when the state trajectory is on the boundary of  $X_s^\delta$ , Figure 54. However, this constraint is never active, and does not affect other system responses.

**Example 4b.**

Figures 56-60 show simulation results obtained for a statically inadmissible input. As in Section 4.4, the initial condition and exogenous reference signal are

$$x(0) = [ 0.9, 0 ]^T$$

and

$$r(t) = \begin{cases} 0.9, & 0 \leq t \leq 50 \\ 2.0, & 51 \leq t \leq 300 \\ -3.0, & 301 \leq t \leq 2000 \end{cases} .$$

Again, the most noticeable difference from the results of Section 4.4 is in the admissibility constraints. Except for a short time interval,  $380 \leq t \leq 430$ , the controlled process state is near or on the boundary of  $X_s^\delta$ , and the admissibility constraints are  $r'(t) \in R_{as}^\delta(t)$ . However, except for the initial 0.5 seconds,  $0 \leq t \leq 50$ , the active admissibility constraint is one of the linear inequalities. Thus, the only substantial difference in system responses is a smaller magnitude control signal spike at  $t = 50$ .

Unlike the previous case, the admissibility and feasibility constraints are inconsistent over two short time intervals, Figure 60. In accordance with the strategy presented here, the modified reference signal is limited by the appropriate feasibility constraint in these situations. Figure 61 is a plot of the state trajectory associated with the bottom plot in Figure 60. The dots in Figure 61 are the vertices of  $X_s^\varepsilon$ , only one of which is contained in  $X_s^\delta$ . As expected, during the time interval that the admissibility and feasibility constraints are inconsistent, the situation  $x(t) \in X_s^\varepsilon \cap (X_s^\delta)^c$  transpires.

The simulation results presented here clearly demonstrate the effectiveness of implementing the proposed constraint avoidance strategy with the linear inequalities that characterize the set  $X_s^\delta$ .

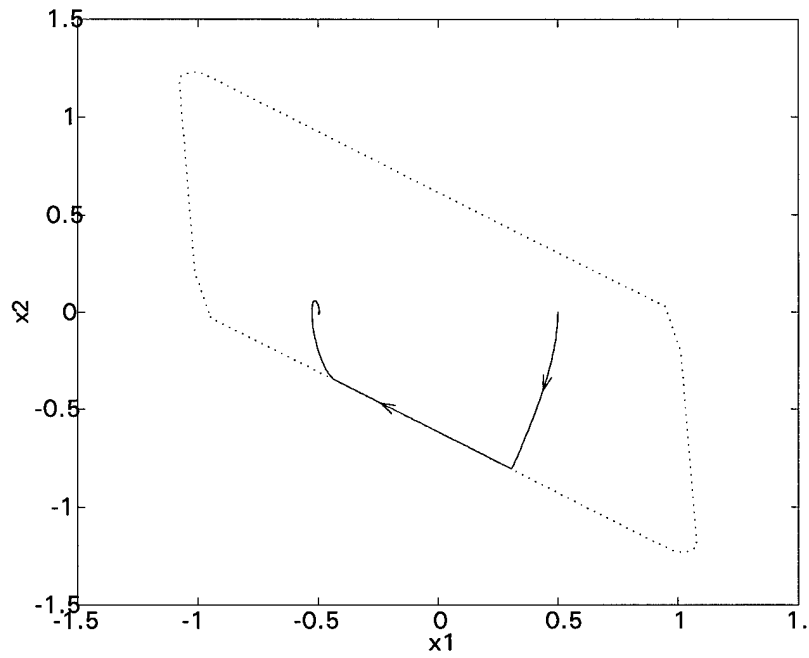


Figure 52. State trajectory for Example 4a.

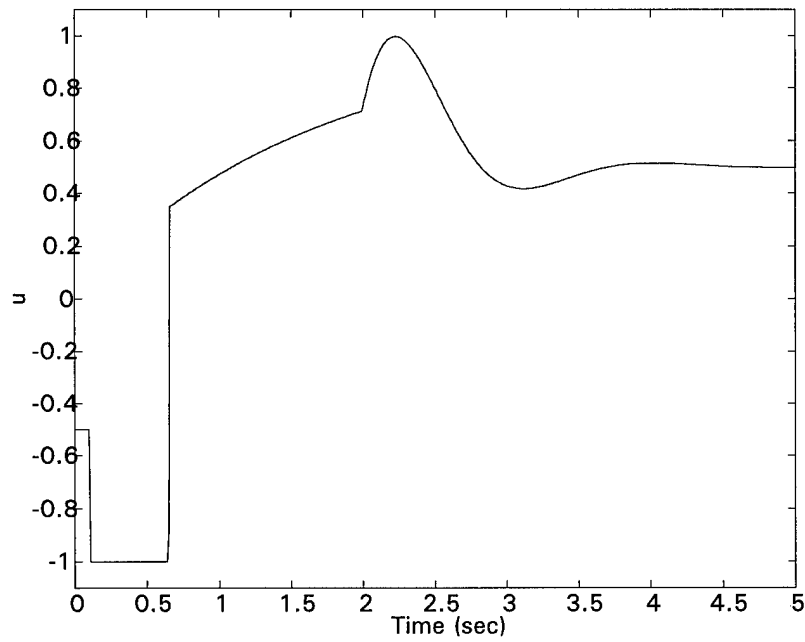


Figure 53. Control signal response for Example 4a.

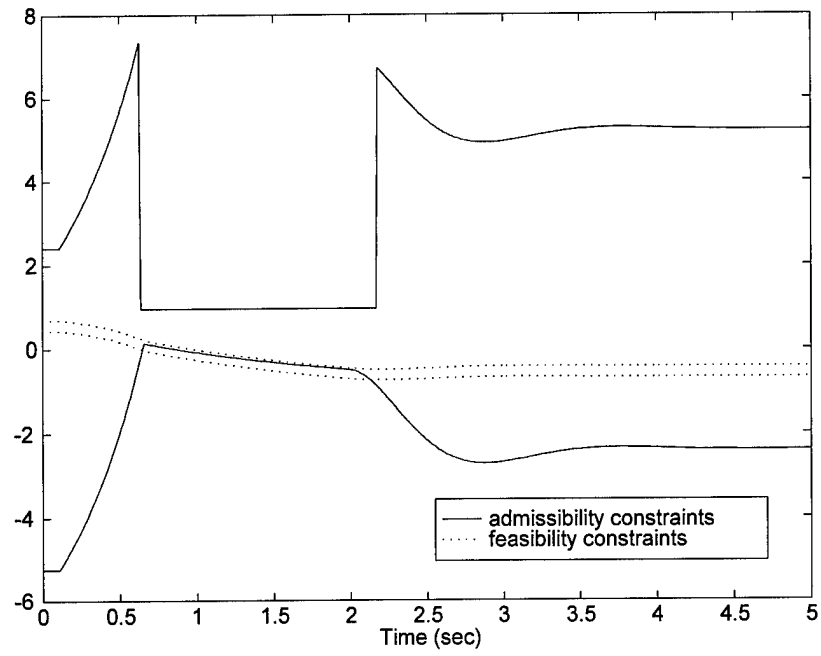


Figure 54. Reference signal constraints for Example 4a.

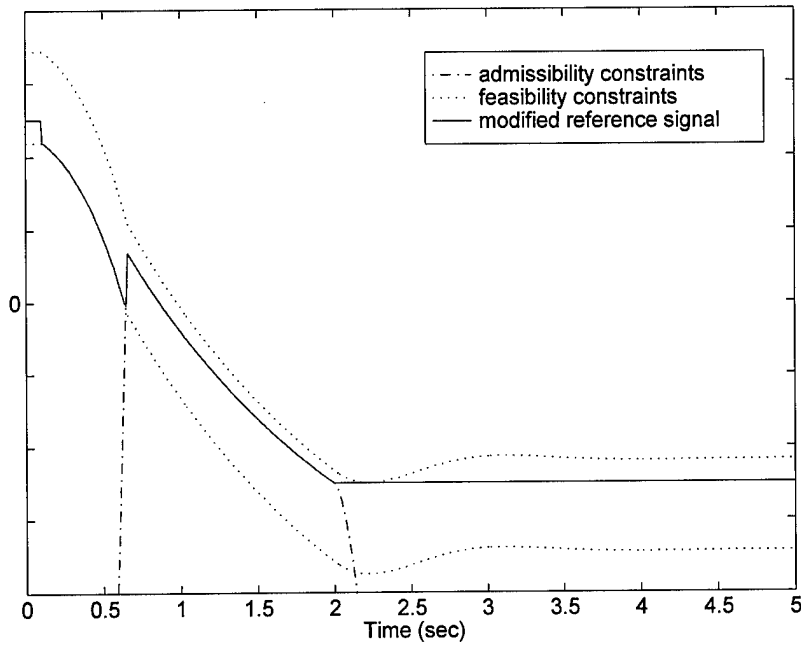


Figure 55. Modified reference signal response for Example 4a.

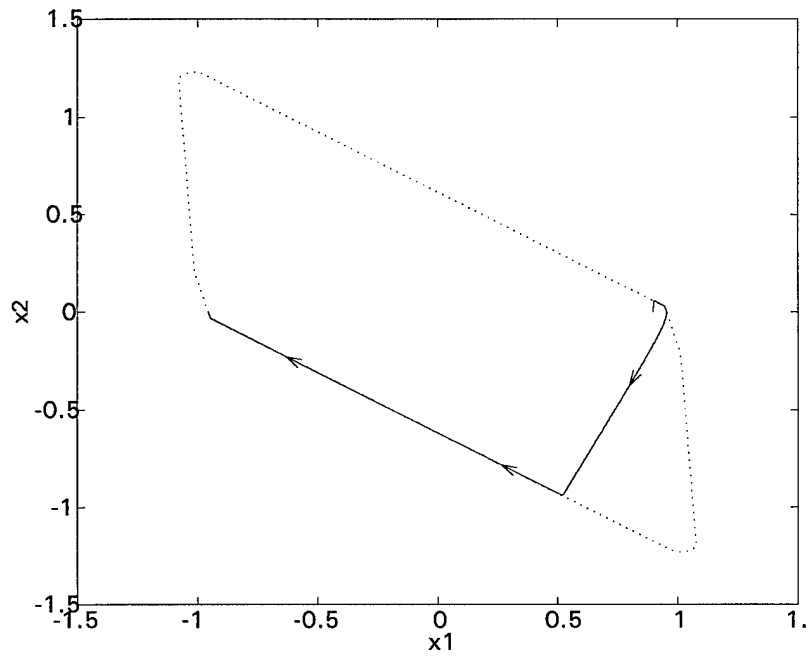


Figure 56. State trajectory for Example 4b.

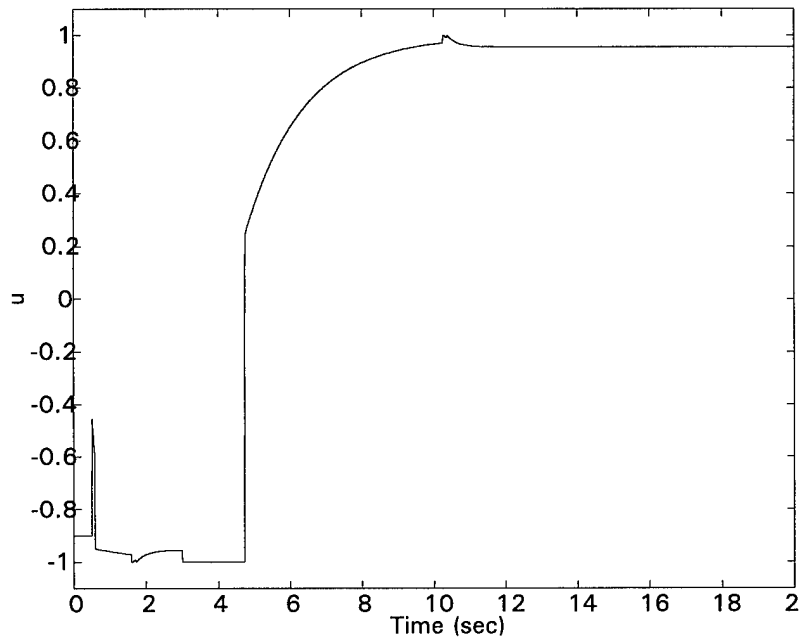


Figure 57. Control signal response for Example 4b.

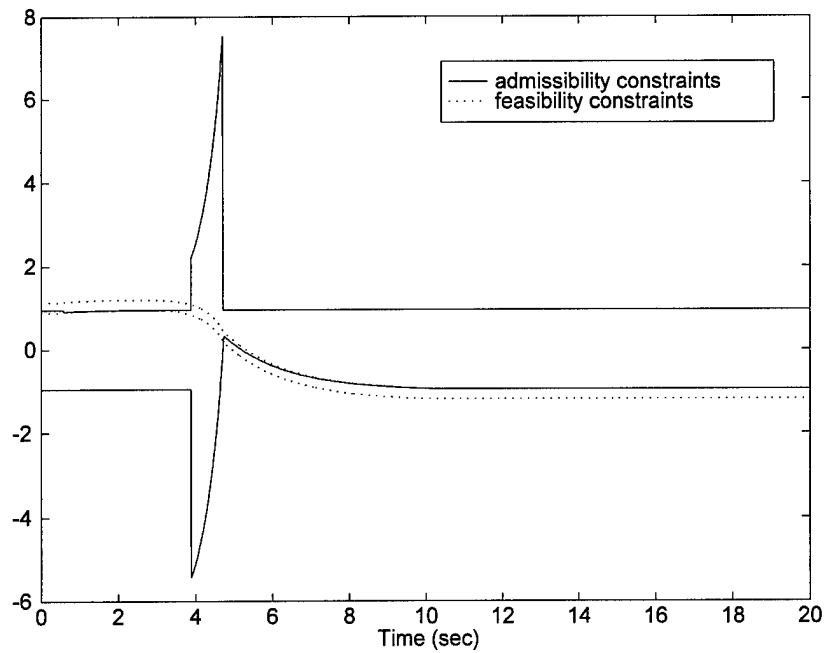


Figure 58. Reference signal constraints for Example 4b.

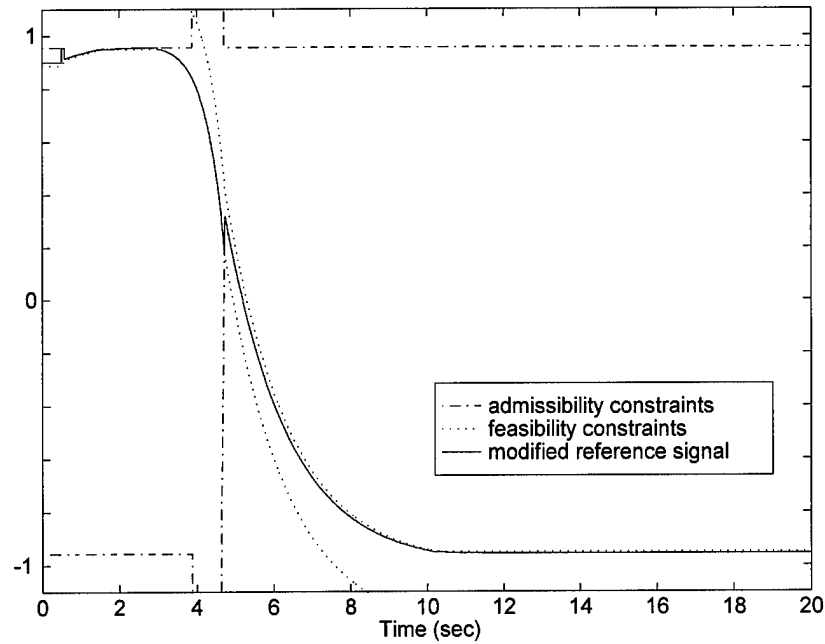


Figure 59. Modified reference signal response for Example 4b.

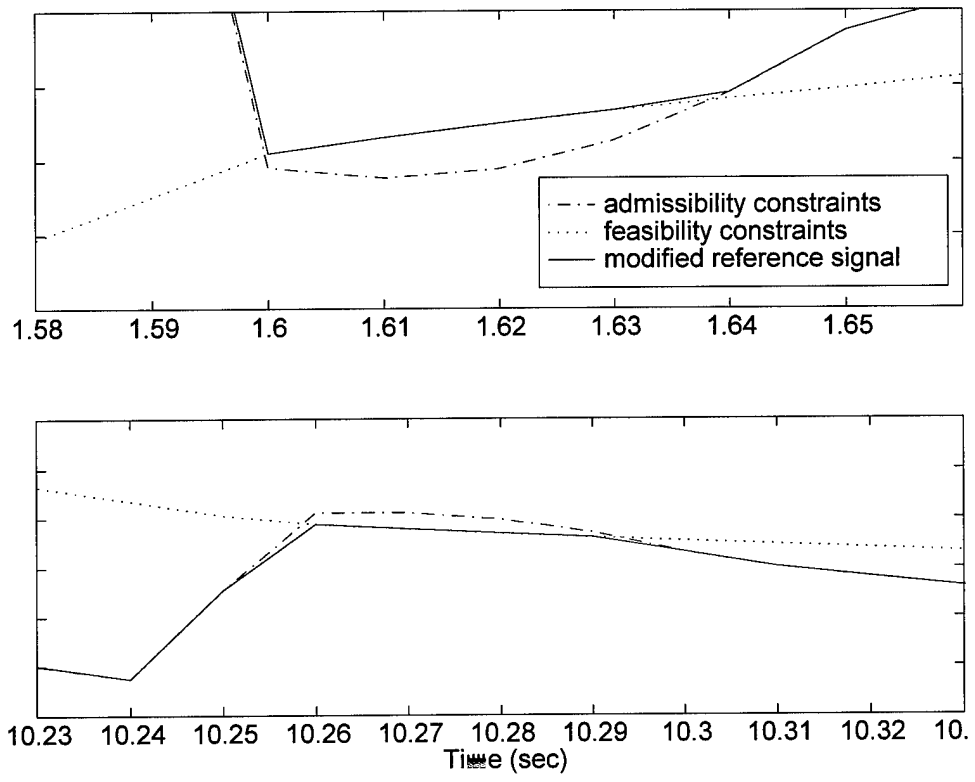


Figure 60. Inconsistent constraints for Example 4b.

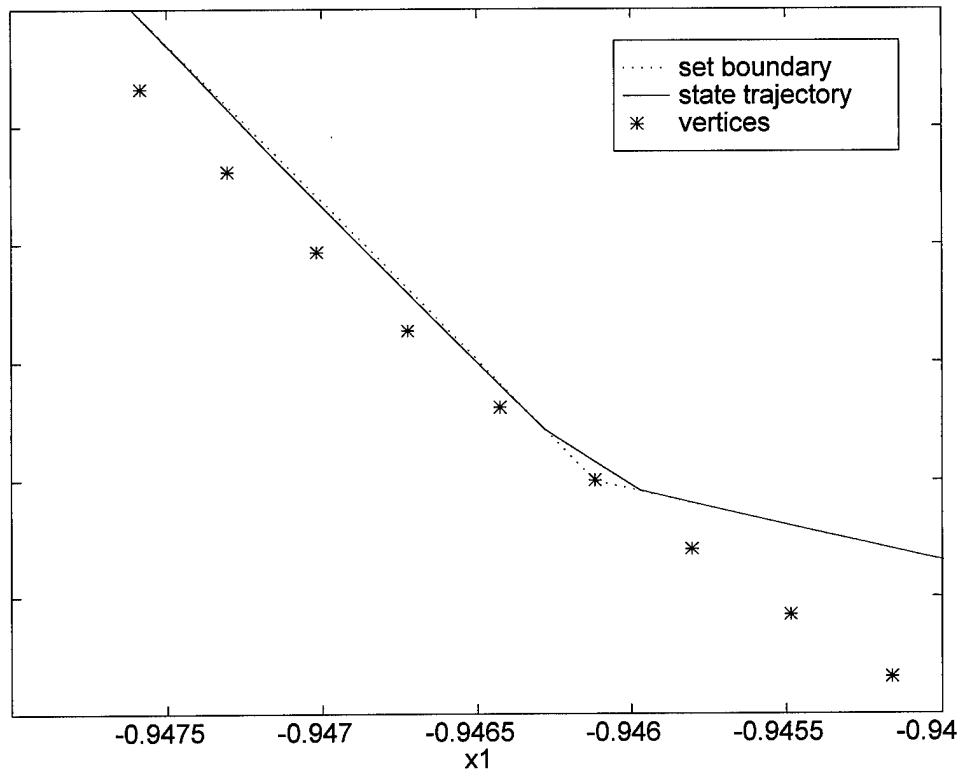


Figure 61. State trajectory resulting from inconsistent constraints for Example 4b.

Regardless of the exogenous reference signal, the controlled process constraints are not violated, and the state is constrained to the invariant set  $X_s^\varepsilon$ . Moreover, a stable closed-loop system response is obtained.

#### 4.5.2 Constraint Avoidance Using Ellipsoidal Subsets

Here constraint avoidance is based on the ellipsoidal set,  $E_s$ , which is inscribed inside either  $X_s^\varepsilon$  or  $X_s^\delta$ . A major advantage of this method is the on-line computational burden is much less than that imposed by polyhedral sets. Ellipsoidal subsets are also invariant with respect to the controlled process. Thus, the problems associated with the set  $X_s^\delta$  are avoided. A major disadvantage of this method is the resulting volume of the invariant ellipsoidal subset may be substantially smaller than that obtained with polyhedral subsets.

The locus of points on the boundary of the set  $E_s$  satisfy

$$(x - m)^T P (x - m) = 1$$

where  $m$  is the ellipsoid center, and  $P$  is a positive definite symmetric matrix. If the constraint set is symmetric about the origin, as is generally the case in problems of concern here, then  $X_s^\varepsilon$  is also symmetric about the origin. In this case the greatest volume will be obtained by an ellipsoid that is also symmetric about the origin. Then the locus of points on the boundary of the set  $E_s$  satisfy

$$x^T P x = 1$$

where the positive definite symmetric matrix,  $P$ , is obtained by solving a convex optimization problem.

Now, given  $x(t) \in E_s$ ,  $r'(t)$  is admissible with respect to the set  $E_s$  if

$$x(t+1)^T P x(t+1) \leq 1$$

Substituting for  $x(t+1)$  gives

$$\left[ A_{cl}x(t) + B_{cl}r'(t) \right]^T P \left[ A_{cl}x(t) + B_{cl}r'(t) \right] \leq 1 \quad (90)$$



Since  $P$  is a positive definite symmetric matrix, the admissibility constraint (90) reduces to a quadratic equation with real roots,

$$ar'(t)^2 + b(t)r'(t) + c(t) \leq 0 \quad (91)$$

where  $a = B_{cl}^T P B_{cl}$ ,  $b(t) = 2B_{cl}^T P A_{cl} x(t)$ , and  $c(t) = x(t)^T A_{cl}^T P A_{cl} x(t) - 1$ . Thus, the set of reference signals that are admissible with respect to the set  $E_s$  is given by

$$R_a^E(t) = \left\{ r'(t) : \lambda(t)_{\min} \leq r'(t) \leq \lambda(t)_{\max} \right\}$$

Where  $\lambda(t)_{\min}$  and  $\lambda(t)_{\max}$  are the minimum and maximum roots of eq. (91) respectively.

#### 4.5.2.1 Simulation Results

##### Example 5.

An example of constraint avoidance using an elliptical subset of  $X_s^e$  is presented here. The initial condition and reference signal are

$$x(0) = [0.5, 0]^T$$

and

$$r(t) = \begin{cases} 0.5, & 0 \leq t \leq 10 \\ -0.5, & 11 \leq t \leq 500 \end{cases}$$

System responses to this initial condition and reference signal are shown in Figures 62-65. As expected, the state trajectory is constrained to the ellipse, and the admissibility and feasibility constraints are consistent for all time increments. Moreover, the control signal magnitude is limited to  $|u| \leq 1$  for all time increments.

## 4.6 Summary

In this chapter specific reference signal governor (RSG) algorithms were developed based on the set  $X_s^e$ . The modified reference signal,  $r'(t)$ , was chosen at each time step so that  $x(t+1) \in X_s^e$  and  $y_c(t) \in Y$ . This allowed the greatest flexibility in selecting  $r'(t)$  at each time step, while also providing a BIBO stable closed-loop system. It was shown that under appropriate conditions

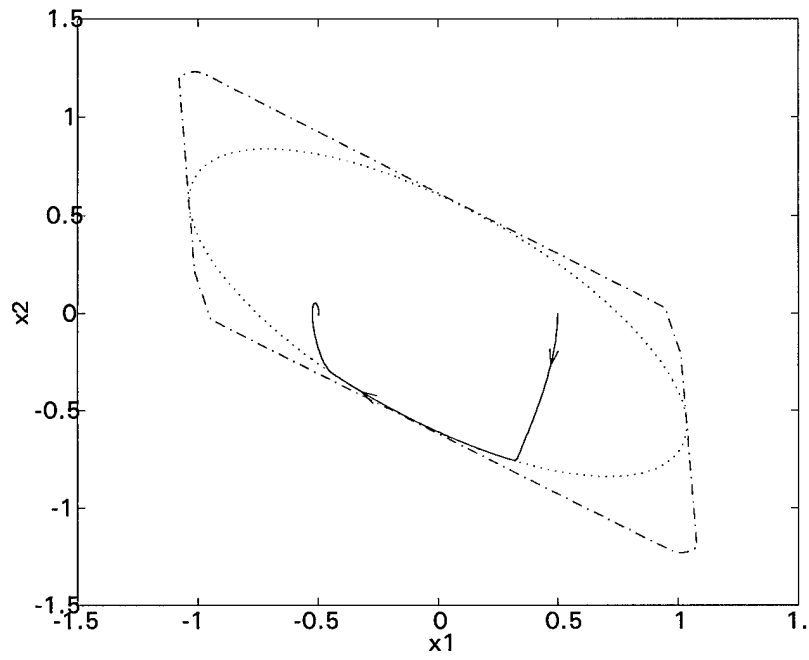


Figure 62. State trajectory for Example 5.

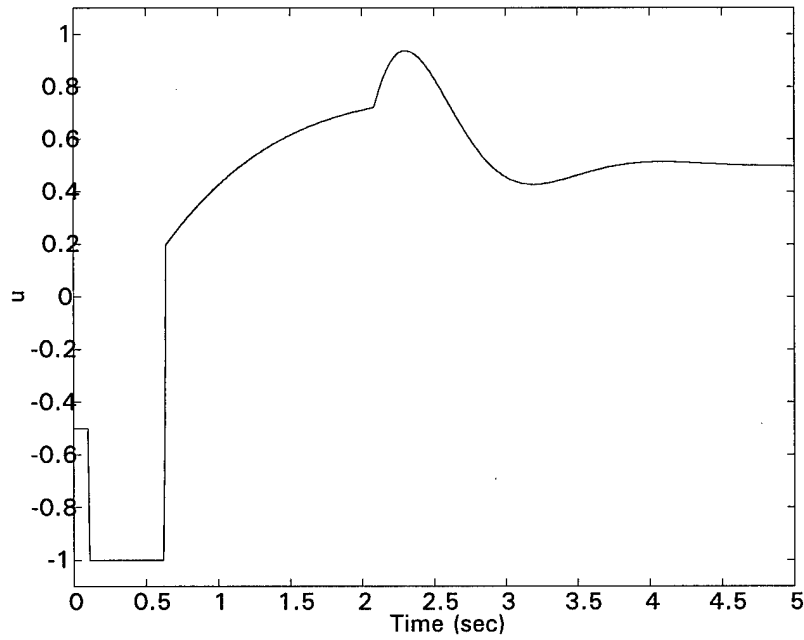


Figure 63. Control signal response for Example 5.

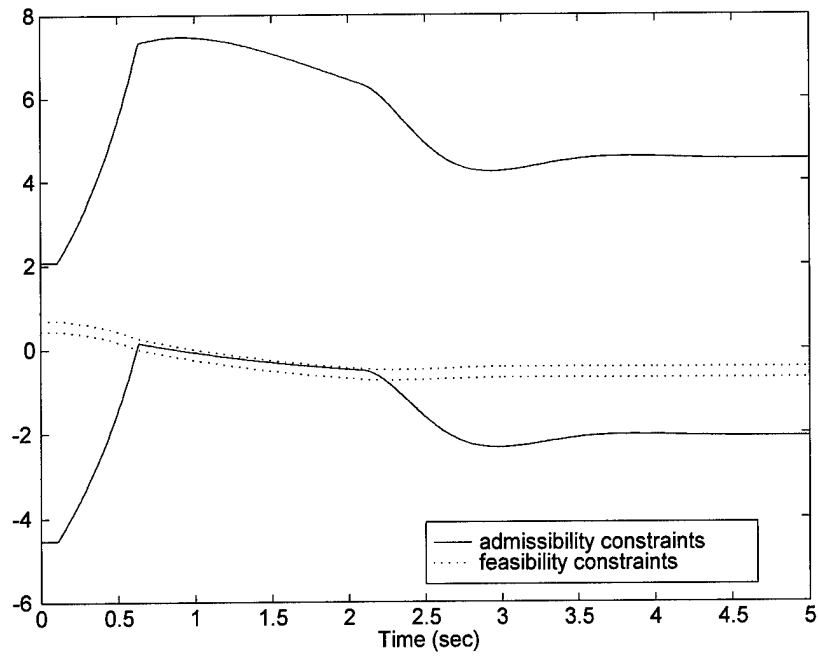


Figure 64. Reference signal constraints for Example 5.

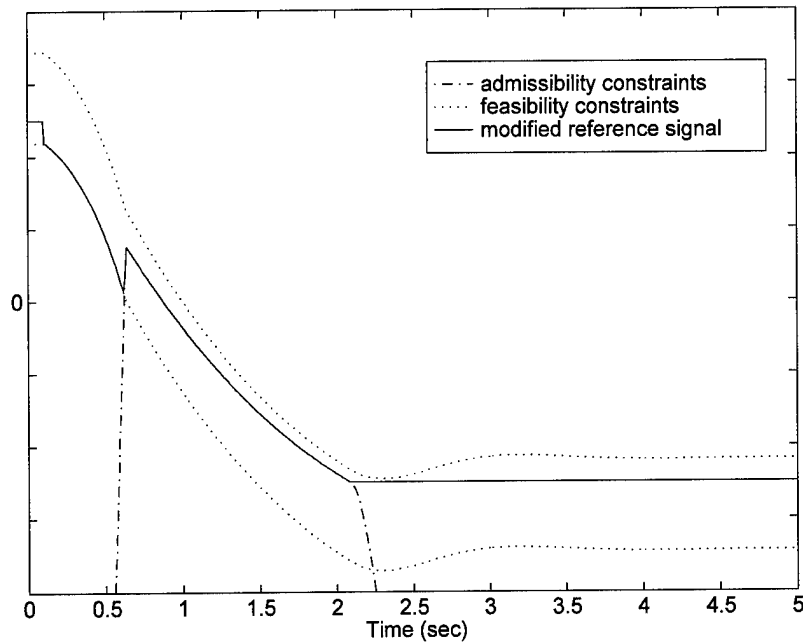


Figure 65. Reference signal response for Example 5.

the SRSG control concept can provide a substantial improvement in tracking performance over that of the MDRSG and DTRG control concepts. Finally, Constraint avoidance methods based on polyhedral and elliptical subsets of the set  $X_s^\varepsilon$  were also presented.

## Chapter 5 - Flight Control Application

### 5.1 Overview

In this chapter a realistic flight control application of the methods developed in Chapters 2, 3, and 4 is presented. The constrained system includes a second-order short period approximation of the longitudinal dynamics of an F-16 aircraft, augmented with a first-order actuator model and an integral control state. The resulting fourth-order system is subject to state and control constraints which arise from actuator displacement and rate constraints.

To reduce the on-line computational burden, reference signal governor algorithms are based on polyhedral subsets of  $X_s^\varepsilon$ . Maximum and minimum allowable steady state conditions are compared for several different polyhedral subsets. This provides one measure to determine if a particular set,  $X_s^{\delta_i} \subset X_s^\varepsilon$ , is adequate.

Validation of the recursive convex hull algorithm for this fourth-order problem is also accomplished. Recursive convex hull algorithm results are compared to those obtained from a brute force algorithm. Additional validation of the recursive algorithm is also accomplished by examining the vertices, subfacets, and facets that are involved in the incorporation of each new vertex.

### 5.2 Aircraft Model and Constraints

A continuous time second-order short period approximation of the F-16 longitudinal dynamics, for a flight condition of 10,000 feet, Mach 0.7 is

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -1.15 & 0.9937 \\ 3.724 & -1.26 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -0.177 \\ -19.5 \end{bmatrix} \delta_e.$$

The tracking task requires pitch rate,  $q$ , to follow an exogenous reference signal,  $r$ . An integral control state is given by  $\dot{z} = r - q$ . The actuator is modeled as a first-order lag with a bandwidth of 20 rad/sec. Augmenting the bare plant model with the actuator model and integral control state

results in

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -1.15 & 0.9937 & -0.177 & 0 \\ 3.724 & -1.26 & -19.5 & 0 \\ 0 & 0 & -20 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_{ec}(t) \\ r(t) \end{bmatrix} \\ y(t) &= [0 \ 1 \ 0 \ 0] x(t) \end{aligned}$$

where  $x = [\alpha \ q \ \delta_e \ z]^T$ . Actuator position and rate constraints are  $\pm 0.44$  rad and  $\pm 1.0$

rad/sec respectively, and may be expressed in terms of output constraints as

$$\begin{bmatrix} -0.44 \\ -1.0 \end{bmatrix} \leq y_c(t) = \begin{bmatrix} \delta_e(t) \\ \dot{\delta}_e(t) \end{bmatrix} = C_c x(t) + D_c \begin{bmatrix} \delta_{ec}(t) \\ r(t) \end{bmatrix} \leq \begin{bmatrix} 0.44 \\ 1.0 \end{bmatrix}$$

where

$$C_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -20 & 0 \end{bmatrix}, \text{ and } D_c = \begin{bmatrix} 0 & 0 \\ 20 & 0 \end{bmatrix}.$$

Using a sampling interval of  $T = 0.01$  seconds, and assuming a zero-order hold on the input yields the following equivalent discrete-time system:

$$x(t+1) = Ax(t) + B \begin{bmatrix} \delta_{ec}(t) \\ r(t) \end{bmatrix}$$

where,

$$A = \begin{bmatrix} 9.887487 \times 10^{-1} & 9.8186 \times 10^{-3} & -2.4949 \times 10^{-3} & 0 \\ 3.67962 \times 10^{-2} & 9.876618 \times 10^{-1} & -1.756338 \times 10^{-1} & 0 \\ 0 & 0 & 8.187308 \times 10^{-1} & 0 \\ -1.8472 \times 10^{-4} & -9.9379 \times 10^{-3} & 9.0937 \times 10^{-4} & 1 \end{bmatrix}$$

and,

$$B = \begin{bmatrix} -2.2624 \times 10^{-4} & 0 \\ -1.81874 \times 10^{-2} & 0 \\ 1.812692 \times 10^{-1} & 0 \\ 6.1686 \times 10^{-5} & 1.00 \times 10^{-2} \end{bmatrix}.$$

The control law is

$$\delta_{ec}(t) = k_x x(t) + k_r r(t)$$

where  $k_x$  and  $k_r$  are chosen to provide satisfactory small signal operation, and are

$$k_x = [0.35090 \ 1.1373 \ -0.9184 \ -6.6107]$$

$$k_r = -0.6.$$

Then, the discrete-time controlled process is

$$\begin{aligned} x(t+1) &= \begin{bmatrix} \alpha(t+1) \\ q(t+1) \\ \delta_e(t+1) \\ z(t+1) \end{bmatrix} = A_{cl}x(t) + B_{cl}r(t) \\ y(t) &= [0 \ 1 \ 0 \ 0]x(t) \end{aligned} \quad (92)$$

where,

$$A_{cl} = \begin{bmatrix} 9.886693 \times 10^{-1} & 9.5613 \times 10^{-3} & -2.2871 \times 10^{-3} & 1.4956 \times 10^{-3} \\ 3.04143 \times 10^{-2} & 9.669773 \times 10^{-1} & -1.589305 \times 10^{-1} & 1.202316 \times 10^{-1} \\ 6.36074 \times 10^{-2} & 2.061658 \times 10^{-1} & 6.522531 \times 10^{-1} & -1.1983166 \\ -1.6307 \times 10^{-4} & -9.8677 \times 10^{-3} & 8.5272 \times 10^{-4} & 9.995922 \times 10^{-1} \end{bmatrix}$$

and,

$$B_{cl} = \begin{bmatrix} 1.3575 \times 10^{-4} \\ 1.091245 \times 10^{-2} \\ -1.087615 \times 10^{-1} \\ 9.9630 \times 10^{-3} \end{bmatrix}.$$

In terms of the discrete time system, the actuator displacement and rate constraints are modeled

as

$$\begin{bmatrix} -0.44 \\ -0.01 \end{bmatrix} \leq \begin{bmatrix} \delta(t) \\ \delta(t+1) - \delta(t) \end{bmatrix} \leq \begin{bmatrix} 0.44 \\ 0.01 \end{bmatrix}$$

or,

$$\begin{bmatrix} -0.44 \\ -0.01 \end{bmatrix} \leq C_{ccl}x(t) + D_{ccl}r(t) \leq \begin{bmatrix} 0.44 \\ 0.01 \end{bmatrix} \quad (93)$$

where,

$$\begin{aligned} C_{ccl} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ A_{cl_{3,1}} & A_{cl_{3,2}} & A_{cl_{3,3}} - 1 & A_{cl_{3,4}} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 6.36074 \times 10^{-2} & 2.061658 \times 10^{-1} & -3.477469 \times 10^{-1} & -1.1983166 \end{bmatrix} \end{aligned}$$

and,

$$D_{ccl} = \begin{bmatrix} 0 \\ B_{cl_3} \end{bmatrix} = \begin{bmatrix} 0 \\ -1.087615 \times 10^{-1} \end{bmatrix}.$$

Equations (92) and (93) define the controlled process and its constraints. Of course, since a dual loop controller architecture is employed, the exogenous input,  $r$ , is replaced with the modified reference signal,  $r'$ .

### 5.3 Linear Inequality Constraints

First, the set,  $O_\infty^\varepsilon \subset \mathfrak{R}^5$ , is characterized using the methods of Chapter 2. This requires that we augment the controlled process (92) with the first-order lag filter of eq. (29). This results in the augmented state vector

$$x_g = \begin{bmatrix} r' \\ x \end{bmatrix} \in \mathfrak{R}^5,$$

with the augmented system dynamics and output constraints given by

$$x_g(t+1) = A_g x_g(t) + B_g \lambda(r(t), x_g(t)) (r(t) - [1 \ 0_{4 \times 1}] x_g(t))$$

$$y_c(t) = C_g x_g(t) \in Y \subset \mathfrak{R}^2$$

where,

$$A_g = \begin{bmatrix} 1 & 0_{1 \times 4} \\ B_{cl} & A_{cl} \end{bmatrix}, B_g = \begin{bmatrix} 1 \\ 0_{4 \times 1} \end{bmatrix}, C_g = [D_{c_{cl}} \ C_{c_{cl}}],$$

and the output constraint set,  $Y$ , is given by

$$Y = \{y_c \in \mathfrak{R}^2 : f_i(y_c) \leq 0, \quad i = 1, 2, 3, 4\}$$

Where,

$$f_1(y_c) = y_c(1) - 0.44$$

$$f_2(y_c) = y_c(2) - 0.01$$

$$f_3(y_c) = -y_c(1) - 0.44$$

$$f_4(y_c) = -y_c(2) - 0.01.$$

Let  $\varepsilon = 0.05$ , and note that

$$H_o = C_{c_{cl}} (I - A_{cl})^{-1} B_{cl} + D_{c_{cl}} = \begin{bmatrix} 0.097536 \\ 0 \end{bmatrix}.$$

Then, the additional constraints of eq. (33) become

$$-3.9985 \leq r' \leq 3.9985,$$

and  $O_\infty^\varepsilon$  is given by

$$O_\infty^\varepsilon = \{x_g \in \mathfrak{R}^5 : f_i(C_g A_g^t x_g) \leq 0, \quad t = 0, \dots, t_i^*, \quad i = 1, 2, 3, 4,$$



and  $-3.9985 \leq r' \leq 3.9985$

where  $t_1^* = t_3^* = 53$ , and  $t_2^* = t_4^* = 31$ . Recall that an equivalent expression for  $O_\infty^\varepsilon$  is

$$O_\infty^\varepsilon = \{x_g \in \mathbb{R}^5 : \Gamma_g x_g \leq \beta_g\}. \quad (94)$$

Here  $\Gamma_g$  is a  $174 \times 5$  matrix, and  $\beta_g$  is a 174 element vector.

Given  $\Gamma_g$  and  $\beta_g$  the Linear Program (54) generates a 112 element set of unique vertices,  $V$ , that lie on  $C_0(X_s^\varepsilon)$ . Validation of the recursive convex hull algorithm's performance in this four dimensional problem is accomplished next. Also, a comparison of the maximum and minimum allowable steady state conditions, the maximum condition numbers of the matrices,  $M_{k_i}$ , and the number of linear inequality constraints associated with several subsets of  $X_s^\varepsilon$ , is performed.

### 5.3.1 Validation

In this case the set  $X_s^\varepsilon$  is a four dimensional polytope. Each facet of  $X_s^\varepsilon$  is a three dimensional polytope, viz., a tetrahedron, and contains four vertices and four two dimensional subfacets, or 2-faces. Each subfacet is shared by exactly two facets, and contains three of the four vertices contained in each facet. As each new vertex is incorporated, a listing of subfacets is generated for each failed facet. The four subfacets contained in each failed facet are represented by the four unique sets of three out of four vertices contained in each facet. Notice that the convex hull algorithm need only construct the affine hull of each facet of  $X_s^\varepsilon$ , and does not construct the two dimensional subfacets, or other lower dimensional faces. failed subfacets are identified by comparing the vertices contained in subfacets from different failed facets. If the same three vertices appear in subfacets from two different failed facets, then the subfacet represented by these three vertices is contained in the intersection of two failed facets, and is declared failed.

As an example, consider the two facets  $f_1$  and  $f_2$  where  $f_1$  contains the vertices (1,2,5,6) and  $f_2$  contains the vertices (1,4,5,6). Assume  $f_1$  and  $f_2$  were invalidated by the incorporation of the

vertex,  $v_7$ . The subfacets contained in  $f_1$  are (125, 126, 156, 256), and those contained in  $f_2$  are (145, 146, 156, 456). Since the vertices (1, 5, 6) appear in a subfacet from both facets, this subfacet is common to both facets and is invalid. In this case incorporation of  $v_7$  results in elimination of two facets and construction of six new facets.

To demonstrate that the recursive edging algorithm performs as intended several intermediate results are analyzed for a 14 element subset of  $V$ . These intermediate results are compared for consistency, and include the number of facet / hyperplanes invalidated during each pass of the algorithm, the vertices and subfacets contained in each failed facet / hyperplane, the number of valid and failed subfacets associated with the set of failed facets / hyperplanes, the vertices associated with each valid subfacet, the number of new facets / hyperplanes generated, and the vertices contained in each new facet / hyperplane.

Table 2 is an example of the data generated and evaluated for each pass of the algorithm. The numbers in columns 2 through 5 of Table 2 refer to the vertices contained in the associated facets and subfacets. In this case incorporation of vertex,  $v_{11}$ , results in 7 failed facets / hyperplanes and generation of 12 new facets / hyperplanes. Each failed facet contains 4 subfacets for a total of 28 subfacets. Eight subfacets appear in two failed facets, leaving 12 valid subfacets. The vertices from each valid subfacet are combined with  $v_{11}$  to generate the 12 new facets / hyperplanes. As Table 2 shows, the recursive convex hull algorithm performs as designed.

Further validation is also accomplished by comparing results obtained from the recursive convex hull algorithm with those obtained from a brute force algorithm for a 14 element subset of  $V$ . The brute force algorithm generates a hyperplane for every possible combination of 4 vertices using eq. (56). The validity of each hyperplane is then tested using eq. (55). While this brute force method provides accurate results for sparse sets of vertices, it has several drawbacks. First, the number of candidate hyperplanes quickly grows large as the number of states,  $n$ , and the number of

Table 2. Recursive edging algorithm data

<b>New Vertex</b>	<b>Failed Hyperplanes</b>	<b>Failed 2-Faces</b>	<b>Valid 2-Faces</b>	<b>New Hyperplanes</b>
$v_{11}$	1, 2, 3, 5	1, 2, 5	1, 2, 3	1, 2, 3, 11
	1, 2, 5, 7	1, 3, 5	2, 3, 5	2, 3, 5, 11
	1, 2, 6, 7	1, 2, 7	2, 5, 7	2, 5, 7, 11
	1, 3, 5, 9	1, 5, 7	1, 2, 6	1, 2, 6, 11
	1, 5, 7, 9	1, 6, 7	2, 6, 7	2, 6, 7, 11
	1, 6, 7, 9	1, 5, 9	1, 3, 9	1, 3, 9, 11
	6, 7, 9, 10	1, 7, 9	3, 5, 9	3, 5, 9, 11
		6, 7, 9	5, 7, 9	5, 7, 9, 11
			1, 6, 9	1, 6, 9, 11
			6, 7, 10	6, 7, 10, 11
			6, 9, 10	6, 9, 10, 11
			7, 9, 10	7, 9, 10, 11

vertices,  $K$ , increase. For example, for  $n = 4$ , the number of candidate hyperplanes is

$$\frac{(K)(K-1)(K-2)(K-3)}{4!}.$$

Thus, with  $n = 4$ , and  $K = 14$ , the brute force algorithm must consider 1,001 possible hyperplanes.

If the number of vertices is increased to 30, the number of candidate hyperplanes becomes 27,405.

The reliability of the brute force algorithm is also sensitive to the numerical precision required by eq. (55). Vertices that are not in general position, or closely spaced vertices, such as those that occur in clusters, increase the numerical precision required for a particular problem, which may in turn exceed the limitations of the computing system. Thus, to insure valid results the brute force algorithm must only be applied to an appropriately thinned set of vertices.

In this case the maximum condition number obtained for the matrices  $M_{k_i}$  is  $2.32 \times 10^5$ . Thus, the 14 vertices clearly lie in general position. The brute force algorithm generated 44 hyperplanes from the 14 element subset of  $V$ . The recursive convex hull algorithm generated the same 44 hyperplanes. The lists of vertices contained in each hyperplane is provided in Appendix I. This is an important result because, in the absence of numerical precision problems, the brute force algorithm must produce valid results.

### 5.3.2 Polyhedral Subset Comparisons

An important consideration in choosing a specific subset of  $V$ , is the set of steady-state conditions supported by the resulting constraints. For a particular set of constraints,  $\Gamma_\delta x \leq \beta_\delta$ , the maximum and minimum steady-state values of  $r'$  may be determined by noting that if  $x_{ss}$  is an admissible steady-state state vector, then

$$\Gamma_\delta x_{ss} \leq \beta_\delta. \quad (95)$$

Also,

$$x_{ss} = (I - A_{cl})^{-1} B_{cl} r'_{ss}$$

where  $r'_{ss}$  is the steady-state modified reference input. Thus,

$$\Gamma_{\delta} (I - A_{cl})^{-1} B_{cl} r'_{ss} \leq \beta_{\delta}. \quad (96)$$

Recall that by construction all elements of  $\beta_{\delta}$  are 1, viz.,  $\beta_{\delta} = [1 \ 1 \ 1 \ \dots \ 1]^T$ . Also, since  $0 \in \text{int}(X_s^{\varepsilon})$  we have that

$$\min \left( \left[ \Gamma_{\delta} (I - A_{cl})^{-1} B_{cl} \right] \right) < 0.$$

Let  $\Phi = \Gamma_{\delta} (I - A_{cl})^{-1} B_{cl}$ , then the maximum and minimum values of  $r'_{ss}$  are given by

$$\begin{aligned} r'_{ss \max} &= \frac{1}{\max(\Phi)} \\ r'_{ss \min} &= \frac{1}{\min(\Phi)}. \end{aligned} \quad (97)$$

The maximum and minimum values of  $r'_{ss}$  provide one measure of the suitability of a particular subset of  $V$ . As a benchmark for this problem consider the maximum and minimum values of  $r'_{ss}$  allowed by  $O_{\infty}^{\varepsilon}$ , which are  $\pm 3.9985$  rad/sec. Also, an important property of  $V$  is that it is composed of 56 pairs of symmetrical vertices. This is a consequence of the symmetrical constraints in this problem. Thus, a requirement that candidate subsets of  $V$  contain a symmetrical set of vertices is enforced. This naturally results in  $r'_{ss \max} = -r'_{ss \min}$ .

Another consideration in choosing a particular subset of vertices is the maximum condition number obtained over all passes of the convex hull algorithm for the matrices,  $M_{k_i}$ . As noted in Chapter 4, the maximum condition number obtained for a particular set of vertices should be less than  $1 \times 10^{15}$  to insure the vertices lie in general position. Table 3 summarizes the results for several subsets of  $V$ .

Table 3 demonstrates the trade-off that exists between the number of linear inequality constraints required to characterize a particular set,  $X_s^{\delta_i}$ , and the resulting allowable steady state conditions. A significant reduction in the magnitude of allowable steady state conditions may indicate a like reduction in the volume of  $X_s^{\delta_i}$ , which in turn may indicate a significant degradation in tracking performance. Also notice the condition numbers in the last column of Table 3. Clearly, the vertices

Table 3. Results for several subsets of V

d	Vertices	Set	No. LIs	$r'_{ss_{\max}} \left( \frac{rad}{sec} \right)$	$r'_{ss_{\min}} \left( \frac{rad}{sec} \right)$	$\max(\text{cond}(M_{k_i}))$
0.1	60	$X_s^{\delta_{60}}$	377	3.847	-3.847	undefined
0.5	36	$X_s^{\delta_{36}}$	182	3.847	-3.847	$6.44 \times 10^{16}$
0.7	28	$X_s^{\delta_{28}}$	136	3.845	-3.845	$3.29 \times 10^{17}$
1.0	20	$X_s^{\delta_{20}}$	85	3.830	-3.830	$1.09 \times 10^{17}$
1.5	18	$X_s^{\delta_{18}}$	64	3.632	-3.632	$2.46 \times 10^5$
1.8	14	$X_s^{\delta_{14}}$	44	3.219	-3.219	$2.32 \times 10^5$

of the 60, 36, 28, and 20 element subsets do not lie in general position, while those of the 18 and 14 element subsets do lie in general position. The results of Table 3 indicate that the 18 element subset may provide the best trade-off between performance and computational burden. This will be demonstrated in the following section.

## 5.4 Simulation Results

The unconstrained system's pitch rate and elevator rate responses to a 0.1 amplitude step input are shown in Fig. 66. Notice that the elevator rate clearly violates its constraint even for this relatively small input. The elevator displacement, not shown, is well within its limits in this case with a maximum amplitude of 0.0332 radians. This is typical of systems with both elevator rate and elevator displacement constraints. That is, elevator rate constraint violation generally transpires well before elevator displacement constraint violation becomes an issue. The constrained system's pitch rate and elevator rate responses to the same 0.1 amplitude step input are shown in Fig. 67. With no saturation effects mitigation method employed the elevator rate saturates immediately, and an unstable system response ensues.

Pitch rate and elevator rate responses of the SRSG and DTRG systems are presented in Fig. 68. Both reference signal governors avoid constraint violation, and produce stable responses. The SRSG provides slightly improved tracking over that of the DTRG. Also, the SRSG is implemented using the set,  $X_s^{\delta_{18}}$ , or 64 inequality constraints. Recall that the DTRG requires 174 inequality constraints in this case. Thus, the SRSG system provides improved tracking performance with less on-line computational burden.

In this case the set,  $X_s^{\delta_{18}}$ , provides a reasonable trade-off between computational burden and system performance. This is also consistent with the results of Table 3. On the other hand Table 3 indicates that the volume of  $X_s^{\delta_{14}}$  is substantially less than that of  $X_s^{\delta_{18}}$ . Thus, implementing the

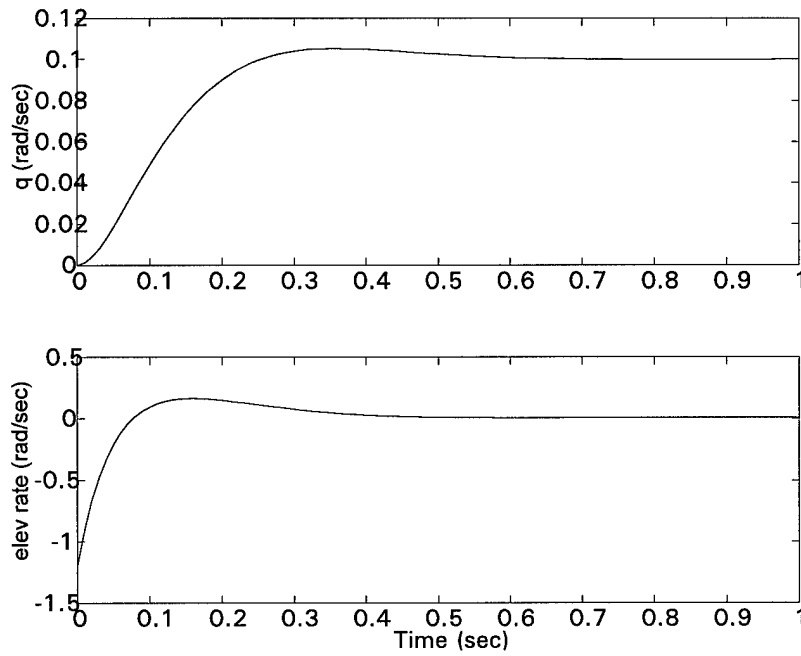


Figure 66. Unconstrained system responses to a 0.1 amplitude step.

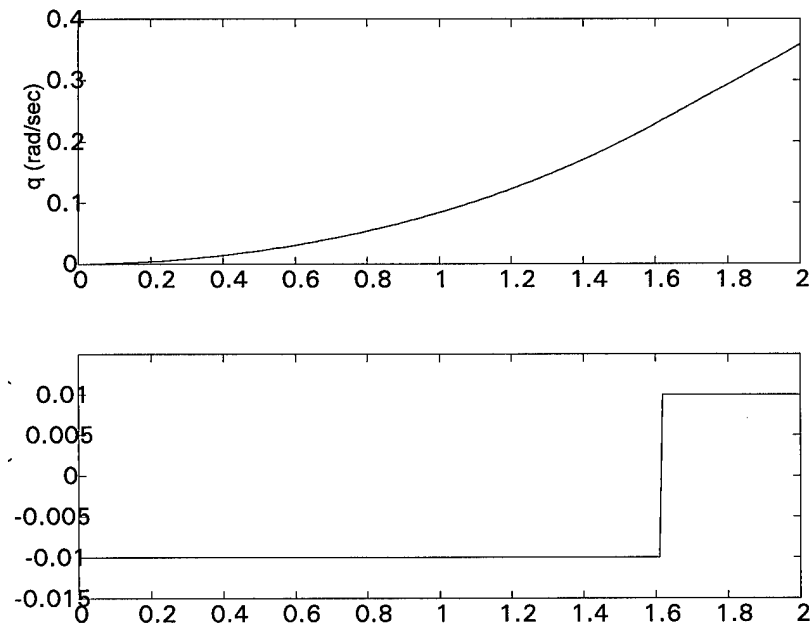


Figure 67. Constrained system responses to a 0.1 amplitude step.



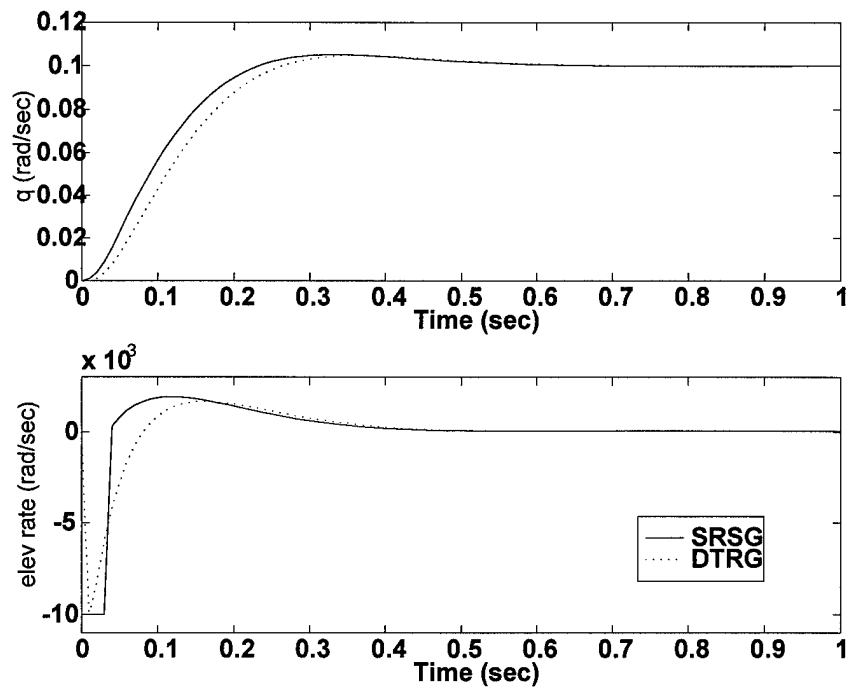


Figure 68. SRS and DTRG system responses to a 0.1 amplitude step.

reference signal governor using  $X_s^{\delta_{14}}$  may result in poor tracking performance. This is demonstrated in Fig. 69. In Fig. 69 the MDRSG system pitch rate response to a unit step input is compared for the sets  $X_s^{\delta_{36}}$ ,  $X_s^{\delta_{18}}$ , and  $X_s^{\delta_{14}}$ . While the sets  $X_s^{\delta_{36}}$  and  $X_s^{\delta_{18}}$  provide similar tracking performance, the set  $X_s^{\delta_{14}}$  results in a much slower response. This is a result of the more restrictive admissibility constraints associated with  $X_s^{\delta_{14}}$ , Fig. 70. Figure 70 is a plot of the admissibility constraint limits imposed by the three sets of linear inequalities. Clearly, the state trajectory encounters the boundary of  $X_s^{\delta_{14}}$  much earlier than it does the boundaries of either  $X_s^{\delta_{36}}$  or  $X_s^{\delta_{18}}$ . It is important to note that the state trajectories for the three cases of Fig. 70 only coincide during the initial 0.13 seconds, or 13 time steps. Thus, this initial interval provides the best measure of the differences and similarities between the three sets of linear inequalities. The results shown in Figs. 69 and 70 are consistent with the sharp reduction in the magnitude of the allowable steady state conditions associated with  $X_s^{\delta_{14}}$ , Table 3. Finally, a comparison of the MDRSG- $X_s^{\delta_{18}}$  and DTRG tracking performance is shown in Fig. 71. While the MDRSG- $X_s^{\delta_{18}}$  response is only slightly faster than that of the DTRG system, it only requires evaluation of 64 versus 174 linear inequality constraints at each time step.

## 5.5 Summary

A realistic flight control application is presented in this chapter. The controlled process is a fourth-order system with an open-loop unstable plant, and is subject to both state and control constraints. An approximation of the maximal output admissible set,  $O_\infty^\varepsilon$ , is characterized with 174 linear inequality constraints, and the LP (54) produces a set of 112 unique vertices,  $V$ . The recursive convex hull algorithm is performed on several subsets of  $V$  to obtain several polyhedral sets,  $X_s^{\delta_i} \subset X_s^\varepsilon$ .

Several intermediate results of the recursive convex hull algorithm are examined to verify that it performs as intended. Moreover, the results of the recursive convex hull algorithm are compared

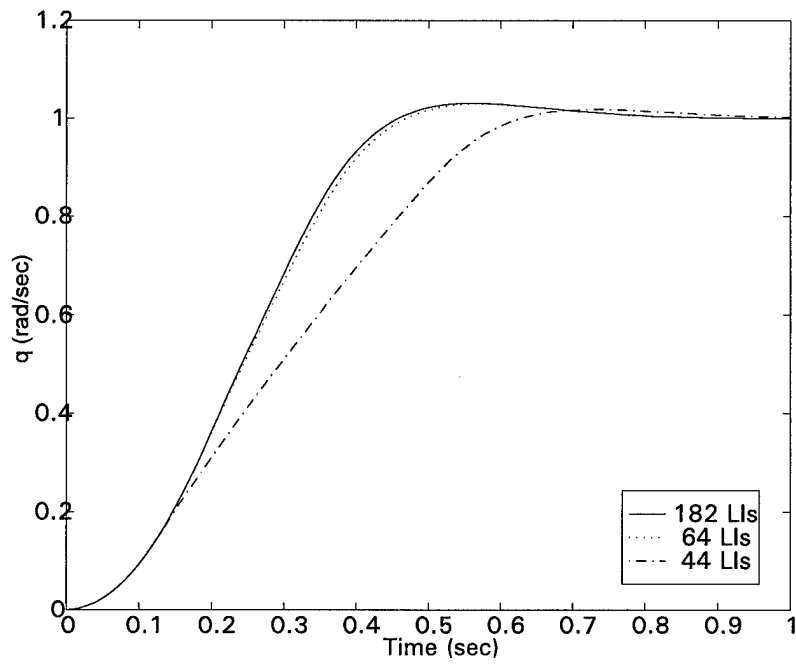


Figure 69. MDRSG tracking performance comparison for a 1.0 amplitude step.

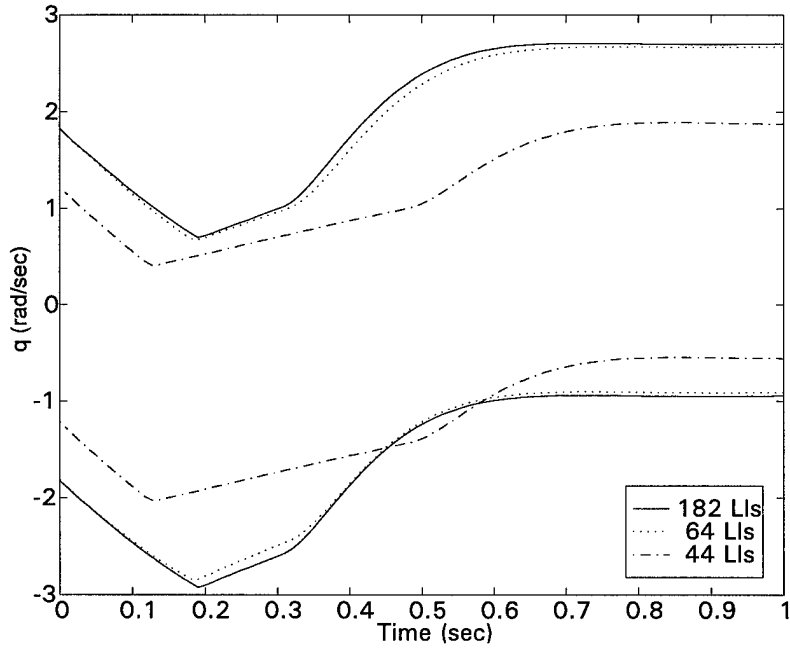


Figure 70. Comparison of admissibility constraints.

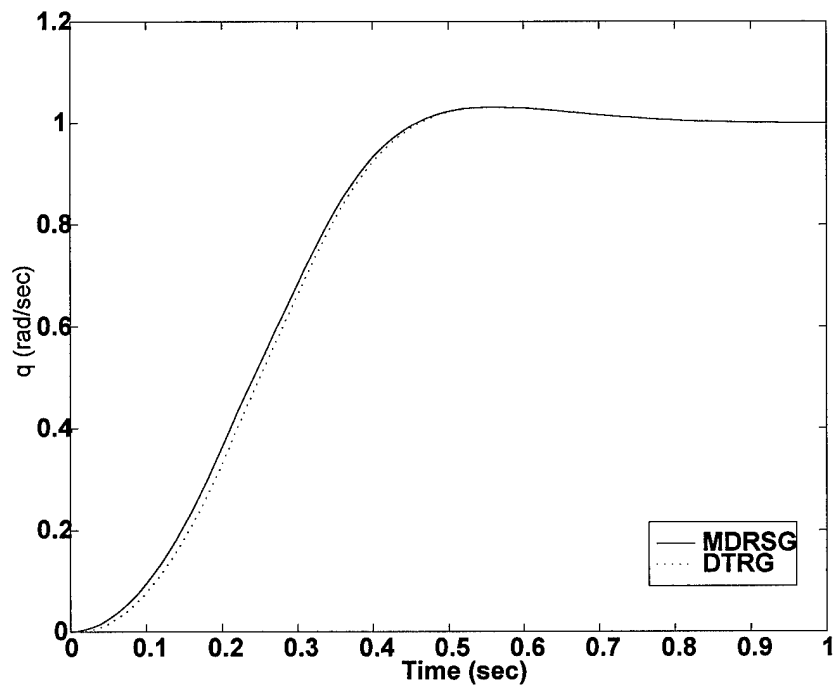


Figure 71. Comparison of MDRSG and DTRG performance

to those obtained with a brute force algorithm for a 14 element subset of  $V$ . This comparison shows that the same 44 linear inequality constraints are obtained with both algorithms, and is a practical demonstration of the validity of the recursive convex hull algorithm.

A comparison of allowable steady state conditions and tracking performance obtained for several sets,  $X_s^{\delta_i}$ , shows that a significant reduction in the number of vertices in  $V' \subset V$ , and thus the number of linear inequalities required to characterize  $X_s^{\delta_i}$ , is possible with little impact on system performance. In this case, very little improvement in closed-loop system performance is obtained when the number of vertices in  $V'$  is increased beyond 18. However, a substantial loss in both allowable steady state conditions and tracking performance are encountered with a 14 element subset. Moreover, implementing reference signal governors with  $X_s^{\delta_{18}}$ , characterized with 64 LIs, provides faster tracking responses to step inputs than that of the DTRG which requires 174 LIs. Finally, comparison of the maximum condition numbers obtained for the matrices,  $M_{k_i}$ , also indicates that the 18 element subset of vertices is an appropriate choice for this problem.

## Chapter 6 - The Constrained Regulator Control Problem

### 6.1 Overview

In this chapter the constrained regulator control problem is viewed as a special case of the constrained tracking control problem, and is addressed with the nonlinear dual loop controller architecture of Fig. 72. In this case the exogenous reference signal is identically zero, and the outer loop nonlinearity is a reference signal generator that produces nonzero reference signal increments, when necessary, to avoid saturation. This results in a substantial increase in the size of the maximal statically admissible set for a given controlled process, and reduces the inherent trade-off between achievable small signal performance, and the size of the maximal statically admissible set.

### 6.2 Reference Signal Generator

The following discrete-time regulator control problem is considered

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^1, \quad t \in Z^+ = \{0, 1, 2, \dots\} \quad (98)$$

where the open-loop dynamics matrix,  $A$ , may have eigenvalues outside the unit circle, viz., the open-loop system is unstable. Also, assume there are hard constraints on either or both the state and control variables, including linear combinations of them. With appropriate choices of matrices,  $C_c$  and  $D_c$ , and output constraint set,  $Y \subset \mathbb{R}^p$ , all such constraints may be expressed in terms of output constraints, viz.,

$$y_c(t) = C_c x(t) + D_c u(t) \in Y \subset \mathbb{R}^p. \quad (99)$$

Assume that the control signal,  $u$ , is given by

$$u(t) = k_x x(t). \quad (100)$$

Then, the closed-loop system and its constraint may be expressed as

$$\begin{aligned} x(t+1) &= A_{cl} x(t) \\ y_c(t) &= C_{cl} x(t) \in Y \subset \mathbb{R}^p \end{aligned} \quad (101)$$

where,  $A_{cl} = (A + Bk_x)$  and  $C_{cl} = (C_c + D_c k_x)$ . The state constraint set,  $X$ , is defined as

$$X = \{x \in \mathfrak{R}^n : C_{cl}x(t) \in Y\} \subset \mathfrak{R}^n.$$

We are concerned with the case where the constraint set,  $Y$ , is a polytope which contains the origin.

That is,  $Y$  may be expressed as

$$Y = \{y_c \in \mathfrak{R}^p : f_i(y_c) \leq 0, \quad i = 1, 2, \dots, s\} \quad (102)$$

where, the  $s$  functionals,  $f_i : \mathfrak{R}^p \rightarrow \mathfrak{R}$ , are linear in  $y_c$ , with  $f_i(0) \leq 0$ .

Since the open-loop system of eq. (78) is unstable, global stability can not be achieved, and one is instead interested in determining a control law such that the resultant closed-loop system is asymptotically stable for the largest possible set of initial states,  $X_0$ . This problem has been addressed in the literature using the concept of positive invariance [3], [4], [5], [10], and [26]. Typically, Lyapunov functions are used to determine a stabilizing state feedback matrix,  $k_x$ , for which a predetermined set of initial states,  $X_0 \subset \mathfrak{R}^n$ , is contained in a set,  $W \subset \mathfrak{R}^n$ , which is admissible and positively invariant with respect to the closed-loop system. That is,  $k_x$  is determined such that  $|\lambda_i(A + Bk_x)| < 1$ , and such that for all  $x(0) \in X_0 \subset W$ , we have that  $x(t) \in W$  and  $y_c(t) \in Y$  for all  $t \in Z^+$ . Clearly,  $W$  is a positively invariant subset of  $X$ .

Since  $Y$  is polyhedral the maximal positively invariant subset of  $X$ ,  $X_r$ , associated with a particular state feedback matrix,  $k_x$ , is also polyhedral. Thus, non-quadratic Lyapunov functions are exploited in [3], [4], [5] and [26] to obtain polyhedral positively invariant sets,  $W \subset X \subset \mathfrak{R}^n$ . However, using a Lyapunov function to determine  $k_x$  and  $W$  generally results in  $W \subset X_r$ , because the existence of a Lyapunov function is a sufficient, but not necessary, condition for asymptotic stability. Thus, these methods result in a conservative estimate of the set of allowable initial conditions for a given state feedback matrix. This conservatism may be eliminated by using the methods of [8] to characterize  $X_r$ . More importantly, these methods generally provide very conservative results because only linear control laws are considered. Thus, small signal performance must be sacrificed

to insure constraint violation is avoided for large initial state disturbances [7]. In general, a trade-off exists between the size of  $X_0$  and closed-loop system performance. Increasing the size of  $X_0$  generally results in degraded closed-loop system performance, such as slower response times. Evidently, and as noted in [7], nonlinear control laws can produce improved closed-loop system performance, for a given predetermined set  $X_0$ , over that achievable with linear control laws. However, as noted in Chapter 1, nonlinear control laws based on solutions to optimal control problems are generally impractical to implement.

Thus, the stated problem is addressed with the nonlinear dual loop controller architecture of Fig. 72. Of course, implementation of the dual loop control law requires modification of the existing linear regulator control law to afford tracking of the synthetically generated reference signal,  $r'$ .

As an example, consider the system of eq. (78), and assume the constrained control signal is given by

$$-1 \leq u(t) = k_x x(t) \leq 1.$$

Let  $X_r \subset \mathfrak{R}^n$  denote the maximal positively invariant subset of  $X$  obtained for a particular stabilizing state feedback matrix,  $k_x$ . The set,  $X_r$ , is the set of all initial states in  $\mathfrak{R}^n$  such that  $|u(t)| \leq 1$  for all  $t \in Z^+$ . Now, consider the associated tracking control law

$$-1 \leq u(t) = k_x x(t) + k_r r'(t) \leq 1.$$

Here,  $k_x$  is the same as before, and  $k_r$  is chosen to provide asymptotic tracking. Let  $X_s \subset \mathfrak{R}^n$  denote the set of all initial states for which there exists a constant reference input,  $r'$ , such that  $|u(t)| \leq 1$  for all  $t \in Z^+$ . The set,  $X_s$ , is the *maximal statically admissible* set for the closed-loop system. Now we can avoid constraint violation when  $|k_x x(t)| > 1$  by allowing non-zero reference signals,  $r'$ . Thus, we have that  $X_r \subset X_s$ . While this example considers the case where the regulator control law consists of a state feedback matrix, it is important to note that the proposed methodology is applicable to any linear regulator control system.



The reference signal generator may be implemented so that for small disturbances, for which  $x(0) \in X_r$ , we have  $r' = 0$ . In this case, the system response is determined by the original regulator control law for small disturbances. Thus, for a given regulator control law, the size of the set of allowable initial conditions may be substantially increased without sacrificing small signal performance. Conversely, small signal performance may be improved for a prespecified set of initial conditions,  $X_0$ , by implementing a more aggressive inner-loop linear control law. The proposed methodology requires extensive off-line calculations, but on-line implementation of the control law is practical. Moreover, the closed-loop system is asymptotically stable for all  $x \in X_s$ .

### 6.3 Second-Order Example

The regulator synthesis method described here is demonstrated with a second-order example.

Consider the constrained regulator problem of Fig. 73 and defining equation

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \\ -1 \leq u(t) &= k_x x(t) \leq 1 \end{aligned} \tag{103}$$

where,

$$\begin{aligned} A &= \begin{bmatrix} 1.0001 & 1.0051 \times 10^{-2} \\ 2.0101 \times 10^{-2} & 1.0102 \end{bmatrix}, B = \begin{bmatrix} 1.0034 \times 10^{-4} \\ 2.0101 \times 10^{-2} \end{bmatrix} \\ C &= [ 1 \ 0 ], \text{ and } k_x = [ -9 \ -2.5 ]. \end{aligned}$$

Notice that this is the open-loop system and state feedback matrix of eqs. (65) and (67), Section 3.4.1. The constrained variable is the control signal,

$$y_c(t) = u(t) \in Y$$

where

$$Y = \{y_c \in \mathfrak{R} : f_i(y_c) \leq 0, \quad i = 1, 2\}$$

and

$$f_1(y_c) = y_c - 1$$

$$f_2(y_c) = -y_c - 1.$$

Also, the open-loop system is unstable with  $\lambda_1(A) = 0.99005$  and  $\lambda_2(A) = 1.0202$ .

The closed-loop system and constraint are given by

$$\begin{aligned} x(t+1) &= A_{cl}x(t) \\ y(t) &= Cx(t) \\ y_c(t) &= C_{ccl}x(t) \in Y \end{aligned} \quad (104)$$

where,  $A_{cl} = (A + Bk_x)$ , and  $C_{ccl} = k_x$ . Now,  $|\lambda_{1,2}(A_{cl})| = 0.9802$ , the pair  $(C_{ccl}, A_{cl})$  is observable, and  $Y$  is a polyhedron that contains the origin. Thus, the methods of Section 2.3 may be used to characterize  $X_r$  in terms of eq. (25).

In this case 84 inequalities are required to characterize  $X_r$ , and  $\Gamma_r$  is an  $84 \times 2$  element matrix. Figure 74 shows  $X_r$  bounded by the 84 half-spaces. The set,  $X_r$ , is the maximal statically admissible set for the closed-loop system of eq. (104). To obtain a larger statically admissible set for this control system, the state feedback matrix must be modified. However, if the constrained regulator control problem is viewed as a special case of the constrained tracking control problem, with  $r \equiv 0$ , then a much larger statically admissible set may be obtained for the existing state feedback matrix. Thus, consider the nonlinear tracking control system of Fig. 75. Then the system of eq. (103) becomes

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y &= Cx(t) \\ -1 &\leq u(t) = k_x x(t) + k_r r'(t) \leq 1. \end{aligned} \quad (105)$$

Now, although  $r \equiv 0$ , the nonlinear feedback reference generator,  $N$ , may generate a nonzero exogenous reference signal,  $r'(t)$ , in the event  $|k_x x(t)| > 1$ . Thus, for a given state feedback

matrix,  $k_x$ , control signal saturation may be avoided for a much larger set of initial conditions. This improves the robustness of the closed-loop system with respect to external disturbances without sacrificing small signal performance.

Since the existing regulator control system does not include integral action in this example, we enforce asymptotic tracking by choosing  $k_r$  such that

$$k_r = CA_{cl}^{-1}B = 8.$$

Then, the linear controlled process of Fig. 75 is given by

$$\begin{aligned} x(t+1) &= A_{cl}x(t) + B_{cl}r'(t) \\ y &= Cx(t) \\ y_c(t) &= C_{c_{cl}}x(t) + D_{c_{cl}}r'(t) \in Y \end{aligned} \quad (106)$$

where  $A_{cl} = (A + Bk_x)$ ,  $B_{cl} = Bk_r$ ,  $C_{c_{cl}} = k_x$ , and  $D_{c_{cl}} = k_r$ . A finitely determined approximation of  $X_s \subset \mathfrak{R}^2$  may be developed using the methods of Section 3.4.1, where  $X_s^\varepsilon \subset X_s$  is given by eq. (52). In Section 3.4.1  $X_s^\varepsilon$  is characterized for the controlled process of eq. (106) with 86 linear inequality constraints.

Figure 20, repeated below in Fig. 76, is a plot of the 86 supporting hyperplanes of  $X_s^\varepsilon$ . A comparison of  $X_r$  and  $X_s^\varepsilon$  is shown in Fig. 77. Clearly,  $X_r \subset X_s^\varepsilon$ . Thus, the size of the positively invariant set has been increased without sacrificing small signal performance. Moreover,  $X_s^\varepsilon$  is an arbitrarily close approximation to the maximal statically admissible set,  $X_s$ , for the given controlled process. Finally it should be noted that  $X_s^\varepsilon \subset X_I$ , and that by enforcing the feasibility constraints when  $x \in X_I \overset{set}{=} X_s$ , the controlled process state can be made to converge to the origin for any  $x \in X_I$ .

Constraint mitigation is accomplished using the MDRSG algorithm of Section 4.3 with the additional caveat that when  $x \notin X_s^\varepsilon$ , only the feasibility constraints are enforced. Then for small

disturbances we have that  $x(0) \in X_r$ , and  $r'(t) = r(t) = 0$  for all  $t$ . In this case, the closed-loop system response is dictated by  $k_x$ . However, when  $x(0) \in \{X_I - X_r\}$  transpires, as may result from larger disturbances, then  $r'(t) \neq 0$  ensues, and the closed-loop system response is modified, but stable. Thus, for the specified state feedback matrix,  $k_x$ , we have substantially increased the size of the set of admissible initial conditions without sacrificing small signal performance. Of course, if  $x(0) \notin X_I$  should transpire, the system may diverge because the open-loop plant is unstable. In general a characterization of  $X_I$  is difficult or impossible to obtain. Hence, we require that  $X_0 \subset X_s^\epsilon$ .

#### 6.4 Simulation Results

An example of the simulation results obtained for the second-order problem with the dual-loop controller architecture and nonlinear reference generator are presented in Figures 78 and 79. The initial condition for this case is  $x(0) = [ 0.9, -0.5 ]^T$ . Figure 78 shows the state trajectory transposed on plots of  $X_s^\epsilon$  and  $X_r$ . This demonstrates that for any  $x(0) \in X_s^\epsilon$  we have  $x(t) \in X_s^\epsilon$  for all  $t \in Z^+$ . The nonlinear nature of the reference generator is evident in the control signal and reference signal responses shown in Fig. 79. The control signal response also shows that constraint violation is avoided. The transitions in the control signal and reference signal responses at approximately 0.4 and 0.95 seconds coincide with the state trajectory encountering and leaving the boundary of  $X_s^\epsilon$ . At about 0.95 seconds we have  $x(t) \in X_r$ , and thereafter the system response is dictated by  $k_x$ .

An example of simulation results obtained for the state feedback control law is shown in Fig. 80. The initial condition is again  $x(0) = [ 0.9, -0.5 ]^T$ . In this case control signal saturation is encountered immediately since  $x(0) \notin X_r$ , and the system diverges. While there are certainly cases

for which the linear state feedback control system does not diverge when  $x(0) \notin X_r$ , constraint violation will always occur in these cases.

## 6.5 Summary

Positively invariant sets play a central role in controller synthesis methodologies that address the constrained regulator control problem. Many of these methods exploit Lyapunov functions to obtain state feedback matrices for which a statically admissible set,  $W \subset \mathbb{R}^n$ , is positively invariant. However, these techniques must trade-off small signal performance against the size of  $W$ . On the other hand, nonlinear controllers based on solutions to optimal control problems are generally impractical to implement.

Here, the constrained regulator problem is addressed with a dual-loop nonlinear controller synthesis methodology. An outer-loop reference signal generator injects admissible non-zero reference signals into an inner-loop feedback control system. The reference signals are tracked by the inner-loop linear control law. Small signal performance is then preserved while simultaneously the size of the maximal statically admissible set associated with the controlled process is significantly increased. Moreover, on-line implementation of the control law is practical, and, most importantly, the closed-loop system is asymptotically stable for all  $x \in X_I$ .

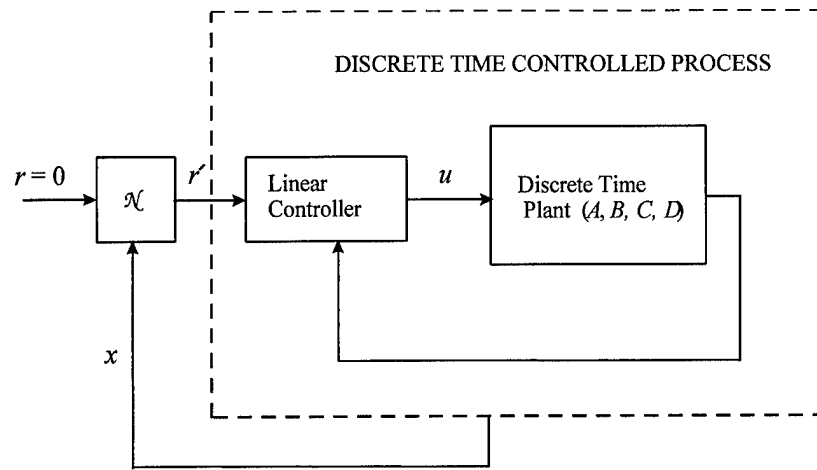


Figure 72. Nonlinear dual-loop controller architecture.

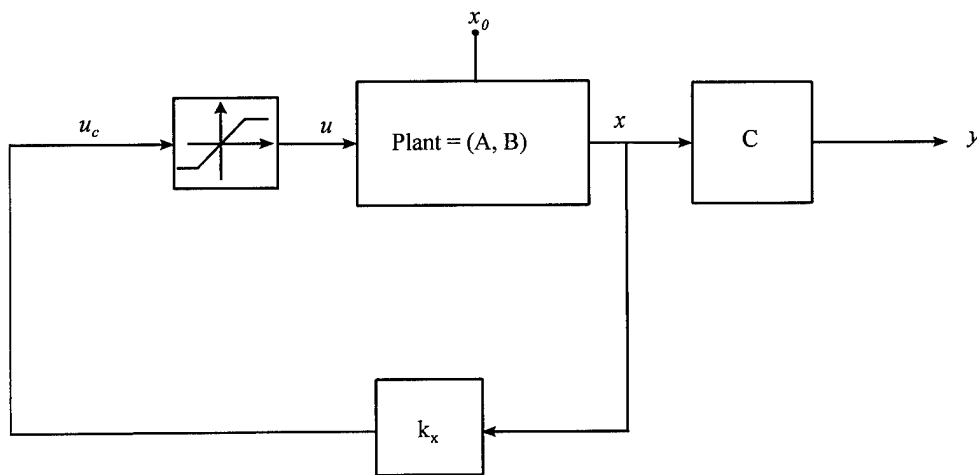


Figure 73. Constrained regulator control problem.

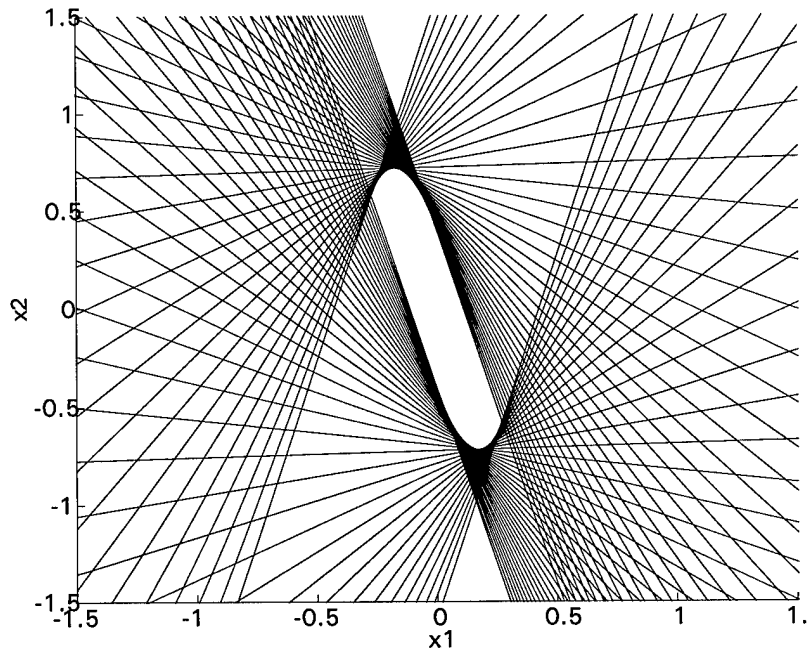


Figure 74.  $X_r$  for the second-order regulator control system.

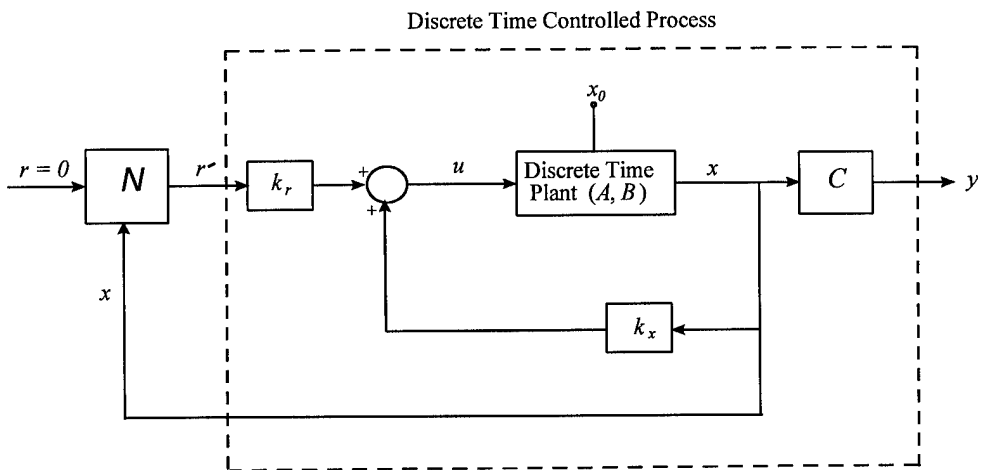


Figure 75. Nonlinear dual-loop controller architecture.

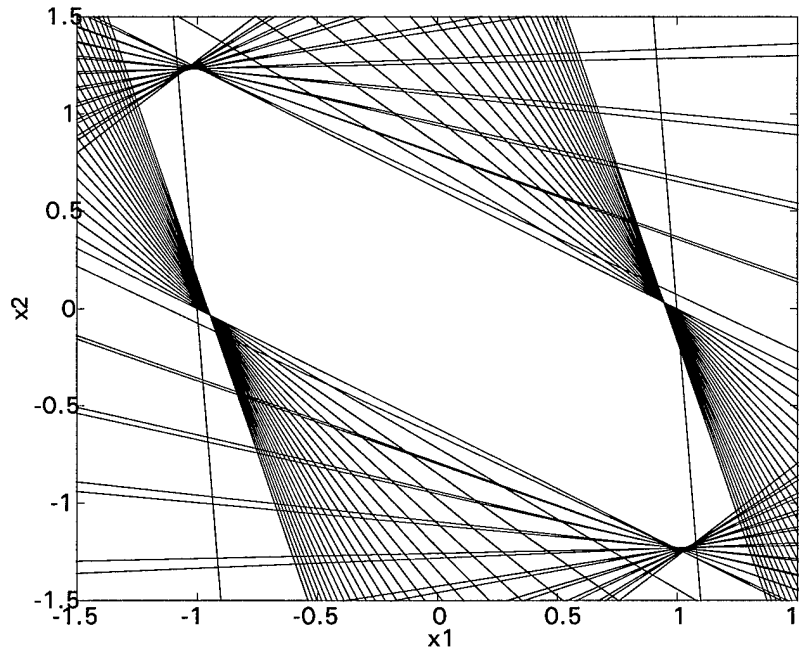


Figure 76. Supporting hyperplanes for  $X_s^\epsilon$ .

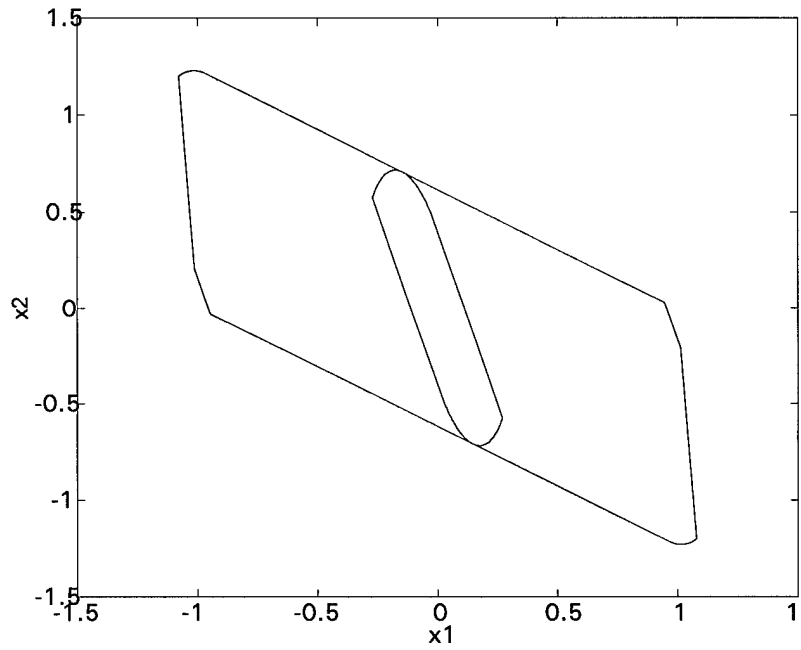


Figure 77. Comparison of  $X_r$  and  $X_s$ .



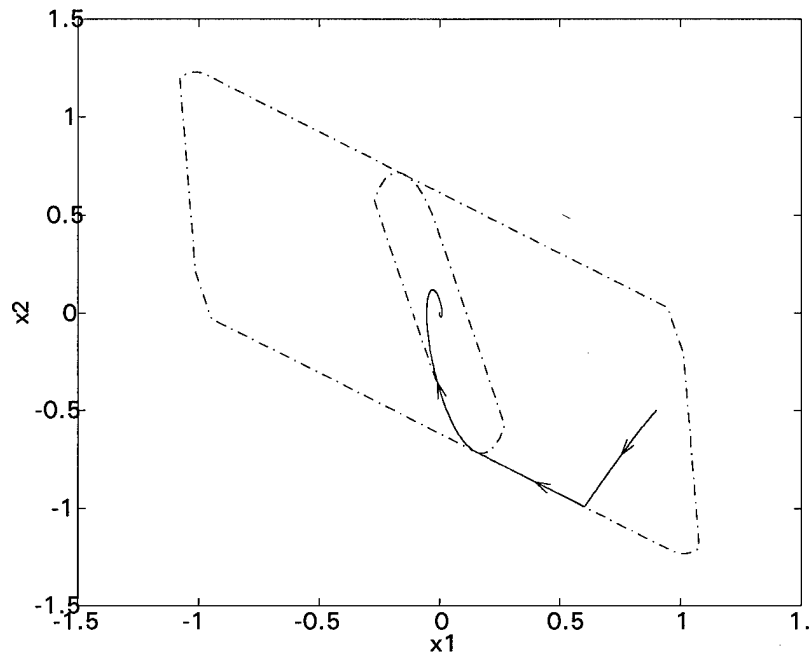


Figure 78. Dual-loop control system response for  $x(0) = [0.9, -0.5]^T$ .

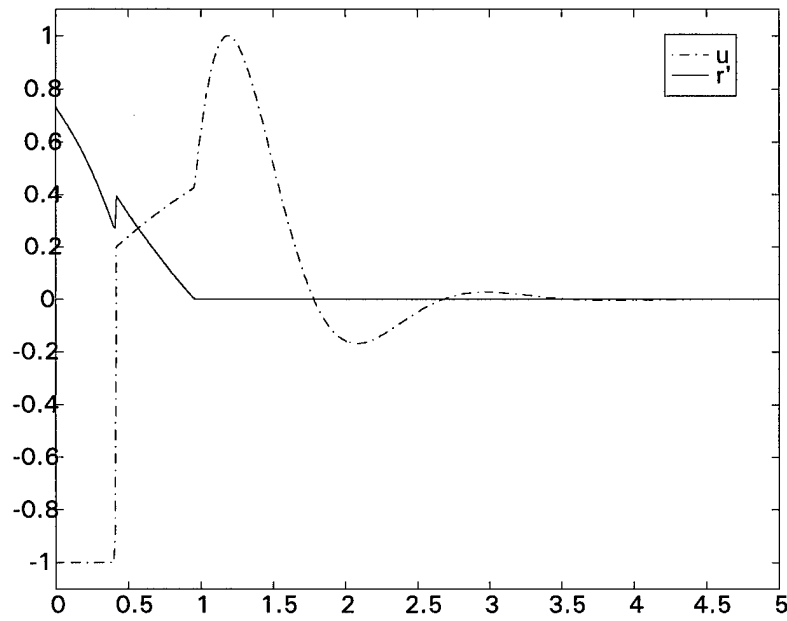


Figure 79. Responses of  $u$  and  $r'$  for  $x(0) = [0.9, -0.5]^T$ .

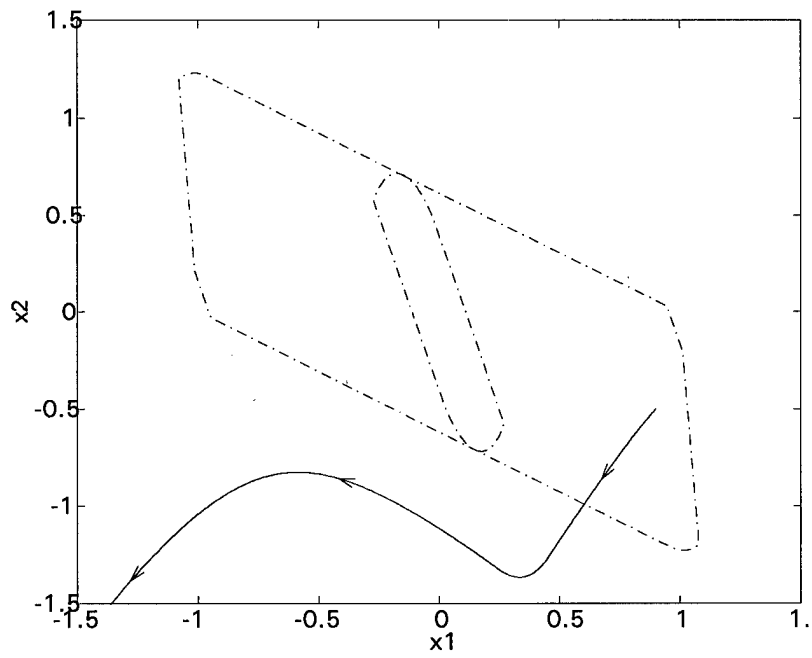


Figure 80. Regulator control system response for  $x(0) = [0.9, -0.5]^T$  when saturation mitigation is not employed.

## Chapter 7 - Conclusion

### 7.1 Overview

The problem of high amplitude tracking control of dynamic reference signals in the face of hard control and state constraints, and open-loop unstable plants is investigated. In the case of open-loop unstable plants and control and state constraints, global stability cannot be achieved. Then, one is interested in characterizing the largest possible positively invariant set of initial states, and in restricting the state vector to this set. Here, the maximal statically admissible set is characterized for a discrete-time constrained control system, and a nonlinear control methodology is proposed that restricts the state vector to this positively invariant set. Moreover, the proposed method does not unnecessarily restrict the feasible reference signal to statically admissible values. The proposed reference governor methodology is outlined in this chapter, and specific contributions of this research are presented. Specific conclusions and recommendations for further research are also provided.

### 7.2 Contributions and Accomplishments

Real world physical systems are generally subject to physical limitations such as actuator displacement and rate constraints. These physical limitations translate into hard constraints on linear system model control and state variables. Unfortunately, physical system limitations are often ignored in the initial controller design process since linear control system design methods cannot accommodate hard nonlinearities. This leads to control laws that may generate unrealizable control signals, which result in constraint violation, even for moderate exogenous reference signals. Thus, many ad hoc post-design techniques have been proposed in the literature to mitigate the effects of constraint violation. For the most part, constraint effects mitigation methodologies focus on improving system performance through various anti-windup techniques. These techniques do not

fully prevent constraint violation, and thus do not perform well in the case of open-loop unstable plants and high amplitude dynamic reference signals.

Several recent research efforts have attacked the constrained tracking control problem with dual-loop controller architectures based on the concepts of static admissibility and positive invariance. In these investigations a linear inner-loop control law is implemented which provides good small signal tracking performance and stability in the absence of constraint violation. Then the controlled process is augmented with a nonlinear supervisory loop which contains a reference signal governor. The reference signal governor prevents constraint violation by generating a modified reference signal which depends nonlinearly on the exogenous reference signal and the controlled process state. A common characteristic of these reference signal governor methods is that to obtain a BIBO stable closed-loop system, the modified reference signal is restricted to statically admissible values. However, This is not strictly necessary. It is sufficient to restrict the controlled process state vector to a positively invariant set to obtain a BIBO stable closed-loop system. Thus, these methods unnecessarily sacrifice achievable tracking performance to obtain guaranteed BIBO stability of the resulting closed-loop system. While the current dual loop reference signal governor methods do sacrifice tracking performance to some degree, they do have several admirable attributes, including: guaranteed BIBO stability, preservation of small signal performance, improved large signal performance, complete constraint violation avoidance, and reasonable on-line computational burden. Thus, the focus of this research is the development of a dual-loop reference signal governor methodology that removes the requirement of static admissibility from the modified reference signal, results in a BIBO stable closed-loop system, and has similar, or less, on-line computational burden than current reference signal governor methods.

Specifically, the discrete time controlled process

$$x(t+1) = A_d x(t) + B_d r'(t), \quad x \in \mathfrak{R}^n, \quad r' \in \mathfrak{R}$$

$$y(t) = C_{cl}x(t) + D_{cl}r'(t)$$

with constraint vector,  $y_c$ , given by

$$y_c(t) = C_{ccl}x(t) + D_{ccl}r'(t) \in Y \subset \mathbb{R}^p$$

is considered. Here the constraint set,  $Y$ , is a polytope that contains the origin, and  $r'$  is the modified reference signal generated by the reference signal governor. The proposed reference signal governor restricts the controlled process state to an admissible positively invariant set without unnecessarily restricting  $r'$  to statically admissible values. This is accomplished by obtaining an arbitrarily close approximation,  $X_s^\varepsilon$ , of the controlled process' maximal statically admissible set,  $X_s$ , in terms of a finite set of linear inequality constraints, viz.,

$$X_s^\varepsilon = \{x \in \mathbb{R}^n : \Gamma_\varepsilon x \leq \beta_\varepsilon\}$$

No previous characterization of this set has been accomplished. Moreover, this set is shown to be positively invariant with respect to the controlled process. Thus, to obtain a BIBO stable closed-loop system, at each time increment,  $t$ , it is sufficient to insure that  $r'(t)$  is chosen such that  $y_c(t) \in Y$  and  $x(t+1) \in X_s^\varepsilon$ . Additional criteria are also selected by the designer to determine the "optimal" allowable modified reference signal.

Implementation of the proposed dual-loop tracking control law consists of:

1. Development of a linear, small signal controller that provides the desired level of performance. Together, the open-loop plant and linear control law comprise the controlled process. Excessive small signal tracking control specifications should be avoided since this may reduce the volume of the controlled process' maximal statically admissible set.
2. Express the controlled process state and control constraints in terms of an output constraint vector,  $y_c$ , and a single polyhedral set inclusion,  $Y$ .
3. Augment the controlled process with Gilbert's variable bandwidth first-order low-pass filter,

and characterize an approximation to the maximal output admissible set in terms of a finite set of linear inequality constraints.

$$O_{\infty}^{\varepsilon} = \{x_g \in \mathfrak{R}^{n+1} : \Gamma_g x_g \leq \beta_g\}$$

4. Find the set,  $V$ , of extremal points of  $X_s^{\varepsilon}$  using the series of LPs

$$\max_{x_g} (c_i^T x_g) \quad \text{s. t.} \quad \Gamma_g x_g \leq \beta_g, \quad i = 1, 2, \dots, M$$

where  $c_i^T$  is obtained from the rows of the  $M \times (n+1)$  matrix  $\Gamma_g$ .

5. Generate subsets of  $V$  by only including vertices that are a specified Euclidean distance,  $d_i$ , from each other.
6. Characterize the sets  $X_s^{\delta_i} \subset X_s^{\varepsilon}$  in terms of linear inequality constraints using the recursive convex hull algorithm of Section 3.3.
7. Specify a criteria to determine the best allowable modified reference signal, such as the MDRSG or SRSG control concepts.
8. Perform a trade-off study to determine the most appropriate set,  $X_s^{\delta_i}$ , for on-line implementation of the reference signal governor. This is accomplished by comparing allowable steady state conditions and tracking performance for different subsets.

The resulting closed-loop system is globally BIBO stable. Moreover, Tracking performance is emphasized by not explicitly restricting the modified reference signal to statically admissible values. Instead, the controlled process state vector is constrained to a positively invariant set, viz., an arbitrarily close approximation to the maximal statically admissible set. The control law is nonlinear, but the nonlinearity is confined to the outer supervisory loop. Thus, it is sufficient to insure the modified reference signal does not result in immediate saturation, or drive the controlled process state out of the positively invariant set. On-line computational burden is reduced by implement-

ing the reference signal governor with the smallest subset,  $X_s^{\delta_i}$ , consistent with required system performance. While a trade-off exists between the volume of the subset and the number of linear inequalities, there is generally a threshold above which increasing the number of linear inequalities provides little improvement in volume of the subset, tracking performance, and allowable steady state conditions.

Specific criteria are specified by the designer to determine the “optimal” modified reference signal that satisfies the feasibility and admissibility constraints. Moreover, the reference signal governor may be implemented so that it is transparent in the case of small exogenous reference signals. Thus, small signal performance is preserved. Also, a graceful degradation in tracking performance is provided in the case of large inputs or large slewing maneuvers. Control concepts demonstrated here include the MDRSG and SRSG control concepts. More sophisticated concepts which include reference signal prediction and Receding Horizon control paradigms, such as LQT, are also possible. In the case of the LQT control paradigm any reference signal extrapolation scheme could be used in conjunction with one step ahead constraint enforcement to obtain a BIBO stable closed-loop system. Finally, the proposed methodology is applicable to tracking control systems subject to high amplitude dynamic reference signals, with both actuator displacement and actuator rate constraints, actuator dynamics, and open-loop unstable plants.

It should be noted that the maximal statically admissible set is a subset of the maximal positively invariant set. However, it is shown here that the maximal positively invariant set is not generally convex, and cannot be characterized by a finite set of inequality constraints. Thus, the maximal statically admissible set is the largest positively invariant set that is useful for our purpose.

The reference governor system architecture is also applied to the constrained regulator problem. This novel approach can substantially increase the size of the statically admissible set of initial states, for a given regulator control problem, without degrading small signal performance. More-

over, a more aggressive inner-loop linear control law may be implemented for a given predetermined initial set of states. In this case the outer-loop nonlinearity is a reference signal generator that produces a non-zero reference signal, as needed, to prevent saturation.

An additional contribution of this research is the development of a new computationally efficient recursive convex hull algorithm. Unlike existing algorithms the complete Hasse diagram, or facial graph, is not generated. The  $n$ -tope is built up simplicially with each  $(n - 1)$  dimensional facet containing exactly  $n$  vertices. The recursive algorithm only generates the  $(n - 1)$  dimensional facets and a listing of vertices (0-faces) included in each facet. Listings of all intermediate faces are not explicitly generated. A listing of all  $(n - 2)$ -faces are generated for invalidated facets, however, during each pass of the algorithm. Since the algorithm is recursive, it lends itself to on-line operation, although it is not employed in this manner here. Finally, the algorithm is applicable to cases where the set of points are not all vertices, and where the points are not in general position.

### 7.3 Research Conclusions

Regarding the constrained tracking control problem of eqs. (14)-(102), the following specific conclusions are drawn based on this research:

1. An arbitrarily close approximation to the maximal statically admissible set can be characterized with a finite set of inequality constraints for the stated problem.
2. The set,  $X_s^\varepsilon \subset \mathfrak{R}^n$ , inherits the property of positive invariance from the set  $O_\infty^\varepsilon \subset \mathfrak{R}^{n+1}$ .
3. Closed-loop system BIBO stability can be achieved using the dual loop reference governor methodology without restricting the modified reference signal to statically admissible values.
4. Removing the restriction of static admissibility from the modified reference signal allows greater flexibility in choosing the “optimal” modified reference signal from the set of allowable



reference signals. This allows use of aggressive control concepts such as the SRSG and LQT concepts which may provide significant improvements in tracking performance.

5. On-line computational burden can be substantially reduced with little or no impact on system performance by implementing the reference signal governor with polyhedral subsets,  $X_s^\delta \subset X_s^\varepsilon$ .
6. Positively invariant elliptical subsets result in the least on-line computational burden. However, a substantial loss in volume may be incurred with a like degradation in closed-loop system performance.
7. The maximal positively invariant set,  $X_I$ , is generally not convex, and thus cannot be characterized with a single set of linear inequality constraints.

Regarding the regulator control problem of eq. (78), the following conclusions are drawn:

1. For a given linear regulator control law, the size of the maximal statically admissible set may be significantly increased using the nonlinear dual loop reference signal generator methodology. This increases the robustness of the closed-loop system to external disturbances without sacrificing small signal performance.
2. The dual loop reference signal generator methodology can improve small signal performance by allowing a more aggressive inner-loop linear control law to be implemented for a given predetermined set of initial states,  $X_0$ .

Regarding the recursive convex hull algorithm, the following conclusions are drawn:

1. The convex hull of a set of points,  $V$ , may be built-up simplicially without generating a complete facial graph. Only a listing of vertices for each facet is required.
2. Given an initial polytope generated from a subset of  $V$ , each additional point is easily

incorporated by considering which  $(n - 1)$ -faces,  $(n - 2)$ -faces, and 0-faces (points) lie in the shadow of the associated supporting cone. Specifically, it is not necessary to consider other intermediate faces regardless of the dimension,  $n$ , of the problem.

3. Algorithm efficiency is improved if the initial polytope contains the origin. This is not strictly necessary however, and the algorithm is applicable to problems for which the origin is not an interior point of the convex hull.
4. Cases for which the points do not lie in general position are handled by eliminating points that lie in the interior of existing facets during the construction process, and by partitioning facets that are not simplexes into two or more simplexes. Thus, a simplicial polytope is constructed regardless of the geometric relationship between points.

### 7.3.1 Recommendations

The following areas are recommended for future research:

1. Demonstrate a complete MIMO extension of the proposed reference signal governor methodology. This requires on-line Linear Programming to determine the set of admissible reference signals at each time increment. It also requires on-line solution of convex optimization problems when the reference signal governor is based on a polyhedral subset,  $X_s^\delta \subset X_s^\varepsilon$ .
2. Investigate robustness issues. For example, can model uncertainty be addressed by constraining  $y_c$  such that  $y_c \in Y^\varepsilon \subset Y$ .
3. Investigate a BIBO stable LQT control concept with one-step ahead constraint enforcement. This is accomplished by choosing  $r'(t)$  at each time increment,  $t$ , such that  $y_c(t) \in Y$  and  $x(t + 1) \in X_s^\varepsilon$ .

## APPENDIX A - Recursive Convex Hull Algorithm Results

Results generated by the recursive convex hull algorithm for the fourth order problem of Chapter 5, using several subsets of vertices, are provided here. These results may be reproduced with the script, "edge4", provided in Appendix B.

### A.1 Recursive Convex Hull Algorithm and Brute Force Algorithm Results Comparison

In this section results obtained with both the recursive convex hull and brute force algorithms are presented. As noted in Chapter 5, these results were generated from a 14 element subset of vertices, Fig. 81. Results obtained with the recursive convex hull algorithm are shown in Fig. 82. The linear functionals that comprise the rows of  $\Gamma_s^{\delta_{14}}$  and the vertices contained in each facet of  $X_s^{\delta_{14}}$  are listed. Figure 83 shows the results obtained with the brute force algorithm. A careful inspection shows that the results obtained with both algorithms are consistent.

Vertices

1.5206	0.4416	2.7208	-2.9293	2.9293	-1.81
0.0049	3.6854	4.3563	-4.1820	4.1820	-5.65
0.4400	-0.0830	0.4351	-0.4267	0.4267	-0.37
-0.0101	0.3103	0.4006	-0.3799	0.3799	-0.59
-1.5206	-0.4416	-2.7208	2.6056	-2.6056	1.81
-0.0049	-3.6854	-4.3563	-5.3490	5.3490	5.65
-0.4400	0.0830	-0.4351	0.4284	-0.4284	0.37
0.0101	-0.3103	-0.4006	-0.5517	0.5517	0.59

Figure 81. Fourteen element subset of vertices

Linear Functionals				Vertices		
0.4165	0.8474	-3.9173	-8.4827	2	6	7
0.6801	-0.9916	-4.4167	8.4649	6	7	8
-0.0960	-0.5048	0.5433	3.0550	4	6	9
0.8814	-2.4293	-4.7586	23.1025	7	8	9
-0.3621	2.6592	2.8696	-27.0770	1	2	3
-0.3543	2.6467	2.8296	-26.9501	2	3	5
-0.1952	2.4019	1.9575	-24.5028	2	5	7
-0.4898	2.9327	3.2394	-30.0449	1	2	6
-0.1521	2.4322	1.3068	-25.0978	2	6	7
0.6423	-1.1504	-3.6098	9.5715	6	7	10
-0.0924	-0.5089	0.5261	3.0939	6	9	10
0.7401	-1.8521	-3.7707	16.7120	7	9	10
0.5102	0.1534	-4.0470	-0.4082	2	7	8
0.0995	1.0330	-2.8578	-9.9527	2	6	8
-0.1228	0.3067	-1.7479	-3.0987	4	6	10
-0.0494	0.2426	-2.1613	-2.4511	6	8	10
0.3543	-2.6467	-2.8296	26.9501	4	9	10
0.3621	-2.6592	-2.8696	27.0770	8	9	10
0.0494	-0.2426	2.1613	2.4511	1	3	11
-0.0995	-1.0330	2.8578	9.9527	1	9	11
0.1228	-0.3067	1.7479	3.0987	3	5	11
0.1911	-0.3916	1.3596	3.9424	5	7	11
0.5453	-1.9888	-0.8323	19.3987	7	9	11
0.0960	0.5048	-0.5433	-3.0550	2	3	5
0.0924	0.5089	-0.5261	-3.0939	2	3	12
0.3608	0.1483	-1.9632	0.4221	2	5	7
0.3894	0.0687	-2.2258	1.2571	2	7	12
0.9888	-3.2249	-4.9073	32.3591	7	8	9
0.7493	-1.5547	-4.4536	16.9670	7	8	12
0.4898	-2.9327	-3.2394	30.0449	8	9	12
-0.3608	-0.1483	1.9632	-0.4221	4	6	9
-0.7493	1.5547	4.4536	-16.9670	1	6	11
-0.5102	-0.1534	4.0470	0.4082	1	9	11
-0.3894	-0.0687	2.2258	-1.2571	6	9	11
-0.8814	2.4293	4.7586	-23.1025	1	2	3
-0.7401	1.8521	3.7707	-16.7120	2	3	12
-0.9888	3.2249	4.9073	-32.3591	1	2	6
-0.5453	1.9888	0.8323	-19.3987	2	6	12
-0.1911	0.3916	-1.3596	-3.9424	4	6	12
0.1952	-2.4019	-1.9575	24.5028	4	9	12
-0.6801	0.9916	4.4167	-8.4649	1	3	13
-0.6423	1.1504	3.6098	-9.5715	3	12	13
-0.4165	-0.8474	3.9173	8.4827	1	9	13
0.1521	-2.4322	-1.3068	25.0978	9	12	13

Figure 82. Recursive convex hull algorithm results, 14 element subset of vertices

-0.3621	2.6592	2.8696	-27.0770	1	2	3
-0.8814	2.4293	4.7586	-23.1025	1	2	3
-0.4898	2.9327	3.2394	-30.0449	1	2	6
-0.9888	3.2249	4.9073	-32.3591	1	2	6
0.0494	-0.2426	2.1613	2.4511	1	3	11
-0.6801	0.9916	4.4167	-8.4649	1	3	13
-0.7493	1.5547	4.4536	-16.9670	1	6	11
-0.0995	-1.0330	2.8578	9.9527	1	9	11
-0.5102	-0.1534	4.0470	0.4082	1	9	11
-0.4165	-0.8474	3.9173	8.4827	1	9	13
-0.3543	2.6467	2.8296	-26.9501	2	3	5
0.0960	0.5048	-0.5433	-3.0550	2	3	5
0.0924	0.5089	-0.5261	-3.0939	2	3	12
-0.7401	1.8521	3.7707	-16.7120	2	3	12
-0.1952	2.4019	1.9575	-24.5028	2	5	7
0.3608	0.1483	-1.9632	0.4221	2	5	7
0.4165	0.8474	-3.9173	-8.4827	2	6	7
-0.1521	2.4322	1.3068	-25.0978	2	6	7
0.0995	1.0330	-2.8578	-9.9527	2	6	8
-0.5453	1.9888	0.8323	-19.3987	2	6	12
0.5102	0.1534	-4.0470	-0.4082	2	7	8
0.3894	0.0687	-2.2258	1.2571	2	7	12
0.1228	-0.3067	1.7479	3.0987	3	5	11
-0.6423	1.1504	3.6098	-9.5715	3	12	13
-0.0960	-0.5048	0.5433	3.0550	4	6	9
-0.3608	-0.1483	1.9632	-0.4221	4	6	9
-0.1228	0.3067	-1.7479	-3.0987	4	6	10
-0.1911	0.3916	-1.3596	-3.9424	4	6	12
0.3543	-2.6467	-2.8296	26.9501	4	9	10
0.1952	-2.4019	-1.9575	24.5028	4	9	12
0.1911	-0.3916	1.3596	3.9424	5	7	11
0.6801	-0.9916	-4.4167	8.4649	6	7	8
0.6423	-1.1504	-3.6098	9.5715	6	7	10
-0.0494	0.2426	-2.1613	-2.4511	6	8	10
-0.0924	-0.5089	0.5261	3.0939	6	9	10
-0.3894	-0.0687	2.2258	-1.2571	6	9	11
0.8814	-2.4293	-4.7586	23.1025	7	8	9
0.9888	-3.2249	-4.9073	32.3591	7	8	9
0.7493	-1.5547	-4.4536	16.9670	7	8	12
0.7401	-1.8521	-3.7707	16.7120	7	9	10
0.5453	-1.9888	-0.8323	19.3987	7	9	11
0.3621	-2.6592	-2.8696	27.0770	8	9	10
0.4898	-2.9327	-3.2394	30.0449	8	9	12
0.1521	-2.4322	-1.3068	25.0978	9	12	13

Figure 83. Brute force algorithm results, 14 element subset of vertices

## **A.2 Results for Selected Subsets of Vertices.**

Results obtained for the 18, and 20 element subsets of vertices are provided here. Only the lists of vertices contained in each facet are provided. The associated linear functionals are easily generated using eq. (56) from Chapter 3.

vertices 1 through 6

1.5206e+000 4.4162e-001 2.7208e+000 -2.9293e+000 3.8229e+0  
4.9356e-003 3.6854e+000 4.3563e+000 -4.1820e+000 3.3111e+0  
4.4000e-001 -8.3014e-002 4.3513e-001 -4.2667e-001 2.6734e-001  
-1.0150e-002 3.1030e-001 4.0063e-001 -3.7988e-001 3.4041e-001

vertices 7 through 12

-1.9424e+000 -2.6056e+000 4.1618e+000 -1.5206e+000 -4.4162e-0  
-5.3558e+000 5.3490e+000 5.5195e-001 -4.9356e-003 -3.6854e+0  
-4.0269e-001 -4.2842e-001 2.9805e-001 -4.4000e-001 8.3014e-002  
-5.3640e-001 5.5169e-001 3.9452e-003 1.0150e-002 -3.1030e-001

vertices 13 through 18

2.6056e+000 2.9293e+000 -1.8420e+000 1.9424e+000 -3.8229e+0  
-5.3490e+000 4.1820e+000 -5.5948e+000 5.3558e+000 -3.3111e+  
4.2842e-001 4.2667e-001 -3.7904e-001 4.0269e-001 -2.6734e-001  
-5.5169e-001 3.7988e-001 -5.8396e-001 5.3640e-001 -3.4041e-001

Figure 84. 18 element subset of vertices

vertices 1 through 6

1.5206e+000 4.4162e-001 2.7208e+000 -2.1489e+000 2.0362e+000 -2.  
4.9356e-003 3.6854e+000 4.3563e+000 -4.9985e+000 5.1725e+000 5.  
4.4000e-001 -8.3014e-002 4.3513e-001 -4.2086e-001 4.1565e-001 -4.2  
-1.0150e-002 3.1030e-001 4.0063e-001 -4.8061e-001 5.0607e-001 5.51

vertices 7 through 12

-2.9293e+000 -7.7823e-001 3.6195e+000 4.1618e+000 -1.5206e+000 -4.  
-4.1820e+000 3.8991e+000 3.8458e+000 5.5195e-001 -4.9356e-003 -3.  
-4.2667e-001 -4.4000e-001 3.1908e-001 2.9805e-001 -4.4000e-001 8.30  
-3.7988e-001 3.8592e-001 4.0659e-001 3.9452e-003 1.0150e-002 -3.10

vertices 13 through 18

-2.7208e+000 7.7823e-001 -2.0362e+000 2.6056e+000 2.9293e+000 2.  
-4.3563e+000 -3.8991e+000 -5.1725e+000 -5.3490e+000 4.1820e+000 4  
-4.3513e-001 4.4000e-001 -4.1565e-001 4.2842e-001 4.2667e-001 4.20  
-4.0063e-001 -3.8592e-001 -5.0607e-001 -5.5169e-001 3.7988e-001 4.8

vertices 19 through 20

-3.6195e+000 -4.1618e+000  
-3.8458e+000 -5.5195e-001  
-3.1908e-001 -2.9805e-001  
-4.0659e-001 -3.9452e-003

Figure 85. 20 element subset of vertices



Facets 1 - 32

2 5 6 8  
 2 5 8 10  
 2 5 9 10  
 5 7 9 10  
 4 5 8 11  
 5 6 8 11  
 5 7 10 12  
 4 5 8 12  
 5 8 10 12  
 4 5 11 12  
 4 7 11 12  
 5 7 11 12  
 1 3 6 13  
 1 6 11 13  
 5 6 11 13  
 5 7 9 13  
 5 7 11 13  
 2 5 9 14  
 3 6 13 14  
 5 6 13 14  
 2 9 13 14  
 5 9 13 14  
 2 8 10 15  
 2 9 10 15  
 7 9 10 15  
 4 7 12 15  
 4 8 12 15  
 7 10 12 15  
 8 10 12 15  
 2 9 13 15  
 7 9 13 15  
 7 11 13 15

Facets 33 - 64

1 3 6 1  
 2 6 8 1  
 2 3 14 1  
 3 6 14 1  
 2 5 6 1  
 2 5 14 1  
 5 6 14 1  
 1 3 13 1  
 2 3 14 1  
 2 13 14  
 3 13 14  
 2 8 15 1  
 4 8 15 1  
 2 13 15  
 1 11 13  
 11 13 15  
 4 7 11 1  
 4 7 15 1  
 7 11 15  
 1 3 16 1  
 2 3 16 1  
 2 8 16 1  
 4 8 11 1  
 4 8 17 1  
 4 11 17  
 1 6 11 1  
 1 11 17  
 6 8 11 1  
 1 6 16 1  
 1 16 17  
 6 8 16 1  
 8 16 17

Figure 86. List of vertices / facet for the 18 element subset of vertices

Facets 1 - 43				Facets 44 - 85			
4	5	6	7	5	10	16	
1	3	5	9	1	3	9	1
2	6	7	9	2	6	9	1
5	6	7	9	1	9	14	1
2	5	7	10	2	3	17	1
5	7	8	10	2	9	17	1
5	7	8	11	3	9	17	1
4	5	6	12	1	3	14	1
4	5	7	13	4	12	14	
5	7	11	13	2	6	7	1
5	8	11	13	2	7	15	1
4	6	7	13	6	7	11	1
6	7	11	13	7	11	15	
4	5	12	13	4	6	13	1
4	8	12	13	4	13	15	
5	8	12	13	6	11	13	
4	8	12	14	11	13	15	
1	5	9	14	4	14	15	
5	9	12	14	2	10	15	
2	7	10	15	2	10	16	
7	8	10	15	10	15	16	
7	8	11	15	14	15	16	
4	8	13	15	2	3	17	1
8	11	13	15	3	14	17	
4	8	14	15	2	16	17	
8	12	14	15	14	16	17	
5	8	10	16	1	3	18	1
5	8	12	16	2	3	18	1
1	3	5	16	1	14	18	
1	3	14	16	2	6	18	1
1	5	14	16	5	6	9	2
5	12	14	16	5	6	12	2
8	10	15	16	5	9	12	2
8	12	15	16	9	12	14	
12	14	15	16	4	6	12	2
2	5	7	17	4	6	19	2
2	7	9	17	4	12	19	
3	5	9	17	12	14	19	
5	7	9	17	6	9	18	2
2	5	10	17	6	18	19	
2	10	16	17	9	14	18	
3	14	16	17	14	18	19	
3	5	16	17				

Figure 87. List of vertices / facet for the 20 element subset of vertices

## APPENDIX B - Selected Script Files

This appendix contains selected Matlab script files. Each script file contains a header which explains its intended use. Script files are provided in the order in which they are used in solving a problem.

### B.1 Script Files for the Second Order Problem

```
EX2LPM % Determines the maximal output admissible set for the fast
% overdamped second-order example. Implements algorithm 3.2
% of Gilbert's first paper.
%
%
clear

format short e

% Continuous time open-loop system with output = x1
a = [0 1;2 1];
b = [0;2];
c = [1 0];
d = 0;

% State feedback matrix
kx = [-33 -10.5];

% Compute kr so that we have perfect tracking
ac=a+b*kx;
aci=inv(ac);
kr=-1/(c*aci*b);
```

```

bc = b*kr;

cc = c;

dc = d;

eig(ac)

% Equivalent open-loop discrete time system with output equal to x1
[ad,bd,cd,dd]=c2dm(a,b,c,d,0.01,'zoh');

% form closed-loop discrete-time and augmented system dynamics

% matrices, and output matrix associated with constraints, Cg

acl=ad+bd*kx;

bcl=bd*kr;

ag = [1 0 0;bcl acl];

cg = [kr kx];

H0 = kr + kx*(inv(eye(2) - acl))*bcl;

% Compute static admissibility constraints and specify epsilon.

epsi = 0.05;

brs = (1 - epsi)

% set up linear programming problem

t = -1;

lim = [1;1];

% To obtain t_star use (while all), to obtain t_i_star use (while any)

while any(lim > 0)

%while all(lim > 0)

t = t + 1;

% Construct f_i(ca^(t+1)) for i=1,2 Neglecting the bias term

```

```

% We can ignore the bias term for now since maximizing  $f_i(y_c)$ 
% without the bias term gives the same result. But we will use
% the bias term later in determining the actual max value achieved
% for each  $f_i(y_c)$ .

f1 = cg*ag^(t+1);
f2 = -cg*ag^(t+1);

% lp finds the min, but we want the max. Also, lp needs
% f as a column vector rather than a row vector. So negate
% and transpose each of the fi

f1 = -f1';
f2 = -f2';

% Now build the constraint matrix and vector. The first two rows
% of AGC and BGC are associated with the static admissibility
% constraints on rprime.

AGC = [1 0 0;-1 0 0];
BGC = [brs;brs];

for k = 0:t;

AGC = [AGC;cg*ag^k];
BGC = [BGC;1];

end

for k = 0:t;

AGC = [AGC;-cg*ag^k];
BGC = [BGC;1];

end

```

```

% Solve the linear programs
X1 = lp(f1,AGC,BGC);
X2 = lp(f2,AGC,BGC);

% compute the max value of each of the  $f_i(ca^{(t+1)x})$  obtained.

% If all max values are less than zero we are done, and t_star is
% equal to the current value of t.

lim(1) = -f1'*X1 - 1
lim(2) = -f2'*X2 - 1

end

tstar = t

VERT2.M

% Find the vertices of Xs for example 2, using the
% 60 rows of the constraint matrix, AGC, as the linear functionals.
% Must either run ex2lp.m to generate the
% AGC matrix and BGC vector, or load ex2gam.mat.

format short e

% Delete the first and second rows of AGC and BGC since these deal
% with the statically admissible r-prime constraints

Ag = [AGC(3:60,:)];
Bg = [BGC(3:60)];

for k = 1:58

c = Ag(k,:)'

c(1)=0;

% lp finds the min, but we want the max. Thus, negate c.

```

```

c = -c;

% Note: Use the complete set of constraints in AGC and BGC

% when performing the LP

xgstar(:,k)=lp(c,AGC,BGC);

end

EDGE2.M

% Constructs Xs for example 2 using the

% recursive construction algorithm. Needs xgstar which was generated

% with vert2.m, and is also stored in ex2gam.mat.

format short e

clear

load ex2gam

% get rid of the r-prime elements of xgstar

vta = xgstar(2:3,:);

% save only unique vertices of Xs which

vt(:,1) = vta(:,1);

l = 1;

for k = 2:58;

for I = 1:l;

% testxg(:,I) = vta(:,k) - vt(:,I);

testxg(I) = sqrt((vta(1,k) - vt(1,I))^2 + (vta(2,k) - vt(2,I))^2);

end

% testm = min(max(abs(testxg)));

testm = min(testxg);

```

```

clear testxg;

if testm > .001;

l = l+1;

vt(:,l) = vta(:,k);

end

end

ys=size(vt);

numv = ys(2)

numv2 = ys(2)/2;

% generate first n+1 = 3 hyperplanes using the first n+1 = 3

% vertices, each hyperplane contains n = 2 vertices

% a1, a2, a3 are row vectors and are actually a1', a2', a3'

% in my notes

a1 = [1 1]*inv([vt(:,1) vt(:,2)]);

a2 = [1 1]*inv([vt(:,1) vt(:,3)]);

a3 = [1 1]*inv([vt(:,2) vt(:,3)]);

Ak = [a1;a2;a3];

Bk = [1;1;1];

% keep track of which vertices are in which hyperplanes.

% Numbers in h refer to columns of vt. Each row of h

% is associated with the hyperplane of the corresponding rows

% of Ak and Bk.

h = [1 2;1 3;2 3];

% Insure vectors are outward pointing

```



```

clear ch

ch(1) = a1*vt(:,3);

ch(2) = a2*vt(:,2);

ch(3) = a3*vt(:,1);

for I = 1:3;

if ch(I) > 1

Ak(I,:) = -Ak(I,:);

Bk(I) = -Bk(I);

end

end

% Now construct Xs recursively by adding remaining vertices

% one at a time.

% n = no. of states, k = number of vertices included,

% mk*n = dimensions of Ak

n = 2;

k = 3;

mk = (k - n)*(n - 1) + 2;

%k1 = 4;

for k1 = 4:numv;

ckplane = Ak*vt(:,k1);

Jerror = 0;

for L2 = 1:mk;

if ckplane(L2) > Bk(L2);

J = L2;

```

```

Jerror = Jerror + 1;

% if Jerror == 1

% jtest = J

% end

end

end

if Jerror ~= 1

error('none or more than one hyperplane failed for new point')

end

% Now update Ak, Bk, and h

for L3 = 1:(J-1);

Ak1(L3,:) = Ak(L3,:);

Bk1(L3,:) = Bk(L3,:);

h1(L3,:) = h(L3,:);

end

for L3 = J:(mk-1);

Ak1(L3,:) = Ak(L3+1,:);

Bk1(L3,:) = Bk(L3+1,:);

h1(L3,:) = h(L3+1,:);

end

% Generate n = 2 new hyperplanes and append new vertices to h

% This code must be modified for problems with n ~= 2

% Also verify outward pointing vectors

Ak1(mk,:) = [1 1]*inv([vt(:,h(J,1)) vt(:,k1)]);

```

```

Ak1(mk + 1,:) = [1 1]*inv([vt(:,h(J,2)) vt(:,k1)]);
Bk1 = [Bk1;1;1];
ch1 = Ak1(mk,:)*vt(:,h(J,2));
if ch1 > Bk1(mk)
    Ak1(mk,:) = -Ak1(mk,:);
    Bk1(mk) = -Bk1(mk);
end
ch2 = Ak1(mk + 1,:)*vt(:,h(J,1));
if ch2 > Bk1(mk + 1)
    Ak1(mk + 1,:) = -Ak1(mk + 1,:);
    Bk1(mk + 1) = -Bk1(mk + 1);
end
k = k + 1;
mk = (k - n)*(n - 1) + 2;
Ak = Ak1;
Bk = Bk1;
h = [h1;h(J,1) k1;h(J,2) k1];
clear Bk1;
clear Ak1;
clear h1;
end

```

## B.2 Script Files for the Fourth Order Problem

**FOURLPM** % Determines the maximal output admissible set for the 4th-

```
% order f-16 model. Implements algorithm 3.2 of Gilbert's
```

```
% first paper.
```

```
%
```

```
%
```

```
%
```

```
%
```

```
%
```

```
clear
```

```
format short e
```

```
% F16 continuous time system and output constraint
```

```
ac = [-1.15 .9937 -0.177 0;3.724 -1.26 -19.5 0;
```

```
0 0 -20 0;0 -1 0 0];
```

```
bu = [0;0;20;0];
```

```
br = [0;0;0;1];
```

```
bc = [bu br];
```

```
kx=[0.3509 1.1373 -0.9184 -6.6107];
```

```
kr = -.6
```

```
cc=[0 0 1 0;0 0 -20 0];
```

```
dc=[0 0;20 0];
```

```
[ad,bd,cd,dd]=c2dm(ac,bc,cc,dc,0.01,'zoh');
```

```
% Closed-loop discrete-time system and output constraints
```

```
adcl = ad + (bd(:,1))*kx;
```

```
bdcl = bd(:,1)*kr + bd(:,2);
```

```
ccdcl = [0 0 1 0;ad(3,:)-[0 0 1 0] + bd(3,1)*kx];
```

```

dcdcl = [0;bd(3,1)*kr + bd(3,2)];

% Augmented system and H0. Note H0(2)=0 because the actuator rate is zero
% in steady state. Thus, get rid of garbage in H0(2) by setting it to zero.

Ag = [1 0 0 0 0;bdcl adcl];

Bg = [1;0;0;0;0];

Cg = [dcdcl ccdcl];

H0 = dcdcl + ccdcl*inv(eye(4)-adcl)*bdcl;

H0(2)=0;

% Compute static admissibility constraints and specify epsilon.

% Note, for this problem H0 = [h;0] because the second row
% of H0 is associated with the elevator rate constraints, and
% the elevator rate is zero in steady state.

epsi = 0.05;

brs = (0.44 - epsi)/H0(1);

% set up LP

t = -1;

lim = [1;1;1;1];

%while any(lim > 0)

while all(lim > 0)

t = t + 1;

% Construct fi(ca^(t+1)) for i=1,2,3,4 Neglecting the bias term

% We can ignore the bias term for now since maximizing fi(yc)

% without the bias term gives the same result. But we will use

% the bias term later in determining the actual max value achieved

```

```

% for each fi(yc).

f1 = Cg(1,:)*Ag^(t+1);

f2 = Cg(2,:)*Ag^(t+1);

f3 = -Cg(1,:)*Ag^(t+1);

f4 = -Cg(2,:)*Ag^(t+1);

% lp finds the min, but we want the max. Also, lp needs
% f as a column vector rather than a row vector. So negate
% and transpose each of the fi

f1 = -f1';

f2 = -f2';

f3 = -f3';

f4 = -f4';

% Now build the constraint matrix and vector. The first two rows
% of AGC and BGC are associated with the static admissibility
% constraints on rprime.

AGC = [1 0 0 0 0;-1 0 0 0 0];

BGC = [brs;brs];

for k = 0:t;

AGC = [AGC;Cg*Ag^k];

BGC = [BGC;0.44;.01];

end

for k = 0:t;

AGC = [AGC;-Cg*Ag^k];

BGC = [BGC;0.44;.01];

```

```

end

% Solve the linear programs

X1 = lp(f1,AGC,BGC);
X2 = lp(f2,AGC,BGC);
X3 = lp(f3,AGC,BGC);
X4 = lp(f4,AGC,BGC);

% compute the max value of each of the  $f_i(ca^{(t+1)x})$  obtained.

% If all max values are less than zero we are done, and t_star is
% equal to the current value of t.

lim(1) = -f1'*X1 - .44;
lim(2) = -f2'*X2 - .01;
lim(3) = -f3'*X3 - .44;
lim(4) = -f4'*X4 - .01;

end

tstar = t

VERT4.M

% Find the vertices of Xs for the 4th-order system, using the
% 174 rows of the constraint matrix AGC as the linear functionals
% AGC was generated with four2lp.m

clear

format short e

load xg4

% Delete the first two rows of AGC since these deal with
% the static admissibility of rprime, and lead to

```

```

% c_i = 0.

AG = AGC(3:174,:);

BG = BGC(3:174);

% Find the vertex associated with each row of AGC. Also, normalize
% each linear functional.

for k = 1:172;

c = AG(k,:);

c(1) = 0;

% Normalize the linear functionals

c = c/sqrt(c(2)^2 + c(3)^2 + c(4)^2 + c(5)^2);

% lp finds the min, and we want the max. Also, lp
% wants c as a column vector. So, negate and transpose c.

c = -c';

% Note: Use the complete set of constraints in AGC and BGC
% when performing the LP.

xgstar(:,k) = lp(c,AGC,BGC);

end

EDGE4.M

% Constructs Xs for the fourth-order f-16 system using the edge
% recursive construction algorithm. Needs xgstar which was generated
% with vert4.m, and is also stored in xg4.mat.

%

clear

format short e

```



```

load xg4

% get rid of the r-prime elements of xgstar

verts = xgstar(2:5,:);

% save only unique vertices of Xs, or thin the set of

% vertices.

vt(:,1) = verts(:,1);

l = 1;

% There are 172 elements in verts, but to enforce

% symmetry, only process the first half

% Then, append the negative of the selected

% vertices. The constraints are symmetric, so the vertices

% in xgstar and verts are also symmetric. Also, the vertices have

% been ordered in verts such that vertices 87 - 172 are the negative

% of vertices 1 - 86.

for k = 2:86;

for I = 1:l;

testxga = (verts(1,k)-vt(1,I))^2 + (verts(2,k) - vt(2,I))^2;

testxgb = (verts(3,k)-vt(3,I))^2 + (verts(4,k) - vt(4,I))^2;

testxg(I) = sqrt(testxga + testxgb);

end

testm = min(testxg);

clear testxg;

clear testxga;

clear testxgb;

```

```

if testm > 1.5;

l = l+1;

vt(:,l) = verts(:,k);

end

end

vt = [vt -vt];

ys=size(vt);

numv = ys(2)

% rearrange vt so that the origin is included in the initial

% polyhedron.

vttemp = vt;

% rearrangement for 60 vertices, d = 0.1

%vttemp(:,2)=vt(:,10);vttemp(:,3)=vt(:,20);vttemp(:,4)=vt(:,30);vttemp(:,5)=vt(:,60);

%vttemp(:,10)=vt(:,2);vttemp(:,20)=vt(:,3);vttemp(:,30)=vt(:,4);vttemp(:,60)=vt(:,5);

% rearrangement for 36 vertices, d = 0.5

% vttemp(:,5)=vt(:,11);vttemp(:,11)=vt(:,5);

% rearrangement for 20 and 18 vertices, d = 1.0 and 1.5 respectively

vttemp(:,4)=vt(:,5);vttemp(:,5)=vt(:,8);vttemp(:,8)=vt(:,4);

% rearrangement for 28 vertices, d = 0.7

%vttemp(:,4)=vt(:,5);vttemp(:,5)=vt(:,10);vttemp(:,10)=vt(:,4);

% rearrangement for 14 vertices, d = 1.8

%vttemp(:,4)=vt(:,5);vttemp(:,5)=vt(:,12);vttemp(:,12)=vt(:,4);

vt = vttemp;

% generate first n+1 = 5 hyperplanes using the first n+1 = 5

```

```

% vertices, each hyperplane contains n = 4 vertices

% a1, a2, a3, a4, a5 are row vectors and are actually a1', a2', a3',
% a4' and a5' in my notes

a1 = [1 1 1 1]*inv([vt(:,1) vt(:,2) vt(:,3) vt(:,4)]);
a2 = [1 1 1 1]*inv([vt(:,1) vt(:,2) vt(:,3) vt(:,5)]);
a3 = [1 1 1 1]*inv([vt(:,1) vt(:,2) vt(:,4) vt(:,5)]);
a4 = [1 1 1 1]*inv([vt(:,1) vt(:,3) vt(:,4) vt(:,5)]);
a5 = [1 1 1 1]*inv([vt(:,2) vt(:,3) vt(:,4) vt(:,5)]);

Ak = [a1;a2;a3;a4;a5];

Bk = [1;1;1;1;1];

% keep track of which vertices are in which hyperplanes.

% Numbers in h refer to columns of vt. Each row of h
% is associated with the hyperplane of the corresponding rows
% of Ak and Bk.

h = [1 2 3 4;1 2 3 5;1 2 4 5;1 3 4 5;2 3 4 5];

% Insure vectors are outward pointing

clear ch

ch(1) = a1*vt(:,5);

ch(2) = a2*vt(:,4);

ch(3) = a3*vt(:,3);

ch(4) = a4*vt(:,2);

ch(5) = a5*vt(:,1);

for I = 1:5;

if ch(I) > 1

```

```

Ak(I,:) = -Ak(I,:);
Bk(I) = -Bk(I);

end

end

Bk

if any(Bk < 0)

Ak

ch

error('Bk error')

end

% Now construct Xs recursively by adding remaining vertices
% one at a time.

% n = no. of states, k = number of vertices included,

% mk x n = dimensions of Ak

n = 4;

k = 5;

mk = 5;

nbf = 0;

k1f = 0;

failnew(1:5,1:2) = zeros(5,2);

edges(1:5,1) = zeros(5,1);

Q1 = 0; Q2 = 0;

for k1 = 6:numv;

% Check to see if v(k+1) is contained in an existing hyperplane

```

```

% Also, compute condition numbers of the matrices M_k_i

hsize = size(h);

for J = 1:hsize(1)

va = (vt(:,h(J,1)) - vt(:,h(J,2)))/norm(vt(:,h(J,1)) - vt(:,h(J,2)));

vb = (vt(:,h(J,1)) - vt(:,h(J,3)))/norm(vt(:,h(J,1)) - vt(:,h(J,3)));

vc = (vt(:,h(J,1)) - vt(:,h(J,4)))/norm(vt(:,h(J,1)) - vt(:,h(J,4)));

vd = (vt(:,h(J,1)) - vt(:,k1))/norm(vt(:,h(J,1)) - vt(:,k1));

psi(J) = cond([va vb vc vd]);

end

[maxpsi(k1), hrow(k1)] = max(psi);

failnew(k1,1) = 0;

ckplane = Ak*vt(:,k1);

nf = 0;

clear hf

clear row

for I = 1:mk;

if ckplane(I) - Bk(I) > 1e-14;

% if ckplane(I) - Bk(I) > 0;

failnew(k1,1) = failnew(k1,1) + 1;

nf = nf + 1;

hf(nf,:) = h(I,:);

row(nf) = I;

Q1 = Q1 + 1;

storold(Q1,:) = h(I,:);

```

```

end

end

if nf < 1

error('No hyperplanes failed for new vertex')

end

% Create the initial h1, Ak1, and Bk1 matrices by deleting the rows
% of h, Ak, and Bk that correspond to failed hyperplanes

mk1 = mk - nf;

for J = 1:nf

if J == 1

if row(J) == 1

h1 = h(2:mk,:);

Ak1 = Ak(2:mk,:);

Bk1 = Bk(2:mk,:);

else

h1 = h(1:(row(J)-1),:);

h1 = [h1;h((row(J)+1):mk,:)];

Ak1 = Ak(1:(row(J)-1),:);

Ak1 = [Ak1;Ak((row(J)+1):mk,:)];

Bk1 = Bk(1:(row(J)-1),:);

Bk1 = [Bk1;Bk((row(J)+1):mk,:)];

end

end

if J > 1

```

```

if row(J) == J
h1 = h1(2:(mk-J+1),:);
Ak1 = Ak1(2:(mk-J+1),:);
Bk1 = Bk1(2:(mk-J+1),:);
elseif row(J) == mk
h1 = h1(1:(row(J)-J),:);
Ak1 = Ak1(1:(row(J)-J),:);
Bk1 = Bk1(1:(row(J)-J),:);
elseif row(J) ~= mk
h1 = h1(1:(row(J)-J),:);
h1 = [h1;h((row(J)+1):mk,:)];
Ak1 = Ak1(1:(row(J)-J),:);
Ak1 = [Ak1;Ak((row(J)+1):mk,:)];
Bk1 = Bk1(1:(row(J)-J),:);
Bk1 = [Bk1;Bk((row(J)+1):mk,:)];
end
end
end

% Generate a list of all subfacets associated with failed facets
clear ha
clear hb
hfsort = sort(hf')';
I1 = 1;
I2 = 1;

```

```

while I1 <= nf
    ha(I2,:) = hfsort(I1,1:3);
    I2 = I2 + 1;
    ha(I2,:) = [hfsort(I1,1:2) hfsort(I1,4)];
    I2 = I2 + 1;
    ha(I2,:) = [hfsort(I1,1) hfsort(I1,3:4)];
    I2 = I2 + 1;
    ha(I2,:) = hfsort(I1,2:4);
    I2 = I2 + 1;
    I1 = I1 + 1;
end

% Find unshared subfacets (subfacets that only appear once in the list
% of subfacets)
hb(1:4,:) = ha(1:4,:);
J1 = 4;
for I = 5:(4*nf)
    clear ht
    for J = 1:J1
        ht1 = abs(ha(I,1) - hb(J,1)) + abs(ha(I,2) - hb(J,2));
        ht2 = abs(ha(I,3) - hb(J,3));
        ht(J) = ht1 + ht2;
        if ht(J) == 0
            if J ~= 1
                hb1(1:J-1,:) = hb(1:J-1,:);

```



```

hb1(J:J1-1,:) = hb(J+1:J1,:);

else

hb1(1:J1-1,:) = hb(2:J1,:);

end

J1 = J1 - 1;

end

end

if all(ht ~= 0)

J1 = J1 + 1;

hb(J1,:) = ha(I,:);

else

hb = hb1;

clear hb1

end

end

% Create new hyepeplanes from vertices in unshared subfacets and the new vertex

numedg = size(hb);

edges(k1,1) = numedg(1);

failnew(k1,2) = 0;

for I = 1:edges(k1,1)

a = [1 1 1 1]*inv([vt(:,hb(I,1)) vt(:,hb(I,2)) vt(:,hb(I,3)) vt(:,k1)]);

Ak1 = [Ak1;a];

Bk1 = [Bk1;1];

h1 = [h1;hb(I,1) hb(I,2) hb(I,3) k1];

```

```

mk1 = mk1 + 1;

Q2 = Q2 + 1;

stornew(Q2,:) = [hb(I,1) hb(I,2) hb(I,3) k1];

failnew(k1,2) = failnew(k1,2) + 1;

end

Ak = Ak1;

Bk = Bk1;

h = h1;

mk = mk1;

clear Ak1

clear Bk1

clear h1

end

yxs = Ak*inv(eye(4)-adcl)*bdcl;

rssmax = 1/max(yxs)

rssmin = 1/min(yxs)

hsize = size(h);

```

## Bibliography

- [1] Astrom, K. J. and L. Rundquist. "Integrator Windup and How to Avoid It," *Proceedings of the 1989 American Conference*, Vol. 2, pp. 1693–1698, 1989.
- [2] Athans, M. and P. L. Falb. *Optimal Control, An Introduction to the Theory and its Applications*. New York: McGraw-Hill, 1966.
- [3] Benzaouia, Abdellah and Christian Burgat. "The Regulator Problem for a Class of Linear Systems with Constrained Control," *Systems and Control Letters*, Vol. 10, pp. 357–363, 1988.
- [4] Benzaouia, Abdellah and Christian Burgat. "Regulator Problem for Linear Discrete-Time Systems with Nonsymmetrical Constrained Control," *Int. J. Control*, Vol. 48, No. 6, pp. 2441–2451, 1988.
- [5] Bitsoris, Georges. "On the Positive Invariance of Polyhedral Sets for Discrete-Time Systems," *Systems and Control Letters*, Vol. 11, pp. 243–248, 1988.
- [6] Chandler, P. R., M. J. Mears and M. Pachter. "A Hybrid LQR/LP Approach for Addressing Actuator Saturation in Feedback Control," *1994 Conference on Decision and Control*, pp. 3860–3867, Dec 14–16, 1994.
- [7] Frankena, J. F. and R. Sivan. "A nonlinear optimal control law for linear systems," *International Journal of Control*, Vol. 30, No. 1, pp. 159–178, 1979.
- [8] Gilbert, E. G. and K. T. Tan. "Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets," *IEEE Trans. on Automatic Control*, Vol. AC-36, No. 9, pp. 1008–1020, September, 1991.
- [9] Gilbert, E. G., I. Kolmanovsky and K. T. Tan. "Nonlinear Control of Discrete-Time Linear Systems with State and Control Constraints: A Reference Governor with Global Convergence Properties," *Proceedings of the 33rd Conference on Decision and Control*, pp. 144–149, December, 1994.
- [10] Gutman, Per-Olof and Per Hagander. "A New Design of Constrained Controllers for Linear Systems," *IEEE Trans. on Automatic Control*, Vol. AC-30, No. 1, pp. 22–33, January, 1985.
- [11] Kapasouris, P., M. Athans and G. Stein. "Design of Feedback Control Systems for Stable Plants with Saturating Actuators," *Proceedings of the 27th Conference on Decision and Control, Austin, TX*, pp. 469–479, December, 1988.
- [12] Kapasouris, P. and M. Athans. "Multivariable Control Systems with Saturating Actuators Antireset Windup Strategies," *Proceedings of the American Control Conference, Boston, MA*, pp. 1579–1584, 1985.
- [13] Kirk, D. E. *Optimal Control Theory, An Introduction*. Englewood Cliffs, New Jersey: Prentice-Hall, 1970.
- [14] Krikelis, N. J. and S. K. Barkas. "Design of Tracking Systems Subject to Actuator Saturation and Integrator Windup," *International Journal of Control*, Vol. 39, pp. 667–683, 1984.
- [15] Lewis, Frank. 199.
- [16] McNamee, Joe and Meir Pachter. "Nonlinear tracking controllers for actuator saturation

- effects mitigation," *Proceedings of the 36th Conference on Decision and Control*, Vol. 5, pp. 4529–4534, Dec 10-12, 1997.
- [17] Miller, R. B. *A New Approach to Manual Tracking Flight Control with Amplitude and Rate Constrained Dynamic Actuators*. PhD dissertation, Air Force Institute of Technology, March 1997.
- [18] Miller, R. B. and M. Pachter. "Manual Tracking Control with Amplitude and Rate Constrained Actuators," *Proceedings of the 1996 Conference on Decision and Control*, pp. 3159–3164, December, 1996.
- [19] Miller, R.B. and M. Pachter. "Maneuvering Flight Control with Actuator Constraints," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 20, July - August, 1997.
- [20] Pachter, M. and D. H. Jacobson. "Observability with a Conic Observation Set," *IEEE Trans. on Automatic Control*, Vol. AC-24, No. 4, pp. 632–633, August, 1979.
- [21] Pachter, Meir and R.B. Miller. "Manual Flight Control with Saturating Actuators," *IEEE Control Systems Magazine*, Vol. 18, No. 1, pp. 10–19, Feb, 1998.
- [22] Pachter, M., P. R. Chandler and M. Mears. "Control Reconfiguration with Actuator Rate Saturation," *Proceedings of the American Control Conference*, June, 1995.
- [23] Popov, V. M. *The Hyperstability of Control Systems*. Springer, 1973.
- [24] Preparata, Franco P. and Michael Ian Shamos. *Computational Geometry*. 175 Fifth Avenue, New York, New York: Springer-Verlag, 1985.
- [25] Ryan, E. P. "Optimal Feedback Control of Saturating Systems," *International Journal of Control*, Vol. 35, pp. 521–534, 1982.
- [26] Vassilaki, M., J. C. Hennet and G. Bitsoris. "Feedback Control of Linear Discrete-Time Systems under State and Control Constraints," *Int. J. Control*, Vol. 47, No. 6, pp. 1727–1735, 1988.
- [27] Zhou, K., J. C. Doyle and K. Glover. *Robust and Optimal Control*. Upper Saddle River, New Jersey: Prentice-Hall, 1996.

## Vita

Major Joseph W. McNamee was born in Pendleton, Oregon on Aug 6, 1955, and graduated from David Douglas High School, Portland, Oregon in 1973. His preliminary undergraduate work was accomplished at Mt. Hood Community College, Gresham Oregon. Joe enlisted in the United States Air Force in 1976 and was stationed at Rickenbacker AFB, Ohio; Izmir, Turkey; and Grand Forks AFB, North Dakota. He left the Air Force in December 1981 to attend The Ohio State University in Columbus, Ohio, where he met his wife, Cindy. He graduated with a B.S.E.E. in December 1984, entered the Air Force's Officer Training School in January 1985, and was commissioned as a Second Lieutenant on April 5, 1985. Upon graduation from OTS Joe was assigned to the F-16 System Program Office as a radar systems engineer at Wright-Patterson AFB, Ohio from April 1985 to May 1988. In May 1988 he was reassigned to the Air Force Institute of Technology, receiving his M.S.E.E. in December 1989. He was then assigned to the Air Force Flight Test Center at Edwards, AFB, California where he served as the lead avionics test engineer for the 6519th Flight Test Squadron on the Advance Cruise Missile flight test program until 1991, and then as the Chief B-1B Flight Test Engineer for the 419th Flight Test Squadron until May 1995, at which time he was reassigned to the Air Force Institute of Technology as a Ph.D. student. Upon graduation Joe will be assigned to the Air Force Research Labs at Wright-Patterson AFB, Ohio.

Permanent Address: 525 Orchard Drive, Springboro, Ohio 45066

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1998	3. REPORT TYPE AND DATES COVERED Dissertation 1998	
4. TITLE AND SUBTITLE High Amplitude Tracking Control		5. FUNDING NUMBERS	
6. AUTHOR(S) Joseph W. McNamee			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Wright-Patterson AFB, OH 45433		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Mr. Duane Rubertus AFRL/VACS 2210 8th Street, Rm. 225 WPAFB, OH 45430-7521		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION AVAILABILITY STATEMENT Distribution Unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The problem of tracking control in the face of high amplitude dynamic reference signals, hard state and control constraints, and open-loop unstable plants is investigated. Saturation effects mitigation is a current research topic, and a large body of literature exists which addresses this problem. However, with a few notable exceptions, little work has been accomplished which is applicable to the problem of high amplitude dynamic reference signal tracking in the face of hard state and control constraints, and open-loop unstable plants. Moreover, methods that do address the stated problem generally limit the exogenous reference signal to statically admissible values to obtain a BIBO stable closed-loop system. In these investigations a nonlinear reference signal governor generates a modified reference signal based on the exogenous reference signal and the state of the controlled process. This research departs from previous work in that a dual-loop reference signal governor methodology is developed that prevents constraint violation but does not restrict the modified reference signal to statically admissible values. Thus tracking performance is improved. An arbitrarily close approximation of the controlled process' maximal statically admissible set is characterized. On-line computational burden is generally less than that imposed by current reference signal governor methods, and the resulting closed-loop system is BIBO stable.			
14. SUBJECT TERMS Actuator Saturation, Tracking Control, Positively Invariant Sets, Constrained Control, Hard Constraints, Open-loop Unstable Systems, Reference Signal Governor, Nonlinear Control, Convex Hull		15. NUMBER OF PAGES	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL