A Model-Free Control Algorithm Derived Using the Sliding Model Control Method

Arielle Mizov

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
A Model-Free Control Algorithm Derived Using the Sliding Model Control Method

by

Arielle Mizov

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Supervised by

Associate Professor Dr. Agamemnon Crassidis
Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
May 2015

Approved by:

Dr. Agamemnon Crassidis, Associate Professor
_Thesis Advisor, Department of Mechanical Engineering_

Dr. Jason Kolodziej, Associate Professor
_Committee Member, Department of Mechanical Engineering_

Dr. Mark Kempski, Professor
_Committee Member, Department of Mechanical Engineering_

Dr. Wayne Walter, Professor
_Department Representative, Department of Mechanical Engineering_
Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

A Model-Free Control Algorithm Derived Using the Sliding Model Control Method

I, Arielle Mizov, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

__________________________
Arielle Mizov

__________________________
Date
ABSTRACT

In this work, a model-free sliding mode control scheme is derived and applied to linear and nonlinear systems that is solely based on observable measurements and therefore does not require a theoretical system model in developing the controller form. The general sliding mode controller form is derived for an $n^{th}$-order system and is strictly limited to a single-input unit input influence gain case for this work. The controller form is based solely on system measurements assuming the order of the system is known. The switching gain form is developed so that stability of the closed-loop sliding mode controller system is guaranteed using Lyapunov’s Direct Method. The controller form is reformulated using a smoothing moving boundary layer to eliminate chattering of the control effort. A simulation study is presented for a single-input unit input influence gain case applied to both a linear and nonlinear system with and without a smoothing boundary layer. The measurement based controller form is shown to be identical regardless of the system’s kinematics to be controlled assuming the order is known. Results of the simulation efforts show good state tracking performance is achieved with stable convergence for the tracking performance regardless of the system to be controlled.
# Table of Contents

ABSTRACT

TABLE OF FIGURES:

NOMENCLATURE

1 INTRODUCTION

2 THE SLIDING MODE CONTROL METHOD

3 SYSTEM DESCRIPTION

4 APPLICATION OF MODEL-FREE SMC TO LINEAR SYSTEMS

5 APPLICATION OF MODEL-FREE SMC TO NONLINEAR SYSTEMS
TABLE OF FIGURES:

Figure 1: Position Tracking Error for Model-Based Nonlinear System ................................................................. 27
Figure 2: Position Comparison for Model-Based Nonlinear System ........................................................................ 27
Figure 3: Velocity Tracking Error for Model-Based Nonlinear System ................................................................. 28
Figure 4: Velocity Comparison for Model-Based Nonlinear System ................................................................. 28
Figure 5: Acceleration Tracking Error for Model-Based Nonlinear System .......................................................... 28
Figure 6: Acceleration Comparison for Model-Based Nonlinear System ......................................................... 28
Figure 7: Sliding Condition for Model-Based Nonlinear System ........................................................................ 29
Figure 8: Control Effort for Model-Based Nonlinear System ........................................................................ 29
Figure 9: Phase Plane for Model-Based Nonlinear System ........................................................................ 29
Figure 10: Phase Plane of Sliding System for Model-Based Nonlinear System ................................................ 29
Figure 11: Position Tracking Error for Model-Based Nonlinear System with Moving Boundary Layer ........... 31
Figure 12: Position Comparison for Model-Based Nonlinear System with Moving Boundary Layer ............. 31
Figure 13: Velocity Tracking Error for Model-Based Nonlinear System with Moving Boundary Layer .......... 32
Figure 14: Velocity Comparison for Model-Based Nonlinear System with Moving Boundary Layer .......... 32
Figure 15: Acceleration Tracking Error for Model-Based Nonlinear System with Moving Boundary Layer .... 32
Figure 16: Acceleration Comparison for Model-Based Nonlinear System with Moving Boundary Layer ........ 32
Figure 17: Sliding Condition for Model-Based Nonlinear System with Moving Boundary Layer ................. 33
Figure 18: Control Effort for Model-Based Nonlinear System with Moving Boundary Layer ....................... 33
Figure 19: First-Order Linear Simulink Diagram of Sliding Mode Controller ................................................... 39
Figure 20: Open Loop System ......................................................................................................................... 39
Figure 21: Signum or Sat Function .................................................................................................................. 40
Figure 22: Desired Tracking .............................................................................................................................. 40
Figure 23: Position Tracking Error for First-Order Linear System .................................................................... 41
Figure 24: Position Comparison of Desired and Actual Responses for First-Order Linear System .................. 41
Figure 25: Velocity Tracking Error for First-Order Linear System .................................................................... 41
Figure 26: Velocity Comparison of Desired and Actual Responses for First-Order Linear System ............... 41
Figure 27: Sliding Condition for First-Order Linear System ............................................................................. 42
Figure 28: Control Effort for First-Order Linear System ................................................................................ 42
Figure 29: Phase Plane for First-Order Linear System .................................................................................... 42
Figure 30: Phase Plane of the Sliding System for First-Order Linear System .................................................... 42
Figure 31: Higher-order Linear Simulink Diagram of Sliding Mode Controller ............................................... 44
Figure 32: Open Loop System ......................................................................................................................... 45
Figure 33: Signum or Sat Function .................................................................................................................. 45
Figure 34: Desired Tracking .............................................................................................................................. 45
Figure 35: Position Tracking Error for Higher-Order Linear System .............................................................. 46
Figure 36: Position Comparison of Desired and Actual Responses for Higher-Order Linear System ............ 46
Figure 37: Velocity Tracking Error for Higher-Order Linear System ............................................................... 47
Figure 38: Velocity Comparison of Desired and Actual Responses for Higher-Order Linear System .......... 47
Figure 39: Acceleration Tracking Error for Higher-Order Linear System ....................................................... 47
Figure 40: Acceleration Comparison of Desired and Actual Responses for Higher-Order Linear System .... 47
Figure 41: Sliding Condition for Higher-Order Linear System .......................................................................... 48
Figure 42: Control Effort for Higher-Order Linear System ............................................................................. 48
Figure 43: Phase Plane for Higher-Order Linear System ................................................................................ 49
Figure 44: Phase Plane of Sliding System for Higher-Order Linear System .................................................... 49
Figure 45: First-Order Linear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer .... 51
Figure 46: Open Loop System ......................................................................................................................... 51
Figure 47: Signum or Sat Function .................................................................................................................. 52
Figure 48: Desired Tracking .............................................................................................................................. 52
Figure 49: Position Tracking Error for First-Order Linear System with Moving Boundary Layer ............... 53
Figure 50: Position Comparison of Desired and Actual Responses for First-Order Linear System with Moving Boundary Layer .................................................................................................................. 53
Figure 51: Velocity Tracking Error for First-Order Linear System with Moving Boundary Layer .......... 53
Figure 52: Velocity Comparison of Desired and Actual Responses for First-Order Linear System with Moving Boundary Layer .......................................................... 53
Figure 53: Boundary Layer for First-Order Linear System with Moving Boundary Layer .......................................................... 54
Figure 54: Control Effort for First-Order Linear System with Moving Boundary Layer .......................................................... 54
Figure 55: Phase Plane for First-Order Linear System with Moving Boundary Layer .......................................................... 55
Figure 56: Phase Plane of Sliding System for First-Order Linear System with Moving Boundary Layer .......................................................... 55
Figure 57: Higher-order Linear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer .......................................................... 57
Figure 58: Open Loop System ............................................................................................................................................. 57
Figure 59: Signum or Sat Function ...................................................................................................................................... 58
Figure 60: Desired Tracking .................................................................................................................................................. 58
Figure 61: Position Tracking Error for Higher-Order Linear System with Moving Boundary Layer .......................................................... 59
Figure 62: Position Comparison of Desired and Actual Responses for Higher-Order Linear System with Moving Boundary Layer .......................................................... 59
Figure 63: Velocity Tracking Error for Higher-Order Linear System with Moving Boundary Layer .......................................................... 59
Figure 64: Velocity Comparison of Desired and Actual Responses for Higher-Order Linear System with Moving Boundary Layer .......................................................... 59
Figure 65: Acceleration Tracking Error for Higher-Order Linear System with Moving Boundary Layer .......................................................... 60
Figure 66: Acceleration Comparison of Desired and Actual Responses for Higher-Order Linear System with Moving Boundary Layer .......................................................... 60
Figure 67: Boundary Layer for Higher-Order Linear System with Moving Boundary Layer .......................................................... 61
Figure 68: Control Effort for Higher-Order Linear System with Moving Boundary Layer .......................................................... 61
Figure 69: Phase Plane for Higher-Order Linear System with Moving Boundary Layer .......................................................... 61
Figure 70: Phase Plane of Sliding System for Higher-Order Linear System with Moving Boundary Layer .......................................................... 61
Figure 71: First-Order Nonlinear Simulink Diagram of Sliding Mode Controller .................................................................................. 68
Figure 72: Open Loop System .................................................................................................................................................. 68
Figure 73: Signum or Sat Function ...................................................................................................................................... 69
Figure 74: Desired Tracking .................................................................................................................................................. 69
Figure 75: Position Tracking Error for First-Order Nonlinear System .................................................................................. 70
Figure 76: Position Comparison of Desired and Actual Responses for First-Order Nonlinear System .......................................................... 70
Figure 77: Velocity Tracking Error for First-Order Nonlinear System .................................................................................. 70
Figure 78: Velocity Comparison of Desired and Actual Responses for First-Order Nonlinear System .......................................................... 70
Figure 79: Sliding Condition for First-Order Nonlinear System .................................................................................. 71
Figure 80: Control Effort for First-Order Nonlinear System .................................................................................. 71
Figure 81: Phase Plane for First-Order Nonlinear System .................................................................................. 71
Figure 82: Phase Plane of the Sliding System for First-Order Nonlinear System .......................................................... 71
Figure 83: Higher-order Nonlinear Simulink Diagram of Sliding Mode Controller .................................................................................. 73
Figure 84: Open Loop System .................................................................................................................................................. 73
Figure 85: Signum or Sat Function ...................................................................................................................................... 74
Figure 86: Desired Tracking .................................................................................................................................................. 74
Figure 87: Position Tracking Error for Higher-Order Nonlinear System .................................................................................. 75
Figure 88: Position Comparison of Desired and Actual Responses for Higher-Order Nonlinear System .......................................................... 75
Figure 89: Velocity Tracking Error for Higher-Order Nonlinear System .................................................................................. 75
Figure 90: Velocity Comparison of Desired and Actual Responses for Higher-Order Nonlinear System .......................................................... 75
Figure 91: Acceleration Tracking Error for Higher-Order Nonlinear System .................................................................................. 76
Figure 92: Acceleration Comparison of Desired and Actual Responses for Higher-Order Nonlinear System .......................................................... 76
Figure 93: Sliding Condition for Higher-Order Nonlinear System .................................................................................. 77
Figure 94: Control Effort for Higher-Order Nonlinear System .................................................................................. 77
Figure 95: Phase Plane for Higher-Order Nonlinear System .................................................................................. 77
Figure 96: Phase Plane of the Sliding System for Higher-Order Nonlinear System .................................................................................. 77
Figure 97: First-Order Nonlinear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer .................................................................................. 79
Figure 98: Open Loop System .................................................................................................................................................. 79
Figure 99: Signum or Sat Function ...................................................................................................................................... 80
Figure 100: Desired Tracking .................................................................................................................................................. 80
Figure 101: Position Tracking Error for First-Order Nonlinear System with Moving Boundary Layer .......................................................... 81
Figure 102: Position Comparison of Desired and Actual Responses for First-Order Nonlinear System with Moving Boundary Layer .............. 81
Figure 103: Velocity Tracking Error for First-Order Nonlinear System with Moving Boundary Layer
Figure 104: Velocity Comparison of Desired and Actual Responses for First-Order Nonlinear System with Moving Boundary Layer
Figure 105: Boundary Layer for First-Order Nonlinear System with Moving Boundary Layer
Figure 106: Control Effort for First-Order Nonlinear System with Moving Boundary Layer
Figure 107: Phase Plane for First-Order Nonlinear System
Figure 108: Phase Place of Sliding System for First-Order Nonlinear System with Moving Boundary Layer
Figure 109: Higher-order Nonlinear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer
Figure 110: Open Loop System
Figure 111: Signum or Sat Function
Figure 112: Desired Tracking
Figure 113: Position Tracking Error for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 114: Position Comparison of Desired and Actual Responses for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 115: Velocity Tracking Error for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 116: Velocity Comparison of Desired and Actual Responses for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 117: Acceleration Tracking Error for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 118: Acceleration Comparison of Desired and Actual Responses for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 119: Boundary Layer for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 120: Control Effort for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 121: Phase Plane for Higher-Order Nonlinear System with Moving Boundary Layer
Figure 122: Phase Place of Sliding System for Higher-Order Nonlinear System with Moving Boundary Layer
NOMENCLATURE

\[ t \] Time
\[ \frac{d}{dt} \] Derivate in Respect to Time
\[ x^{(n)} \] System to be Controlled
\[ n \] System Order/Number of States
\[ x \] State Measurement
\[ \dot{x} \] Derivative of State Measurement
\[ \ddot{x} \] Double Derivative of State Measurement
\[ x_d \] Desired State
\[ \dot{x}_d \] Derivate of Desired State
\[ \ddot{x}_d \] Double Derivate of Desired State
\[ \bar{x} \] Difference Between State Measurement and Desired State
\[ u \] Controller Input
\[ u_{k-1} \] Previous Value of Controller Input
\[ s \] Sliding Surface
\[ \dot{s} \] Derivative of the Sliding Surface
\[ V(x) \] Candidate Lyapunov Function
\[ \dot{V}(x) \] Derivate of Candidate Lyapunov Function
\[ \text{sgn}(s) \] Signum Function of the Sliding Surfaces
\[ \text{sat}(s/\phi) \] Saturation Function
\[ K \] Switching Gain
\[ \phi \] Boundary Layer
\[ \dot{\phi} \] Time-Varying Boundary Layer
\[ \lambda \] Slope of the Sliding Surface
\[ \eta \] Small Strictly Positive Constant
\[ m \] Mass
\[ c \] Damping Coefficient
\[ k \] Spring Constant
1 INTRODUCTION

1.1 Discussion of Sliding Mode Control/Motivation

Lyapunov based controllers have received much attention recently due to their robustness and ability to control nonlinear systems directly with tracking stability guaranteed in the Lyapunov sense. In particular, the Sliding Mode Control (SMC) strategy uses Lyapunov’s Direct Method to ensure asymptote tracking stability of state trajectories within the phase plane in the presence of modelling uncertainties and can be applied to nonlinear system models directly. However, in most cases a system model must be developed (that can be either linear or nonlinear) to derive the form of the sliding mode controller. If a SMC law can be developed that does not rely on a system model but is solely based on measurements the application of the control law can be generalized encompassing all system types and is the research goal of this work.

1.2 Background Research on Sliding Mode Control

In this section, a selected overview of previous work conducted using the sliding mode control method is presented. This section presents literature for SMC applications that required a system model to be first developed. The next section introduces limited attempts in the development of a model-free approach and outlines the gaps in the previously developed works to be culminated in the thesis work.

1.2.1 Control Schemes Based on Sliding Mode Control

Bandyopdhyay et al. [1] applied a method of a reduced-order model approach to sliding mode control for control of higher-order systems. The work considered a higher-order continuous-time system and transformed the system into Jordan form for separating the dominant and non-dominant modes. By retaining only the dominant modes of the system, they were able to create a reduced model of the system. From there, the authors were able to successfully design a “stable” sliding surface for the system using standard sliding mode control theory. The authors then used an aggregation matrix to express the controller of the reduced-order model in terms of the states of the higher-order system. They applied the new controller to the higher-order system and demonstrated a quasi-sliding mode motion for the system. Finally, the authors applied the method in an illustrative example of a continuous-time 4th-order system. They transformed the system into a reduced-order model and designed a sliding mode controller for the model. Using
the designed controller for the reduced-order model, a new controller (based on the higher-order system model and aggregation matrix) was derived. The closed-loop system proved to be asymptotically globally stable for the tracking performance. Therefore, the authors showed that the reduced-order method can be considered a valid method of control design. The authors also optimized the controller using linear quadratic regulator to minimize the quadratic cost function. The method proved to be successful and viable, but required a known system model.

Pai [2] used a discrete-time integral sliding mode control scheme to accurately track and model uncertain linear systems. Pai used the discrete-time integral sliding mode control method to design a controller in order for an auxiliary system to be stable. He first designed a switching surface in discrete-time using the concept of the integral switching function in continuous-time sliding mode control. From there, a control law was designed such that quasi-sliding mode is reached due to the concept that a discrete-time SMC system can only approach the switching surface as opposed to reaching it and remaining on the switching surface as in continuous-time SMC systems. Pai proved that using the designed controller and integral switching function, stability is guaranteed with zero tracking error, which was displayed in an illustrative example. Using a discrete-time integral sliding mode control scheme, Pai was able to guarantee the stability of closed-loop systems with zero tracking error in the presence of parameter uncertainties and external disturbances. Chattering and a reaching phase were also eliminated. Discrete-time integral sliding mode control was shown to be effective control scheme in this work, but once again a known system model was used to derive the control law.

Laghrouche et al. [3] proposed a higher-order sliding mode control scheme for uncertain nonlinear systems. The authors first proposed an uncertain nonlinear system. A controller was then designed that used the integral sliding mode concept. The controller consisted of two parts. The first part was a feedback controller that was continuous and was stabilized in finite time at the origin when there were no uncertainties. The second part was a discontinuous controller that provided compensation of the uncertainties and ensured the control objective was reached. The authors tested the controller performance by applying the control law to a kinematic model of an automobile. The controller was designed to robustly steer the automobile from an initial position over a specified trajectory. The system state was able to converge within the desired time and
tracked the desired trajectory without chattering. Although the controller was shown to be effective and robust, once again a system model form was required.

Chang and Wang [4] proposed a controller and sliding surface form such that the theorized system would reach a corresponding sliding surface. By using the invariance property, the controlled system was not affected by plant error or the model reference input and enabled the achievement of error state covariance assignment in a stochastic model reference system. First, the authors described a linear time invariant stochastic plant system and the desired model to be tracked. A controller was then developed to ensure the desired error state covariance is met for the system. With the utilization of sliding mode control, the authors were able to design a feedback gain matrix that achieved the error state covariance assignment and determined the sliding surface with the assumed linear time invariant stochastic plant model.

Kwon et al. [5] discussed a method of using a robust Model Predicative Control (MPC) to improve performance during the reaching phase mode. MPC is a control algorithm that optimizes the control performance at each discrete time step. The disadvantage of MPC though is knowledge of an exact system model is required. The performance of the MPC law is severely degraded and limited if there is any uncertainty in the model. To compensate, the authors transformed an uncertain discrete time system into a polytropic uncertain system and applied MPC during the reaching mode. MPC enabled the system to reach the sliding surface faster with improved performance. First, the authors described an uncertain discrete-time system and transformed it into a polytropic uncertain linear time-varying system. Next, a sliding surface and cost penalty function were defined to minimize the time during the reaching mode. To minimize the reaching time, a control law was designed minimizing the cost penalty function and reachability was proved and is satisfied. Finally, a quasi-sliding mode controller was designed to enable the system state to reach the equilibrium point after the controller achieved reachability. The authors then applied the combined controllers in an example and proved the approach was more effective than alternate reaching control methods.

Nizar et al. [6] proposed a predictive sliding mode control for state time delay systems. Time delays are difficult to predict and affect many systems in various ways including state input, state output, stability of the system, and the control performance. In order to overcome the problem of
time delay, the authors used techniques of the sliding mode control method and model-based predictive control to develop a model predictive sliding mode control law. First, a discrete-time delay system and classical sliding function were defined. Using the sliding function, a control law for the system was then derived. From there, a cost function was derived to apply model predictive control principles to the control law. Using the cost function, the authors were able to derive an optimal discrete predictive sliding mode control law. To prove the effectiveness of the control law, the controller was applied to a discrete-time system and results were compared to the performance of a classical sliding mode controller applied to the same system. The simulations of the proposed technique showed a faster convergence time and less tracking performance error in comparison to the classic controller, proving the discrete predictive sliding mode controller is more effective than the classic controller.

Pai [7] proposed a control scheme for linear systems involving mismatched state and input delays. He developed the control algorithm using discrete-time sliding mode control and time-delay control theory. Time-delays are a frequent cause of instability and are difficult to control. Pai’s control scheme enabled robust control over linear systems with mismatched state and input delays. The control scheme also eliminated the reaching phase and chattering. The controller’s effectiveness was proven using simulation analysis.

Runcharoon and Srichatrapimuk [8] discussed developing a controller based on sliding mode control for an autonomous quadrotor. The autonomous quadrotor was classified as an Unmanned Aircraft System (UAS) and much attention has been shown recently in researching UASs, especially in improving the control system design. The authors chose to develop a control system for the quadrotor based on the sliding mode control method due to the accuracy the control method can provide. The controller included two parts: the sliding mode controller was applied to the attitude of the system and a Proportional-Derivative (PD) type controller was applied to the altitude and the position states of the system. Once the controller was developed, the authors simulated a quadrotor system tracking scenario and the controller was integrated into the simulation. The results proved the controller to be robust by achieving the desired position and yaw angle. However, once again, a system model was assumed known to derive the control law.
Cunha et al. [9] developed an Unit Vector Model-Reference Adaptive Control System algorithm. The system used an output-feedback model-reference sliding mode controller in the control of multivariable linear systems. The output-feedback unit vector control allows the system to enter a sliding mode condition. The benefit of the approach is the only information required is the proof that the controller satisfies the Hurwitz condition when including an input gain. The Hurwitz condition ensures the positive definite requirement for stability. Simulation results using the designed controller showed the system is globally exponentially stable, but knowledge of the system model was necessary.

Ding et al. [10] developed a controller based on SMC with applications to nonlinear systems. The controller was developed by adding a power integrator and using a nested saturation scheme. The controller consisted of two parts: a saturated part and a domination part. The saturated part was used to drive the states onto a sliding surface that was defined for the system. The domination part was used to minimize the effects of uncertainty regarding the system to be controlled. Simulation results proved the controller to be effective by establishing global convergence and stability.

As shown in [1-10], sliding mode control design is useful and popular method in the control of nonlinear systems containing modeling uncertainties. Several researchers are developing new control schemes based on the sliding mode control method to enable better control performance for a variety of systems encompassing both linear and nonlinear systems. The researchers described above were able to successfully use the SMC method to derive control laws ensuring stable tracking performance for a wide range of systems. However, each derived control law required some knowledge of the system model, which is not always easily obtainable if at all possible. The following section describes ongoing research in developing a model-free control approach based on the sliding mode control method.

1.2.2 Model-Free Controllers Based on Sliding Mode Control

Salgado-Jimenez et al. [11] applied a model-free method based on the sliding mode control method to control a one degree-of-freedom underwater vehicle. Two issues arise in the control an underwater vehicle: parametric uncertainty and rejection of unknown disturbance. The sliding mode control method is an effective solution to the problem. However, the induced chattering of
the control effort can have negative effects onto the system to be controlled including actuator
damage or deteriorating actuator life time due to infinite band switching. The authors proposed a
method called Model-free High Order Sliding Modes Control removing the chattering effects of
sliding mode control, while maintaining stable tracking. The control technique does not require
the dynamics or parameters of the system except for the exponential convergence to the desired
trajectory. To test the method, a comparison was made of the results for the model-free controller
to a PID controller and a standard sliding mode controller in a one degree-of-freedom underwater
system test. The underwater system was restricted to only heaving motion and the rotational
motion was constrained, i.e., locked. The desired trajectories of the system to be tracked were a
defined sine wave and a triangular wave. The controllers were subjected to various disturbances
during the test to validate the performance. The results proved the model-free controller to be the
most effective, having smooth responses and the least amount of error for both trajectory paths.
The designed controller is based on a combination of Proportional-Derivative control and sliding
mode control, where in this proposed work the controller is based solely on sliding mode control.
The controller is also designed with the assumption the system would only operate in one
direction, where in this work the gap is closed in considering multiple states trajectories.

Martinez-Guerra et al. [12] discussed a synchronization problem for chaotic systems. One of the
solutions to the problem is the use of control law schemes since control laws enable
synchronization of nonlinear systems. In order to develop a control law for nonlinear systems, in-
deepth knowledge of the nonlinear dynamics of the system is required, which is not easily
obtainable. In order to compensate for this problem, the authors proposed a model-free sliding
observer. The observer uses a sliding mode term that is robust against output noise from the
system. The authors then compared their observer against two established model-based
observers. Their analysis showed the model-free observer produced greater error in comparison
to model-based controllers. However, their conclusion stated the model-free observer is more
beneficial due to the fact it does not require knowledge of the system. The gap in this work is an
observer is required, where in this proposed work no observer is required.

Raygosa-Barahona et al. [13] developed a controller using the concepts of backstepping and 2nd.
order sliding mode controller form and applied the controller to a Remotely Operated Vehicle
(ROV). By following the theory of the backstepping technique, the authors were able to develop
a model-free sliding mode controller with similar performance to a traditional PID controller. When applied to the ROV system, the controller enabled the system to follow the desired trajectory without chattering. The designed controller is derived from a traditional PID control strategy with sliding mode theory and backstepping techniques, where in this proposed work is based solely on the sliding mode control method.

Munoz-Vazquez et al. [14] developed a model-free integral sliding mode controller for position control of a quadrotor. The controller consisted of three subsystems: the model-free control subsystem to enforce sliding mode, the velocity field subsystem, and the sliding surface subsystem. The controller was designed by modifying the nominal reference to include the velocity field as the desired velocity to be tracked. The nominal reference was also used to create the sliding surface so that the quadrotor remains on the passive velocity field to ensure stability. Simulation results proved the controller to be effective by showing the robustness of the approach in the control of the quadrotor. The designed controller is derived from velocity field control, where in this work is, once again, based solely on sliding mode control.

1.3 Background Research on Unmanned Aircraft Systems

One use of the work proposed here is the application to Unmanned Aircraft Systems (UASs). UASs will be predominant in the future and will require control systems for maintaining stability and enhancing their performance. Currently, individual control systems for UASs is derived and applied to each UAS. The result of this research work will enable a single control system that can be applied to all types of UASs with little design modifications since the proposed method relies solely on measurements and does not require a system model. The measurements are based on traditional type sensors widely available for UAS applications and are assumed to be known. Therefore, the advantage of the proposed new type of controller is the ability of using one control law applied to many types of UASs. Many types of UAS applications are therefore possible (as outlined next) resulting from the new type of model-free control strategy.

1.3.1 Collision Avoidance for Unmanned Aircraft Systems

Collision avoidance has been an important topic for integration of Unmanned Aircraft Systems (UAS) into the National Airspace System (NAS). A system model is normally required to be developed an autonomous collision avoidance/sense and avoid control law for an UAS. Mcentee
et al. [15] proposed a Concept of Operations including the combination of ground based sensors and ground based alerting to be used along with an on-board autonomous Collision Avoidance System (CAS) to improve UAS collision avoidance performance. McNett et al. conducted a simulation to assess the potential for ground fusion and concluded multi-sensor coverage is important for surveillance in a fusion system. The new model-free controller lends to this type of application to further increase the performance with no knowledge of the system’s model.

Consiglio et al. [16] discussed the importance of a Sense and Avoid (SAA) system for UAS to improve collision avoidance. Consiglio et al. stated the SAA system should be able to determine a threat of collision using Threat Detection and/or Resolution (TD&R) schemes. The TD&R system would then relay a solution to the pilot who would ultimately make the decision unless the autonomous mode is required to make a quick decision. Once again, the model-free approach can be easily developed for providing the autonomous control strategy.

Asmat et al. [17] also proposed an Unmanned Aerial Collision Avoidance System (UCAS) for UASs. The UCAS will interact with the Traffic Alert Collision Avoidance System to allow for better collision avoidance. The UCAS will also be able to sense, detect, and avoid non-cooperative aircraft using sensors along with cooperative aircraft. However, Asmat et al. did not complete their analysis to determine the performance of a UCAS.

Smith and Taylor [18] discussed one of the most important factors in collision avoidance of UAS: the constant attention of Air Traffic Controllers (ATCs). UAS have been operating in Class D airspace at the Southern California Logistics Airport, Victorville, CA (VCV) since 2006. The Serco ATC at the VCV tower provides Visual Flight Rules (VFR) to enable a safe airspace of both manned and unmanned aircraft. ATC also provides see and avoid information for the UAS operator and ensures a safe airspace if communication is lost between the UAS and the operator. ATCs are a key factor in the safe integration of UAS into the NAS, however, advanced control strategies (such as the model-free approach developed in this work) will be required for purely autonomous types of UAS operations.

1.3.2 Perspectives on Integrating UAS into the National Airspace System

Since the integration of UASs into the National Airspace System (NAS) is in its early stages, a myriad of research has been dedicated in gaining the perspectives and opinions of people who
are and will be involved with UASs. Comstock et al. [19] developed a survey to address UAS control, navigation, and communications regarding unmanned aircraft of various sizes and capabilities. Questions were posed to civilian ATCs, military ATCs, pilots of manned aircraft, and pilots of UASs. The answers to the survey concluded most UASs are difficult to notice and both ATCs and manned aircraft pilots desire knowledge of the presence of the UAS in order for the ATC to notify whether the UAS will respond to a traffic alert and collision avoidance system resolution advisory. The survey also described the UAS camera imagery as inefficient. There is still much to be improved before the full integration of UASs into the NAS with a conclusion that advanced control strategies (such as ones researched here) will be vital for integration of UASs into the NAS.

Pestana [20] explained that currently a standard flight crew interface does not exist for UASs. Operating a UAS is also difficult without in-situ sensory input and feedback. These limitations can lead to many problems and frustrations for the pilots. Pestana believes established knowledge, skill sets, training, and qualification standards for UAS pilots are important. The software at the Ground Control Station needs to be user-friendly in order to operate the UAS more efficiently, which will once again require advanced control system technologies.

Logan [21] discussed a possible framework for integration of small Unmanned Aircraft Systems (sUAS) into the NAS. He states that a person should be allowed to operate a sUAS on their property up to 500ft at their own risk due to the fact that manned aircraft operates at a much higher minimum altitude. He also states that in urban areas, people are safe to operate sUAS at or below building heights. Logan goes on to explain that sUAS should be allowed unrestricted access to the NAS if the sUAS are under a certain size and weight. He explains further research needs to be conducted to determine the risk levels of larger sUAS and if certain equipage should be required. Logan concludes that many sUAS operations can be conducted safely without regulation and other operations should be regulated based on the risk the sUAS poses to the NAS and ground personnel. Obviously, if UASs are allowed to operate uninhibitedly by untrained civilians, control strategies will be required for safe and easy operation of UASs.

Van Dyk et al. [22] analyzed the challenges of integrating UASs into the NAS using a systems-of-systems method to determine the risks. Using a Hierarchical Holographic Model, Van Dyk et
al. identified UAS vulnerabilities and important risk scenarios that require further investigation. The areas of concern include: personnel, (and in particular) command and control, security, and cyber-infrastructure. The major risks identified several scenarios of acts of terrorism. Van Dyk et al. also concluded that by using Multi-Objective Decision Trees (MODT), the pilot will be able to regain control of the UAS in the event a malicious pilot gained control, but more research is required to populate the MODT with more scenarios and probable outcomes.

A simulation of four experimental highway patrol police missions were conducted by Fern et al. [23] to determine whether introducing a Cockpit Situation Display (CSD) into a UAS Ground Control Station (GCS) would improve pilot performance. Fern et al. also theorized the use of Air Traffic Control (ATC) is a possible solution to maintain separation assurance. According to the questionnaires, the UAS pilots were able to follow the instructions appropriately concerning the mission and instructions from ATC in comparison to pilots of manned aircraft. The results of the missions also indicated that the UAS pilots had sufficient knowledge of the airspace and procedures, meaning they were able to interact with the ATC efficiently by responding to their instructions. When the CSD was introduced the results concluded that interaction was easier between UAS pilots and ATC as well as significantly improved Situational Awareness (SA) for the UAS pilots. However, the UAS pilots may have only believed SA was improved because the CSD provided more information. The experiment showed that ATC plays a critical role in maintaining safe separation between the UAS and other manned aircraft.

Paczan et al. [24] discussed the Next Generation Air Transportation System (NextGen) plan set out by the Federal Aviation Administration (FAA) to improve the NAS. The integration of UAS into NextGen is a crucial part. A set of standards needs to be created for UAS in order to be properly incorporated into the NAS. The information will enable NextGen to accommodate UAS operations in the new automation system for both terminal and en-route ATC. A main goal of NextGen is to improve communication between aircrafts and ATC. Communication with UAS is more complicated due to the lack of an onboard pilot so it is important to create an efficient system in order to prevent future problems. Voice recognition capabilities for the UAS could benefit communications and improve efficiency for relaying commands. Matolak et al. [25] also discuss the importance of reliable communication links. The environment can affect the signal to
the UAS. Testing must be conducted in different scenarios to ensure the UAS is always receiving the necessary command.

As show in [15-25], concerns over the control of Unmanned Aircraft Systems (UASs) is inhibiting their integration into the NAS. The model-free control scheme developed in this work can directly be applied to all types of UASs platforms and missions. The control system will enable more accurate measurements of the position and better performance of the UAS, which will enable more control over the UAS. Having complete control over the UAS without uncertainty will aid in the safe integration of UASs into the NAS.

1.3.3 UAS Testing Site
The Federal Aviation Administration announced Griffiss International Airport will be authorized to test commercial Unmanned Aircraft Systems (UAS), one of six sites nationwide. The success of the application is credited to Griffiss’ close proximity to other UAS testing sites and military bases who utilize UAS. Rome Laboratory and a technology park are also located at Griffiss, which encouraged many technology related businesses to locate in the area. Now that testing of UASs is allowed at Griffiss, the businesses will be able to become involved with the development and implementation of UAS. Griffiss will now play a key role in the FAA’s goal of integrating UAS into the national airspace by 2015. [26]

Results from testing at the site will help to develop regulations to allow integration of UAS into the NAS as well. The testing will include determining the safety of flying UAS and analyzing the performance of the pilots control over the UAS with instrument only flight. Testing will also ensure the pilot has constant control over the UAS and cannot be overtaken by another source. [27] Flight testing of new and revolutionary advanced control concepts (such as the one developed here) can be evaluated at the test site.

1.4 Research Goals
The goal of the research is to develop and demonstrate a model-free sliding mode control scheme to achieve accurate tracking performance for both linear and nonlinear systems along with guaranteeing stability for the tracking convergence. As discussed earlier, model-free methods using sliding mode control have been researched in the past. This work presents a new approach
to a model-free control strategy that proves to have robust control over various system types. The control scheme, researched here, can then later be applied to control of unmanned aircraft systems. The control system will enable better guidance and navigation control, and enable improved performance of UASs. Ultimately, improved control over the UAS will aid in the integration into the NAS more quickly and more easily.

The thesis is outlined as follows. Chapter 2 introduces the sliding mode control method along with Lyapunov stability concepts including an illustrative example for a nonlinear system with a known model. Chapter 3 defines the sliding surface and control law of the described system using a model-free approach and outlines a proof of the new controller form. The switching gain and boundary layer concept of the new controller is also developed and presented in Chapter 3. Chapters 4 and 5 provide illustrative linear and nonlinear examples respectively of the derived model-free Sliding Mode Control scheme. Conclusions, suggestions for future work, and applications to society of the work are described in Section 6.
THE SLIDING MODE CONTROL METHOD

In this chapter, the Sliding Model Control (SMC) method is introduced and an illustrative example using a known nonlinear model is presented. The SMC method is based on Lyapunov’s Direct Method, which provides a stability analysis approach for nonlinear type systems. Lyapunov’s Direct Method is first introduced and a basic example is presented illustrating the utilizing of using the method in proving the stability of a nonlinear system. Next, an overview of the SMC method is outlined including a practical approach for handling chattering issues for the SMC control algorithm. Finally, a 2nd-order nonlinear control example with an assumed model form is presented showing the effectiveness of the SMC method in maintaining stable tracking performance which forms a basis for developing a model-free SMC control strategy.

2.1 Lyapunov’s Direct Method

The concept of Lyapunov’s Direct Method [28] is based on the rate of change of the energy in a system for predicting stability. Since the concept is based on the energy contained within the system the method has applications to both linear and nonlinear systems. The method is based on the notion if the rate of change of the system’s energy is continuously dissipating (after a disturbance is injected to the system) then the system trajectories will eventually reach an equilibrium point. Therefore, positive stability can be inferred for any system, i.e., the system is stable since it reaches and remains at an equilibrium point. Mathematically, a candidate “Lyapunov function” for a system is first developed that ensures a positive energy state is inferred on the system. If the rate of change in the candidate Lyapunov function (i.e., the derivative of the function) is continuously decreasing, i.e., negative, then the system is shown to be stable. Furthermore, if the system reaches an equilibrium point then the system is asymptotically stable. If the system is asymptotically stable for any disturbance then the system is globally asymptotically stable. If the candidate Lyapunov function cannot be shown to be continuously decreasing, no inference can be made about the system’s stability characteristics. The method is extremely powerful in analyzing the stability of nonlinear systems and forms a basis of the SMC method in the application of the method to nonlinear systems.

To begin with, consider a scalar function, $V$, with domain, $D$, such that $V: D \rightarrow \mathbb{R}$ is continuous and satisfies $V(0) = 0$. Then:
- If $V(x) \geq 0$ for every $x \in D$, then $V$ is positive semi-definite.
- If $V(x) > 0$ for every $x \in D$, then $V$ is positive definite.
- If $V(x) \leq 0$ for every $x \in D$, then $V$ is negative semi-definite.
- If $V(x) < 0$ for every $x \in D$, then $V$ is negative definite.

The Lyapunov Stability System states that if the function, $V$, is positive definite and the derivative of the function, $\dot{V}$, is negative semi-definite, then the equilibrium point $x = 0$ is stable for a function $x = f(x)$. If $\dot{V}$ is negative definite and $V$ is radially unbounded, i.e., $V(x) \to \infty$ as $|x| \to \infty$, then the equilibrium point $x = 0$ is globally asymptotically stable.

A candidate Lyapunov function can be used to determine the stability of the system. However, if the candidate Lyapunov function does not satisfy the above criteria, it is not conclusive the system is unstable. If any candidate Lyapunov function exists that satisfies the above criteria, the system is considered stable.

2.1.1 Example of Lyapunov’s Direct Method

Consider a pendulum system without friction with the following system model:

$$\ddot{\theta} + \frac{g}{l} \sin(\theta) = 0 \quad (1)$$

where $l$ is the length of the pendulum, $m$ is the mass of the pendulum, and $\theta$ is the position of the pendulum.

The state variables of the system are:

$$x_1 = \theta \quad (2)$$
$$x_2 = \dot{\theta} \quad (3)$$

The time derivatives of the state variables are thus equal to:

$$\dot{x}_1 = x_2 \quad (4)$$
$$\dot{x}_2 = -\frac{g}{l} \sin(x_1) \quad (5)$$

The above transformed system gives the desired form of $\dot{x} = f(x)$ with the origin as the equilibrium point since $f(0) = 0$. In order to prove stability of the system, a candidate Lyapunov function is chosen that satisfies the stability criteria. In order to determine a candidate Lyapunov function, consider the energy of the system:

$$E = K + P \quad (6)$$
\[ E = \frac{1}{2} m(\omega l)^2 + mgh \]  

(7)

where:

\[ \omega = \dot{\theta} = x_2 \]  

(8)

\[ h = l(1 - \cos(\theta)) = l(1 - \cos(x_1)) \]  

(9)

Substituting \( \omega \) and \( h \) into Eq. (7) yields:

\[ E = \frac{1}{2} ml^2 x_2^2 + mgl(1 - \cos(x_1)) \]  

(10)

A logical choice of the candidate Lyapunov function is:

\[ V(x) = E \]  

(11)

The above Lyapunov function satisfies the following criteria:

\[ V(0) = 0 \]  

(12)

\[ V(x) > 0 \text{ for the interval } (-2\pi, 2\pi) \]  

(13)

Taking the derivative of Eq. (11) yields:

\[ \dot{V}(x) = ml^2 x_2 \dot{x}_2 + \dot{x}_1 mgl \sin(x_1) \]  

(14)

and substituting in for \( \dot{x}_2 \) yields:

\[ \dot{V}(x) = \dot{x}_1 mgl \sin(x_1) + \dot{x}_1 mgl \sin(x_1) = 0 \]  

(15)

Thus, based on the stability criteria, the system is marginally stable. Since \( \dot{V}(x) \) is not less than zero, the system is stable in the Lyapunov sense, but it is not asymptotically stable.

### 2.2 The Sliding Mode Control Method

In Slotine and Li [29] a method for developing a sliding mode control law based on an assumed model was introduced. A switching law is used to drive the system trajectory onto a sliding surface. Stability of the closed-loop system is ensured by Lyapunov’s Direct Method. The switching law is determined based on a candidate Lyapunov function in order to guarantee stability such that the system’s state trajectories in the phase plane point towards the origin. The control law switches between two sets of control laws depending upon the location of the system trajectories. When the system trajectory is above or below the sliding surface, the switching law drives the system to the sliding surface and the discontinuous controller forces the states to slide towards the origin thus stability is ensured.
A drawback to this method is the control law will cause chattering due to the controller discontinuous term. The chattering phenomenon can be resolved by applying a time-varying boundary layer to smooth the control effort much like a first-order low-pass filter. When applying a boundary layer, the state trajectories are required to tend towards the origin and the updated sliding condition is required to be maintained to ensure the distance to the boundary layer is always decreasing. An example of the utility of the control method is shown next.

### 2.2.1 Derivation

Consider the following single-input, single-output system:

\[ x^n = f(x) + b(x)u \]  

(16)

for any system order \( n \), where \( x \) represents the state variable, \( f(x) \) is some function of \( x \), \( b(x) \) is the control gain, and \( u \) is the input into the system. Define the desired tracking of the system to be \( x_d \) where \( x_d(0) = x(0) \) defines the initial condition. Consider the tracking error of the system to be:

\[ \tilde{x} = x - x_d \]  

(17)

A time-varying sliding surface for the system is defined as:

\[ s = \left( \frac{d}{dt} + \lambda \right)^{n-1} \tilde{x} = 0 \]  

(18)

In order to satisfy the initial condition constraint and for the system to remain stable, the system must remain on the time-varying surface, or sliding surface. Thus the value of \( s \) must remain equal to zero. In order to force the system onto this sliding surface, a control law, \( u \), must be derived for the system such that outside of \( s \) the sliding condition is satisfied [29]:

\[ \frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \]  

(19)

where \( \eta \) is a strictly positive constant. The sliding condition forces the state trajectories towards the sliding surface, \( s \). Once on the surface, the control law forces the system trajectories to remain on the surface, achieving the “sliding” mode.

### 2.2.2 SMC Nonlinear Example with an Assumed Model

Consider the following system to be controlled:

\[ \dot{x} = -a(t)x^2 \cos(3x) + u = f + u \]  

(20)

where \( a(t) \rightarrow 1 \leq a(t) \leq 2 \) and \( f = -a(t)x^2 \cos(3x) \)

A best estimate for \( f \) is assumed to be the mean of the bounds shown above, i.e.:
\[
\hat{f} = -1.5\dot{x}^2\cos(3x) \tag{21}
\]

Define sliding surface for the 2nd-order system as:
\[
s = \dot{x} - \dot{x}_d + \lambda(x - x_d) = \dot{x} + \lambda \ddot{x} \tag{22}
\]
To ensure there is no motion from the sliding surface once the state trajectories reach the sliding surface, \(\dot{s}\) is set equal to 0, i.e.:
\[
\dot{s} = \ddot{x} - \dot{x}_d + \lambda \ddot{x} = 0 \tag{23}
\]
From Eq. (20) and Eq. (22):
\[
\hat{f} + u - \dot{x}_d + \lambda \ddot{x} = 0 \tag{24}
\]
The best estimate of \(u\) is then found to be:
\[
\hat{u} = -\hat{f} + \dot{x}_d - \lambda \ddot{x} \tag{25}
\]
The sliding condition for the system is [29]:
\[
s\dot{s} = \frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \tag{26}
\]
In order to satisfy the sliding condition, a discontinuous term is added to Eq. (25):
\[
u = \hat{u} - ks\text{sgn}(s) = -\hat{f} + \dot{x}_d - \lambda \ddot{x} - ks\text{sgn}(s) \tag{27}
\]
The sliding condition is then used to find the minimum value of \(k\). Combine Eq. (23) and Eq. (26):
\[
(\ddot{x} - \dot{x}_d + \lambda \ddot{x})s \leq -\eta |s| \tag{28}
\]
From Eq. (20): \(\ddot{x} = f + u\) and substituting into Eq. (28) results in:
\[
(f + u - \dot{x}_d + \lambda \ddot{x})s \leq -\eta |s| \tag{29}
\]
Then by substituting in for \(u\) from Eq. (27):
\[
[f - \hat{f} - ks\text{sgn}(s)]s \leq -\eta |s| \tag{30}
\]
To guarantee \(k\) is positive, assume \(\eta\) is positive and using \(\text{sgn}(s)s = |s|\), \(k\) is found to be:
\[
k \geq |f - \hat{f}| + \eta \tag{31}
\]
In order to be conservative, the largest bound is chosen for \(f\) so that Eq. (31) becomes:
\[
k \geq 0.5\dot{x}^2|\cos(3x)| + \eta \tag{32}
\]
With the final form of the sliding mode control law as:
\[
u = -\hat{f} + \dot{x}_d - \lambda \ddot{x} - ks\text{sgn}(s) \tag{33}\]
2.2.2.1 Simulation Results

Using the derived control law, sliding condition, and switching gain a control system was developed and the closed-loop system was simulated using Simulink and MATLAB for the second-order nonlinear system described above. A fixed-sampling time of 0.0001 seconds was used and the system was simulated for 30 seconds. The value of $\lambda$ was set to 20 (rad/sec) and the value of $\eta$ used was 0.1. The desired tracking of the system was $x_d(t) = \sin(\pi/2)$.

Figure 1 displays the position tracking error for the closed-loop nonlinear system. The figure shows the controller is robust with outstanding tracking performance, having a maximum tracking error ranging from -2.5e-6 to 2.5e-6. However, the controller does produce a large amount of chattering as shown next. Figure 2 displays the position comparison of the desired and actual responses for the higher-order linear system. The difference between the responses is negligible, proving the controller is robust with “perfect” tracking performance.

Figure 3 displays the velocity tracking error for the nonlinear system. The figure shows significant chatter due to the discontinuous control law. The maximum tracking error ranges from -1e-4 to 1e-4 illustrating the “perfect” tracking performance of the discontinuous control law. Figure 4 displays the velocity comparison of the desired and actual responses for the nonlinear system. The difference between the responses is negligible, proving the controller is robust and perfect tracking is achieved by at a high cost.
Figure 3 displays the velocity tracking error for the nonlinear system. The figure shows significant chatter due to the controller, having a high tracking error ranging from $-2.5$ to $2.5$. Figure 4 displays the velocity comparison for the model-based nonlinear system. The controller produced significant chatter in the system response.

Figure 5 displays the acceleration tracking error for the nonlinear system. The figure shows significant chatter due to the controller, having a high tracking error ranging from $-2.5$ to $2.5$. Figure 6 displays the acceleration comparison of the desired and actual responses for the nonlinear system. The controller produced significant chatter in the system response.

Figure 7 displays the sliding condition from Eq. (26) for the nonlinear system. The figure proves the sliding condition is satisfied at all times. However, the controller does produce a large amount of chattering. Figure 8 displays the control effort for the nonlinear system, which
experiences a significant amount of control law chatter unrealistic to be implemented in an actual physical system.

Figure 7: Sliding Condition for Model-Based Nonlinear System

Figure 8: Control Effort for Model-Based Nonlinear System

Figure 9 displays the phase plane for the nonlinear system and Figure 10 displays the phase plane of the sliding system for the nonlinear system. The controller produced a smooth response for the phase plane of the nonlinear system and proves the Lyapunov stability of the closed-loop system. However, the controller produced a significant amount of chattering for the response of the phase plane for the sliding system of the nonlinear system.

Figure 9: Phase Plane for Model-Based Nonlinear System

Figure 10: Phase Place of Sliding System for Model-Based Nonlinear System
The model-based sliding mode controller produced significant chattering for the system responses of nonlinear system. Therefore, a smoothing boundary layer should be applied to the controller to reduce the amount of chatter during the system response.

2.2.3 Inclusion of a Moving Boundary Layer

In order to eliminate chattering of the control effort, a time varying smoothing boundary layer should be applied to the controller form. A new sliding condition can be derived as shown in [29] as:

$$|s| \geq \phi \rightarrow \frac{1}{2} \frac{ds^2}{dt} \leq (\phi - \eta)|s|$$

(34)

In order to satisfy the updated sliding condition, the term $K\text{sgn}(s)$ is replaced by $(K - \dot{\phi})\text{sat}(s/\phi)$ where “sat” is a saturation function defined as:

$$\text{sat}(y) = y \text{ if } |y| \leq 1$$

(35)

$$\text{sat}(y) = \text{sgn}(y) \text{ otherwise}$$

(36)

The control law form for the system example then becomes:

$$u = 1.5\dot{x}^2|\cos(3x)| + \ddot{x}_d - \lambda\ddot{x} - (0.5\dot{x}^2|\cos(3x)| + \eta - \dot{\phi})\text{sat}(s/\phi)$$

(37)

where:

$$\dot{\phi} = -\lambda \phi + k(x_d) = -\lambda \phi + 0.5\dot{x}_d^2|\cos(3x)| + \eta$$

(38)

with:

$$\phi(0) = \eta / \lambda$$

(39)

2.2.3.1 Simulation Results

Using the derived control law, sliding condition, smoothing boundary layer, and switching gain a control system was developed and programmed in Simulink and MATLAB for the second-order nonlinear system described above. A fixed-sampling time of 0.0001 seconds was once again used and the system was simulated for 30 seconds. The value of $\lambda$ was set to 20 (rad/sec) and the value of $\eta$ used was 0.1. The desired tracking of the system was $x_d(t) = \sin(\pi/2)$.

Figure 11 displays the position tracking error for the nonlinear system with a moving boundary layer. The figure shows the controller is robust, having a maximum tracking error ranging from -3e-3 to 3e-3. The moving boundary layer produced a completely smooth tracking response. Figure 12 displays the position comparison of the desired and actual responses for the nonlinear
system with a moving boundary layer. Good agreement is shown between the position state and the desired state to be tracked. However, “perfect” tracking is not achieved as with the discontinuous control law as shown by comparing Figures 1 and 11.

Figure 11 displays the velocity tracking error for the nonlinear system with a moving boundary layer. The figure shows the controller is robust, having a maximum tracking error ranging from -0.01 to 0.01. The moving boundary layer produced a completely smooth response. Figure 14 displays the velocity comparison of the desired and actual responses for the higher-order nonlinear system with a moving boundary layer. The difference between the responses is negligible, proving the controller is robust.
Figure 15 displays the acceleration tracking error for the nonlinear system with a moving boundary layer. The figure shows the controller is robust, having a maximum tracking error ranging from -0.07 to 0.07. The moving boundary layer produced a completely smooth response. Figure 16 displays the acceleration comparison of the desired and actual responses for the nonlinear system with a moving boundary layer. The difference between the responses is relatively small, proving the closed-loop stability of the control law.

Figure 17 displays the boundary layer and sliding surface for nonlinear system with a moving boundary layer. The figure proves the sliding surface remained within the boundary at all times.
thus satisfying the updated sliding condition. Figure 18 displays the control effort for the nonlinear system with a moving boundary layer. The moving boundary layer produces a smooth control effort response with no chattering.

![Figure 17: Sliding Condition for Model-Based Nonlinear System with Moving Boundary Layer](image1)

![Figure 18: Control Effort for Model-Based Nonlinear System with Moving Boundary Layer](image2)

The model-based sliding mode control law with a moving boundary layer was shown to provide good tracking performance producing smooth system responses including the updated control effort of the nonlinear system. “Perfect” tracking performance is not achieved as with the discontinuous control law, however, “adequate” tracking is achieved without the cost of high control effort chattering. The simulation results also show that the closed-loop nonlinear system is stable in the Lyapunov sense (i.e., satisfying the sliding condition) for both the discontinuous term control law and the boundary layer smoothing control law.
3 SYSTEM DESCRIPTION

In this chapter, the structure of the model-free sliding mode control law is developed for a single-input-single-output system with a unit input influence gain. A general sliding surface is first defined for the assumed model-free form and the sliding mode controller form is then proven that satisfies the sliding condition. Once the controller form is proven, the control law is then derived and the value of the sliding surface switching gain is derived. A time varying smoothing boundary layer is derived to prevent chattering effects for realistic implementation of the new developed model-free control law.

3.1 The Model-Free Form

Consider control of an $n^{th}$-order single-input-single-output system where $n$ is the highest order of the system. The following discrete approximation can be made:

$$
\dot{x}^{(n)} \approx x^{(n)} + u - u_{k-1}
$$

(40)

where $x^{(n)}$ represents the system to be controlled, $u$ is the controller input, and $u_{k-1}$ is the previous value of the controller input. The model-free control concept assumes that measurements (or estimates) are available for the highest order derivative shown in Eq. (40). For example, the assumption is valid for an aircraft (such as for UASs) where estimates of angle-of-attack rate, sideslip rate, and true velocity rate are available using kinematic relationships and traditional type sensors such as rate gyros and linear accelerometers. Estimates of pitch, roll, and yaw accelerations are available if three tri-axial accelerometers are mounted along various locations on the aircraft. The hypothesis for the derivation of the control law is to use the robust properties of the sliding model control concept to ensure tracking stability to account for the approximation model of Eq. (40) assuming the proper measurements are available.

3.2 Define the Model-Free Sliding Surface

The sliding surface for an $n^{th}$-order single-input-single-output system can be defined as:

$$
s(t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} \ddot{x}(t)
$$

(41)

where $\lambda$ is the strictly a positive constant and is slope of the sliding surface or the bandwidth of the closed-loop system, $\ddot{x}(t) = x(t) - x_d(t)$ where $x(t)$ is the state measurement and $x_d(t)$ is the desired state to be tracked and followed. The state trajectories will remain on the sliding
surface if the model form is known exactly and “slide” down to the origin for ensuring perfect tracking performance. However, the model form is not exact and, at best, is an approximation as shown in Eq. (40) and therefore the state trajectories will move off the sliding surface with degraded tracking performance. The control law should compensate to force the state trajectories onto the sliding surface and ensure the state trajectories remain on it. The robust nature of sliding mode control can be used to meet these requirements as shown next.

3.3 The Control Law Form

The derivative of the sliding surface shown in Eq. (41) is set to zero to ensure no movement is allowed of the state trajectories in the state-plane once the trajectories reach the sliding surface. Substituting the approximation of the system model shown in Eq. (40) into the derivative of the sliding surface shown in Eq. (41) and solving for the current value of the update control effort ensures of the state trajectories remain on the sliding surface once the trajectories reach the surface, so that:

\[
\hat{u} = -\left(\frac{d}{dt} + \lambda\right)^n \hat{x}(t) + u_{k-1}
\]  

(42)

The control effort shown above represents the best approximation of the effort since the system model is imperfect and represents only an approximation as shown in Eq. (40). Since the system contains uncertainties, a discontinuous term is added to the control law in order to drive the system trajectories onto the sliding surface in the presence of modeling form error approximation. The control law is updated as follows:

\[
u = -\left(\frac{d}{dt} + \lambda\right)^n \hat{x}(t) + u_{k-1} - \eta \text{sgn}(s)
\]  

(43)

where \(\eta\) is a small strictly positive constant and \(\text{sgn}(s)\) is a signum function of the sliding surface. The underlying premise of the method is assuming that the system model is not known and only state measurements are known and assumed available. The controller input will have little change for each time step, resulting in the controller input and previous value of the controller input to cancel out to zero as the time step tends to zero. A robust controller resulting in a stable closed-loop system in the Lyapunov sense can be derived assuming the time step is finite using knowledge of the previous control law time step value and shown in Eq. (43). By describing a system model in this fashion, a controller form can be developed based on solely on system measurements and therefore does not rely on a system model.
3.4 Proof of the Controller Form

Lyapunov’s direct method is used to ensure the system states trajectories are asymptotically stable during the reaching phase when the state trajectories are not on the sliding surface. Lyapunov’s direct method states that a sufficient condition for stability is that there exists a continuously differentiable function $V(x)$ that is strictly positive definite resulting in $\dot{V}(x)$ being strictly negative definite then the equilibrium point is asymptotically stable. A candidate Lyapunov function that is strictly positive definite is defined by:

$$V(x) = \frac{1}{2} s^2 > 0$$  \hspace{1cm} (44)

which satisfies the first criteria for ensuring asymptotic stability. By taking the derivative of the candidate Lyapunov function and substituting the derivative of the sliding surface:

$$\dot{V}(x) = s \left[ \left( \frac{d}{dt} + \lambda \right)^n \ddot{x}(t) \right]$$  \hspace{1cm} (45)

Substituting the assumed model form shown in Eq. (40) and the control effort shown in Eq. (43) in the resulting equation yields:

$$\dot{V}(x) = -\eta |s| < 0$$  \hspace{1cm} (46)

The Lyapunov function is always negative definite for positive values of $\eta$, thus Lyapunov’s stability criterion is satisfied and the form of the controller effort, $u$ shown in Eq. (43) is realized.

3.5 Definition of the Control Effort and Switching Gain

The control law described in Eq. (43) is now rewritten as follows:

$$u = -\left( \frac{d}{dt} + \lambda \right)^n \ddot{x}(t) + u_{k-1} - K \text{sgn}(s)$$  \hspace{1cm} (47)

where $K$ is a to-be-determined switching gain and is derived to ensure closed-loop stability in the Lyapunov sense of the system during the reaching phase. The reaching phase is the portion of the closed-loop system when the state trajectories are not on the sliding surface and must tend towards the sliding surface ensuring stability. The sliding condition, as described by Eq. (46) and by taking the derivative of Eq. (44), can be used to find the minimum value of $K$ such that:

$$s\ddot{s} = \frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|$$  \hspace{1cm} (48)

Using the definition of the sliding surface, the system model, and the control law described by Eq. (47) we derive the following after canceling out like terms:
\[ -K|s| \leq -\eta|s| \quad (49) \]

From the above proof, the switching gain, $K$, must be greater than or equal to $\eta$ to ensure stability during the reaching phase, i.e.:

\[ K \geq \eta \quad (50) \]

### 3.6 Control with a Boundary Layer

In order to eliminate chattering of the control effort, a time varying smoothing boundary layer should be applied to the controller form. The sliding condition needs to be maintained to guarantee the distance to the boundary layer is always decreasing. The sliding condition can be updated to show:

\[ |s| \geq \phi \rightarrow \frac{1}{2} \frac{d}{dt} s^2 \leq (\dot{\phi} - \eta)|s| \quad (51) \]

In order to satisfy the new sliding condition, the term $K\text{sgn}(s)$ is replaced by $(K - \dot{\phi})\text{sat}(s/\phi)$ where “sat” is a saturation function defined as:

\[
\text{sat}(y) = y \text{ if } |y| \leq 1 \\
\text{sat}(y) = \text{sgn}(y) \text{ otherwise} \quad (52) \quad (53)
\]

The controller then becomes:

\[ u = -\left(\frac{d}{dt} + \lambda\right)^n \ddot{x}(t) + u_{k-1} - (K - \dot{\phi})\text{sat}(s/\phi) \quad (54) \]

where:

\[ \dot{\phi} = -\lambda \phi + \eta \quad (55) \]

with:

\[ \phi(0) = \eta / \lambda \quad (56) \]
4 APPLICATION OF MODEL-FREE SMC TO LINEAR SYSTEMS

In this chapter, results of applying the model-free sliding mode controller to linear systems are presented to test the validity of the proposed method. A first-order and second-order linear system were simulated and results are presented with and without a smoothing boundary layer applied. For the first-order linear system a simple time constant linear system was assumed. For the second-order linear system a mass, spring, damper linear type system was used. Following the results is a section comparing the linear systems and a discussion of how the moving boundary layer affects the tracking performance of the linear systems. The model-free sliding mode controller is also compared to model-based sliding mode controllers regarding their ability to control linear systems.

4.1 Linear Sliding Mode Controller

The following sections show the results of applying the model-free sliding mode controller without a smoothing boundary layer to a first-order and a second-order linear system to test the feasibility of the proposed method.

4.1.1 First-Order Linear Example

The first-order linear system to be controlled was chosen to be:

\[ \dot{x} + 5x = u \]  \hspace{1cm} (57)

where \( x \) and \( \dot{x} \) are the state measurement variables of the system and \( u \) is the input to the system.

Using the control law defined in Eq. (43) and the switching gain defined in Eq. (50), the model-free control law is derived as the following:

\[ u = \dot{x}_d - \dot{x} - \lambda (x - x_d) + u_{n-1} - K \text{sgn}(s) \]  \hspace{1cm} (58)

Using the derived control law, sliding condition, and switching gain a control system was developed and the closed-loop system was simulated using Simulink and MATLAB for the first-order linear system described above. A fixed-sampling time of 0.0001 seconds was used and the system was simulated for 30 seconds. The value of \( \lambda \) was set to 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 19 displays the Simulink diagram of the sliding mode controller for the first-order linear system.
Figure 19: First-Order Linear Simulink Diagram of Sliding Mode Controller

Figure 20 displays the first-order linear system to be controlled.

Figure 20: Open Loop System
Figure 21 displays the “signum” or “sat” function used for the switching gain for the controller.

![Figure 21: Signum or Sat Function](image)

Figure 22 displays the desired tracking responses for the closed-loop system.

![Figure 22: Desired Tracking](image)

Figure 23 displays the position tracking error for the first-order linear system. The figure shows the controller is robust, with “perfect” tracking having a maximum tracking error ranging from $-1.2e-6$ to $1.2e-6$. However, the controller does produce a large amount of chattering. Figure 24 displays the position comparison of the desired and actual responses for the first-order linear system. Outstanding agreement is shown between the desired state and the actual state responses.
Figure 23: Position Tracking Error for First-Order Linear System

Figure 24: Position Comparison of Desired and Actual Responses for First-Order Linear System

Figure 25 displays the velocity tracking error for the first-order linear system. The figure shows the controller produces a maximum tracking error ranging from -0.1 to 0.1. However, the controller does produce a large amount of chattering. Figure 26 displays the velocity comparison of the desired and actual responses for the first-order linear system. The controller produced significant chatter in the system response.

Figure 26: Velocity Comparison of Desired and Actual Responses for First-Order Linear System

Figure 27 displays the sliding condition from Eq. (48) for the first-order linear system. The figure shows the sliding condition was satisfied at all times thus proving stability in the
Lyapunov sense. Figure 28 displays the control effort for the first-order linear system, which shows a significant amount of control chatter.

Figure 27: Sliding Condition for First-Order Linear System

Figure 28: Control Effort for First-Order Linear System

Figure 29 displays the phase plane for the first-order linear system and Figure 30 displays the phase plane of the sliding system for the first-order linear system. The phase plane plots show the asymptotic stability of the control law and near perfect performance tracking but at a price of unrealistic control effort chattering.

Figure 29: Phase Plane for First-Order Linear System

Figure 30: Phase Plane of the Sliding System for First-Order Linear System
The model-free sliding mode controller produced a closed-loop system that was asymptotically stable with near perfect tracking performance, but at a price of excessive chattering for the system responses of the first-order linear system. Control effort chattering is a common problem when using sliding mode control theory. A smoothing boundary layer can be applied to the controller to reduce the amount of chatter during the system response.

**4.1.2 Higher-Order Linear Example**

The higher-order linear system to be controlled was chosen to be:

\[
m\ddot{x} + c\dot{x} + kx = u \tag{59}
\]

where \( m \) is the mass of the system, \( c \) is the damping coefficient of the system, \( k \) is the spring constant of the system, \( u \) is the input to the system, and \( x, \dot{x}, \) and \( \ddot{x} \) are the state measurement variables of the system. The values used for the mass, damping coefficient, and spring constant were 2 (mass units), 0.8 (force units)/(length units)/sec, and 2 (force units)/(length units), respectively.

Using the control law defined in Eq. (43) and the switching gain defined in Eq. (50), the model-free control law is derived as the following:

\[
u = \ddot{x}_d - \ddot{x} - \lambda (\dot{x} - \dot{x}_d) + u_{n-1} - K\text{sgn}(s) \tag{60}
\]

Using the derived control law, sliding condition, and switching gain a control system was developed and a simulation was conducted using Simulink and MATLAB for the mass, spring, damper system above. The sampling time used was 0.0001 seconds and the system was simulated for 30 seconds. The value of \( \lambda \) used was 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 31 displays the Simulink diagram of the sliding mode controller for the second-order linear system.
Figure 31: Higher-order Linear Simulink Diagram of Sliding Mode Controller
Figure 32 displays the higher-order linear system to be controlled.

![Open Loop System Diagram](image)

**Figure 32: Open Loop System**

Figure 33 displays the “signum” or “sat” function for the control law.

![Signum or Sat Function Diagram](image)

**Figure 33: Signum or Sat Function**

Figure 34 displays the desired tracking responses for the closed-loop system.

![Desired Tracking Diagram](image)

**Figure 34: Desired Tracking**
Figure 35 displays the position tracking error for the closed-loop higher-order linear system. The figure shows the controller is robust with “perfect” tracking performance, having a maximum tracking error ranging from -2e-8 to 2e-8. However, the controller does produce a large amount of chattering to be shown next. Figure 36 displays the position comparison of the desired and actual responses for the higher-order linear system. The difference between the responses is negligible, proving the controller is robust with “perfect” tracking performance.

Figure 37 displays the velocity tracking error for the higher-order linear system. The figure shows the controller is robust, having a maximum tracking error ranging from -3.5e-6 to 3.5e-6. Figure 38 displays the velocity comparison of the desired and actual responses for the higher-order linear system. The difference between the responses is negligible, proving the controller is robust and showing the “perfect” tracking behavior of the closed-loop system.
Figure 37: Velocity Tracking Error for Higher-Order Linear System

Figure 38: Velocity Comparison of Desired and Actual Responses for Higher-Order Linear System

Figure 39 displays the acceleration tracking error for the higher-order linear system. The figure shows the controller is robust, having a maximum tracking error ranging from -0.09 to 0.09. However, the controller does produce a large amount of chattering. Figure 40 displays the acceleration comparison of the desired and actual responses for the higher-order linear system. The controller produced significant chatter in the system response.

Figure 39: Acceleration Tracking Error for Higher-Order Linear System

Figure 40: Acceleration Comparison of Desired and Actual Responses for Higher-Order Linear System
Figure 41 displays the sliding condition from Eq. (48) for the higher-order linear system. The figure shows the sliding condition was again satisfied at all times. Figure 42 displays the control effort for the higher-order linear system, which also experienced a significant amount of control law chatter.

Figure 43 displays the phase plane for the higher-order linear system and Figure 44 displays the phase plane of the sliding system for the higher-order linear system. The controller produced a smooth response for the phase plane of the higher-order linear system and proves the Lyapunov stability of the closed-loop system. However, the controller once again produced a significant amount of chattering for the response of the phase plane for the sliding system of the higher-order linear system.
As was with the first-order linear system, the model-free sliding mode controller produced a significant chattering for the system responses of the higher-order linear system. Therefore, a smoothing boundary layer should be applied to the controller to reduce the amount of chatter during the system response for implementation of the control law on an actual system.
4.2 Linear Sliding Mode Controller with Moving Boundary Layer

The following sections show the results of applying the model-free sliding mode controller with a smoothing boundary layer to a first-order and a second-order linear system to test the feasibility of the proposed method and to reduce the control effort chattering.

4.2.1 First-Order Linear Example with Moving Boundary

The first-order linear system to be controlled was chosen once again to be:
\[ \dot{x} + 5x = u \]  \hspace{1cm} (61)

where \( x \) and \( \dot{x} \) are the state measurement variables of the system and \( u \) is the input to the system.

Using the control law defined in Eq. (54) and the smoothing boundary layer defined in Eq. (55), the model-free control law is derived as the following:
\[ u = \dot{x}_d - \dot{x} - \lambda (x - x_d) + u_{n-1} - (K - \phi) \text{sat} \left( \frac{s}{\phi} \right) \]  \hspace{1cm} (62)

with
\[ \dot{\phi} = -\lambda \phi + \eta \]  \hspace{1cm} (63)

and
\[ \eta = K \]  \hspace{1cm} (64)

Using the derived control law, sliding condition, smoothing boundary layer, and switching gain a control system was developed and programmed in Simulink and MATLAB for the first-order linear system described above. A fixed-sampling time of 0.0001 seconds was once again used and the system was simulated for 30 seconds. The value of \( \lambda \) was set to 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 45 displays the Simulink diagram of the sliding mode controller with a moving boundary layer for the first-order linear system.
Figure 45: First-Order Linear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer

Figure 46 displays the first-order linear system to be controlled.
Figure 47 displays the updated “signum” or “sat” function for the model-free control law.

![Diagram of Signum or Sat Function]

Figure 47: Signum or Sat Function

Figure 48 displays the desired tracking responses for the closed-loop system.

![Diagram of Desired Tracking]

Figure 48: Desired Tracking

Figure 49 displays the position tracking error for the first-order linear system with a moving boundary layer. The figure shows the closed-loop tracking is adequate having a maximum tracking error ranging from -6e-5 to 6e-5. The moving boundary layer produced a completely smooth tracking error response. Figure 50 displays the position comparison of the desired and actual responses for the first-order linear system with a moving boundary layer. The difference between the responses is negligible, showing the controller is robust.
Figure 49: Position Tracking Error for First-Order Linear System with Moving Boundary Layer

Figure 50: Position Comparison of Desired and Actual Responses for First-Order Linear System with Moving Boundary Layer

Figure 51 displays the velocity tracking error for the first-order linear system with a moving boundary layer. The figure shows the controller is adequate producing a maximum velocity tracking error ranging from $-4\times10^{-4}$ to $4\times10^{-4}$. The moving boundary layer produces a smooth velocity tracking response. Figure 52 displays the velocity comparison of the desired and actual responses for the first-order linear system with a moving boundary layer. The difference between the responses is small, showing the robust nature of the control law.

Figure 51: Velocity Tracking Error for First-Order Linear System with Moving Boundary Layer

Figure 52: Velocity Comparison of Desired and Actual Responses for First-Order Linear System with Moving Boundary Layer
Figure 53 displays the boundary layer and sliding surface for the first-order linear system with a moving boundary layer. The figure proves the sliding surface remained within the boundary at all times thus satisfying the updated sliding condition and indicating closed-loop stability in the Lyapunov sense. Figure 54 displays the control effort for the first-order linear system with a moving boundary layer. The moving boundary layer produces a control effort with no chattering and one that can be implemented on a real-world system.

![Figure 53: Boundary Layer for First-Order Linear System with Moving Boundary Layer](image1)
![Figure 54: Control Effort for First-Order Linear System with Moving Boundary Layer](image2)

Figure 55 displays the phase plane for the first-order linear system with a moving boundary layer and Figure 56 displays the phase plane of the sliding system for the first-order linear system with a moving boundary layer and proves the Lyapunov stability of the closed-loop system. The moving boundary layer produced a completely smooth response for both planes.

![Figure 55: Phase Plane for First-Order Linear System with Moving Boundary Layer](image3)
![Figure 56: Phase Plane of Sliding System for First-Order Linear System with Moving Boundary Layer](image4)
The model-free sliding mode controller with a moving boundary layer produces smooth system responses of the first-order linear system with adequate closed-loop tracking performance and thus can be implemented in a real-world system. The closed-loop system was shown to be stable in the Lyapunov sense since the sliding condition was satisfied at all times.
4.2.2 Higher-Order Example with Moving Boundary

The higher-order linear system to be controlled was once again chosen to be:

\[ m\ddot{x} + c\dot{x} + kx = u \]  \hspace{1cm} (65)

where \( m \) is the mass of the system, \( c \) is the damping coefficient of the system, \( k \) is the spring constant of the system, \( u \) is the input to the system, and \( x \), \( \dot{x} \), and \( \ddot{x} \) are the state measurement variables of the system. The mass, damping coefficient, and spring constant values are 2 (mass units), 0.8 (force units)/(length units)/sec, and 2 (force units)/(length units), respectively.

Using the control law defined in Eq. (54) and the smoothing boundary layer defined in Eq. (55), the model-free control law is derived as following:

\[ u = \ddot{x}_d - \dot{x} - \lambda (\dot{x} - \dot{x}_d) + u_{n-1} - (K - \dot{\phi}) \text{sat} \left( \frac{\varepsilon}{\phi} \right) \]  \hspace{1cm} (66)

with

\[ \dot{\phi} = -\lambda \phi + \eta \]  \hspace{1cm} (67)

and

\[ \eta = K \]  \hspace{1cm} (68)

Using the derived control law, sliding condition, smoothing boundary layer, and switching gain a control system was developed in Simulink and MATLAB for first-order linear system described above. The sampling time used was 0.0001 seconds and the system was simulated for 30 seconds. The value of \( \lambda \) used was 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 57 displays the Simulink diagram of the sliding mode controller with a moving boundary layer for the higher-order nonlinear system.
Figure 57: Higher-order Linear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer

Figure 58 displays higher-order linear system to be controlled.

Figure 58: Open Loop System
Figure 59 displays the updated “signum” or “sat” function for the controller.

![Figure 59: Signum or Sat Function](image)

Figure 60 displays the desired tracking responses for the closed-loop system.

![Figure 60: Desired Tracking](image)

Figure 61 displays the position tracking error for the higher-order linear system with a moving boundary layer. The figure indicates adequate tracking performance is achieved with a maximum position tracking error ranging from $-4e-6$ to $13e-6$. The moving boundary layer produced a completely smooth response. Figure 62 displays the position comparison of the desired and actual responses for the higher-order linear system with a moving boundary layer. The difference between the responses is small, indicating once again the controller is robust.
Figure 63 displays the velocity tracking error for the higher-order linear system with a moving boundary layer. The figure shows the velocity tracking error is small having a maximum velocity tracking error ranging from $-1.25e-5$ to $1.25e-5$. Also, the moving boundary layer produces a “smooth” response. Figure 64 displays the velocity comparison of the desired and actual responses for the higher-order linear system with a moving boundary layer. The difference between the responses is negligible, proving the controller is robust for the velocity comparison.
Figure 65 displays the acceleration tracking error for the higher-order linear system with a moving boundary layer. The maximum tracking error is -4e-4 to 4e-4 indicating adequate tracking performance is achieved. The moving boundary layer produced a “smooth” response for the acceleration error responses as well. Figure 66 displays the acceleration comparison of the desired and actual responses for the higher-order linear system with a moving boundary layer. The difference between the responses is negligible, proving the controller is robust.

Figure 67 displays the boundary layer and sliding surface for the higher-order linear system with a moving boundary layer. The figure proves the sliding surface remained within the boundary at all times thus satisfying the updated sliding condition proving the closed-loop system is stable in the Lyapunov sense. Figure 68 displays the control effort for the higher-order linear system with a moving boundary layer. The moving boundary layer produces a control effort that does not chatter and is realistic for implementation in a real-world system.
Figure 67: Boundary Layer for Higher-Order Linear System with Moving Boundary Layer

Figure 68: Control Effort for Higher-Order Linear System with Moving Boundary Layer

Figure 69 displays the phase plane for the higher-order linear system with a moving boundary layer and Figure 70 displays the phase plane of the sliding system for the higher-order linear system with a moving boundary layer and proves the Lyapunov stability of the closed-loop system.

Figure 69: Phase Plane for Higher-Order Linear System with Moving Boundary Layer

Figure 70: Phase Plane of Sliding System for Higher-Order Linear System with Moving Boundary Layer

The model-free sliding mode controller with a moving boundary layer produces “smooth” system responses of the higher-order linear system with a moving boundary layer similar in performance to the linear first-order system.
4.3 Discussion of Results

The following sections discuss the results of the application of the model-free sliding mode controller to linear systems. The first-order linear system is compared against the higher-order linear system with and without a moving boundary layer applied. The effects of the moving boundary layer on the first-order linear system and on the higher-order linear system are also discussed. Finally, the model-free sliding mode controller is compared against model-based sliding mode controllers regarding their ability to control linear systems.

4.3.1 First-Order Linear System vs. Higher-Order Linear System

The following sections compare the first-order linear system and higher-order linear system with and without a moving boundary layer applied.

4.3.1.1 Without Moving Boundary Layer

Comparing the position tracking error of the first-order linear system, Figure 23, and the position tracking error of the higher-order linear system, Figure 35, both responses experienced small error with chatter present. The first-order linear system experienced more chatter in respect to the higher-order system. The extreme low tracking error is reflected in the position comparison of the actual and desired responses of both the first and higher-order linear systems as shown in Figure 24 and Figure 36. There is a negligible difference between the actual and desired responses of both the first and higher-order linear systems.

Comparing the velocity tracking error of the first-order linear system, Figure 25, and the position tracking error of the higher-order linear system, Figure 37, both responses experienced significant chatter. However, in comparison to the velocity tracking error of the first-order linear system, the velocity tracking error of the higher-order linear system experienced very little error. The velocity tracking error for the higher-order linear system was between -4e-6 and 4e-6, whereas the velocity tracking error for the first-order linear system was between -0.1 and 0.1. The large difference in error is reflected in the velocity comparison of the actual and desired responses for the first and higher-order linear systems as shown in Figure 26 and Figure 37. The actual velocity response of the first-order linear system experienced significant chatter, making a large difference in comparison to the desired response whereas there is a negligible difference between the actual and desired responses of the higher-order linear system.
The acceleration tracking error for the higher-order linear system, Figure 39, behaved similarly to the velocity tracking error for the first-order linear system, Figure 25. Both experienced higher error than other responses of their respective systems and significant chatter. The acceleration comparison of the desired and actual responses for the higher-order linear system, Figure 39, also behaved similarly to the velocity comparison of the desired and actual responses for the first-order linear system, Figure 26. Both actual responses experienced significant chatter, producing large differences in comparison to the desired responses. The pattern indicates that as the order of the system increases, the highest state response of the system will experience more error and chatter in comparison to the other state responses.

Both the sliding condition of the first-order linear system, Figure 27, and the sliding condition of the higher-order linear system, Figure 41, experienced a significant amount of chatter. The high volume of chatter makes determining if the sliding condition is constantly satisfied difficult for both the first and higher-order linear systems. However, the responses are very close to zero so that the sliding condition can be assumed to be satisfied. Also, both the control effort of the first-order linear system, Figure 28, and the control effort of the higher-order linear system, Figure 42, experienced a significant amount of chatter.

Comparing the phase plane of the first-order linear system, Figure 29, and the phase plane of the higher-order linear system, Figure 43, the phase plane of the first-order linear system experienced significant chatter. Whereas the phase plane of the higher-order linear system experienced a “smooth” response. However, both the phase plane for the sliding system of the first-order linear system, Figure 30, and the phase plane for the sliding system of the higher-order linear system, Figure 44, experienced significant chatter.

4.3.1.2 With Moving Boundary Layer

Comparing the position tracking error of the first-order linear system with a moving boundary layer, Figure 49, and the position tracking error of the higher-order linear system with a moving boundary layer, Figure 61, both responses experienced small error with no chatter. The small error is reflected in the position comparison of the actual and desired responses of both the first and higher-order linear systems with a moving boundary layer, Figure 50 and Figure 62. There is
a negligible difference between the actual and desired responses of both the first and higher-order linear systems with a moving boundary smoothing layer.

Comparing the velocity tracking error of the first-order linear system with a moving smoothing boundary layer, Figure 51, and the velocity tracking error of the higher-order linear system with a moving boundary layer, Figure 63, both responses experienced small error with no chatter. The relatively small error is reflected in the velocity comparison of the actual and desired responses of both the first and higher-order linear systems with a moving boundary layer, Figure 52 and Figure 64. There is a negligible difference between the actual and desired responses of both the first and higher-order linear systems with moving boundary layers.

The acceleration tracking error of the higher-order linear system with a moving boundary layer, Figure 65, continued the trend. The tracking error response had small error with no chatter. The relatively small error is reflected in the acceleration comparison of the actual and desired responses of higher-order linear system with a moving boundary layer, Figure 66. There is a negligible difference between the actual and desired response of the higher-order linear system with a moving boundary layer.

Both the sliding surface of the first-order linear system with a moving boundary layer, Figure 53, and the sliding surface of the higher-order linear system with a moving boundary layer, Figure 67, remained within the boundary limits proving that the closed-loop system in stable in the Lyapunov sense since the updated sliding condition is satisfied. Also, both the control effort of the first-order linear system with a moving boundary layer, Figure 54, and the control effort of the higher-order linear system with a moving boundary layer, Figure 68, experienced no chatter.

Both the phase plane of the first-order linear system with a moving boundary layer, Figure 55, and the phase plane of the higher-order linear system with a moving boundary layer, Figure 69, experienced no chatter. Also, both the phase plane for the sliding system of the first-order linear system with a moving boundary layer, Figure 56, and the phase plane for the sliding system of the higher-order linear system with a moving boundary layer, Figure 70, experienced no chatter. The smoothing boundary layer removed all chatter from the system responses and enabled the robust control of the model-free sliding mode controller.
4.3.2 Impacts of a Moving Boundary Layer

Applying a smoothing boundary layer removed all chatter from the system responses. A detailed explanation of the comparison for both the first-order linear systems and higher-order linear systems is provided in the following sections.

4.3.2.1 First-Order Linear System

Comparing the position tracking error of the first-order linear systems, Figure 23 and Figure 49, clearly shows the moving smoothing boundary layer removes all chatter from the position tracking error. Also, there is a negligible difference between the position comparison of the actual and desired responses for the first-order linear systems, Figure 24 and Figure 50.

The moving boundary layer did have a large impact on the velocity tracking error and velocity comparison of the actual and desired values for the first-order linear systems. Comparing the velocity tracking error of the first-order linear systems, Figure 25 and Figure 51, clearly shows the moving boundary layer removes all chatter and reduced the velocity tracking error to a negligible amount. The result is demonstrated in the velocity comparisons of the first-order linear systems, Figure 26 and Figure 51. Figure 26 displays significant chatter in the system response, creating a large error between the desired and actual response whereas in Figure 64, the smoothing boundary layer removed the chatter, making the difference between the desired and actual response negligible.

Examining Figure 28 and Figure 54, it is clear the smoothing boundary layer removes the chatter on the control effort. Observing the phase planes for the system of the first-order linear systems, Figure 29 and Figure 55, and the phase planes for the sliding system of the first-order linear systems, Figure 30 and Figure 56, the smoothing boundary layer removes all chatter for both phase planes creating clearly defined phase planes. The smoothing boundary layer removed all chatter from the system responses and enables the robust control of the model-free sliding mode controller.

4.3.2.2 Higher-Order Linear System

Comparing the position tracking error of the higher-order linear systems, Figure 35 and Figure 61, indicates the moving boundary layer removes all chatter from the position tracking error.
Also, there is a negligible difference between the position comparison of the actual and desired values for the higher-order linear systems, Figure 36 and Figure 62, showing adequate tracking performance is achieved. Comparing the velocity tracking error of the higher-order linear systems, Figure 37 and Figure 63, shows the moving boundary layer removed all chatter and reduced the velocity tracking error. Also, there is a negligible difference between the velocity comparison of the actual and desired values for the higher-order linear systems, Figure 38 and Figure 64.

The moving boundary layer has a significant impact on the acceleration tracking error and acceleration comparison of the actual and desired values for the higher-order linear systems. Comparing the acceleration tracking error of the higher-order linear systems, Figure 39 and Figure 65, clearly shows the moving boundary layer removed all chatter and reduced the acceleration tracking error to a negligible amount. Once again the result is demonstrated in the acceleration comparisons of the higher-order linear systems, Figure 40 and Figure 66. Figure 40 displays significant chatter in the system response, creating a large error between the desired and actual response whereas in Figure 66, the smoothing boundary layer removed the chatter, which forces the difference between the desired and actual responses to be negligible.

Comparing the sliding condition of the higher-order linear systems, Figure 41 and Figure 67, the benefits of the smoothing boundary layer are clear. Determining if the sliding condition was satisfied in Figure 41 is difficult due to the high amount of chatter but once again is identically zero and thus the sliding condition is satisfied proving the stability nature of the closed-loop system. Whereas in Figure 67, the chatter was removed by using the smoothing boundary layer and the sliding condition is clearly satisfied. Examining Figure 42 and Figure 68, it is clear the smoothing boundary layer also removed the chatter on the control effort. Observing the phase planes for the system of the higher-order linear systems, Figure 43 and Figure 69, the smoothing boundary layer had little effect on tracking performance. However, observing the phase portraits for the sliding system of the higher-order linear systems, Figure 44 and Figure 70, the smoothing boundary layer removed all chatter creating a clearly defined phase plane. The smoothing boundary layer removes all chatter from the system responses and enabled the robust control of the model-free sliding mode controller.
5 APPLICATION OF MODEL-FREE SMC TO NONLINEAR SYSTEMS

In this chapter, results of applying the model-free sliding mode controller to nonlinear systems are presented to test the validity of the proposed method. First-order and second-order nonlinear systems were simulated and results are presented with and without a smoothing boundary layer applied. For the first-order nonlinear system a simple time constant nonlinear system was assumed. For the second-order nonlinear system a mass, spring, damper nonlinear type system was used. Following the results is a section comparing the nonlinear systems and a discussion of how the moving boundary layer affects the tracking performance of the nonlinear systems. The model-free sliding mode controller is also compared to model-based sliding mode controllers regarding their ability to control nonlinear systems.

5.1 Nonlinear Sliding Mode Controller

The following sections show the results of applying the model-free sliding mode controller without a smoothing boundary layer to a first-order and a second-order nonlinear system to test the feasibility of the proposed method.

5.1.1 First-Order Nonlinear Example

The first-order nonlinear system to be controlled was chosen to be:

\[ \dot{x} - 5x^2 = u \]  \hspace{1cm} (69)

where x and \( \dot{x} \) are the state measurement variables of the system and u is the input to the system.

Using the control law defined in Eq. (43) and the switching gain defined in Eq. (50), the model-free control law is derived as the following:

\[ u = \dot{x}_d - \dot{x} - \lambda (x - x_d) + u_{n-1} - K \text{sgn}(s) \]  \hspace{1cm} (70)

Using the derived control law, sliding condition, and switching gain a control system was developed and the closed-loop system was simulated using Simulink and MATLAB for the first-order linear system described above. A fixed-sampling time of 0.0001 seconds was used and the system was simulated for 30 seconds. The value of \( \lambda \) was set to 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 71 displays the Simulink diagram of the sliding mode controller for the first-order nonlinear system.
Figure 71: First-Order Nonlinear Simulink Diagram of Sliding Mode Controller

Figure 72 displays the first-order nonlinear system to be controlled.

Figure 72: Open Loop System
Figure 73 displays the “signum” or “sat” function for the controller.

![Signum or Sat Function](image)

Figure 73: Signum or Sat Function

Figure 74 displays the desired tracking responses for the closed-loop system.

![ Desired Tracking](image)

Figure 74: Desired Tracking

Figure 75 displays the position tracking error for the first-order nonlinear system. The figure shows the controller is robust, with “perfect” tracking having a maximum tracking error ranging from -1.5e-6 to 1.5e-6. However, the controller does produce a large amount of chattering.

Figure 76 displays the position comparison of the desired and actual responses for the first-order nonlinear system. Outstanding agreement is shown between the desired state and the actual state responses.
Figure 75: Position Tracking Error for First-Order Nonlinear System

Figure 76: Position Comparison of Desired and Actual Responses for First-Order Nonlinear System

Figure 77 displays the velocity tracking error for the first-order nonlinear system. The figure shows the controller produces a maximum tracking error ranging from –0.1 to 0.1. However, the controller does produce a large amount of chattering. Figure 78 displays the velocity comparison of the desired and actual responses for the first-order nonlinear system. The controller produced significant chatter in the system response.
Figure 79 displays the sliding condition from Eq. (48) for the first-order nonlinear system. The figure shows the sliding condition was satisfied at all times, thus proving stability in the Lyapunov sense. Figure 80 displays the control effort for the first-order nonlinear system, which shows a significant amount of control chatter.

Figure 81 displays the phase plane for the first-order nonlinear system and Figure 82 displays the phase plane of the sliding system for the first-order nonlinear system. The phase plane plots show the asymptotic stability of the control law and near perfect performance tracking but at a price of unrealistic control effort chattering.
The model-free sliding mode controller produced a closed-loop system that was asymptotically stable with near perfect tracking performance but at a price of excessive chattering for the system responses of the first-order nonlinear system. Control effort chattering is a common problem when using sliding mode control theory. A smoothing boundary layer can be applied to the controller to reduce the amount of chatter during the system response.

5.1.2 Higher-Order Nonlinear Example

The higher-order nonlinear system to be controlled was chosen to be:

\[ m\ddot{x} + c\dot{x}^2 + kx = u \]  

(71)

where \( m \) is the mass of the system, \( c \) is the damping coefficient of the system, \( k \) is the spring constant of the system, \( u \) is the input to the system, and \( x, \dot{x}, \) and \( \ddot{x} \) are the state measurement variables of the system. The mass, damping coefficient, and spring constant values are 2 (mass units), 0.8 (force units)/(length units)/sec, and 2 (force units)/(length units), respectively.

Using the control law defined in Eq. (43) and the switching gain defined in Eq. (50), the model-free control law is derived as following:

\[ u = \ddot{x}_d - \dot{x} - \lambda (\dot{x} - \dot{x}_d) + u_{n-1} - K\text{sgn}(s) \]  

(72)

Using the derived control law, sliding condition, and switching gain a control system was developed and a simulation was conducted using Simulink and MATLAB for the mass, spring, damper system above. The sampling time used was 0.0001 seconds and the system was simulated for 30 seconds. The value of \( \lambda \) used was 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 83 displays the Simulink diagram of the sliding mode controller for the higher-order linear system.
Figure 83: Higher-order Nonlinear Simulink Diagram of Sliding Mode Controller

Figure 84 displays the higher-order nonlinear system to be controlled.
Figure 85 displays the “signum” or “sat” function for the control law.

Figure 85: Signum or Sat Function

Figure 86 displays the desired tracking responses for the closed-loop system.

Figure 86: Desired Tracking

Figure 87 displays the position tracking error for the closed-loop higher-order nonlinear system. The figure shows the controller is robust with “perfect” tracking performance, having a maximum tracking error ranging from $-2.2e-8$ to $1.5e-8$. However, the controller does produce a large amount of chattering to be shown next. Figure 88 displays the position comparison of the desired and actual responses for the higher-order nonlinear system. The difference between the responses is negligible, proving the controller is robust with “perfect tracking performance.
Figure 89 displays the velocity tracking error for the higher-order nonlinear system. The figure shows the controller is robust, having a maximum tracking error ranging from -3.5e-6 to 3.5e-6. Figure 90 displays the velocity comparison of the desired and actual responses for the higher-order nonlinear system. The difference between the responses is negligible, proving the controller is robust and showing the “perfect” tracking behavior of the closed-loop system.
Figure 91 displays the acceleration tracking error for the higher-order nonlinear system. The figure shows the controller is robust, having a maximum tracking error ranging from –0.09 to 0.09. However, the controller does produce a large amount of chattering. Figure 92 displays the acceleration comparison of the desired and actual responses for the higher-order nonlinear system. The controller produced significant chatter in the system response.

Figure 93 displays the sliding condition from Eq. (48) for the first-order nonlinear system. The figure shows the sliding condition was again satisfied at all times. Figure 94 displays the control effort for the first-order nonlinear system, which also experienced a significant amount of control law chatter.
Figure 93: Sliding Condition for Higher-Order Nonlinear System

Figure 94: Control Effort for Higher-Order Nonlinear System

Figure 95 displays the phase plane for the higher-order nonlinear system and Figure 96 displays the phase plane of the sliding system for the higher-order nonlinear system. The controller produced a smooth response for the phase plane of the higher-order nonlinear system and proves the Lyapunov stability of the closed-loop system. However, the controller once again produced a significant amount of chattering for the response of the phase plane for the sliding system of the higher-order nonlinear system.

Figure 95: Phase Plane for Higher-Order Nonlinear System

Figure 96: Phase Plane of the Sliding System for Higher-Order Nonlinear System

As was with the first-order nonlinear system, the model-free sliding mode controller produced significant chattering for the system responses of the higher-order nonlinear system. Therefore, a
smoothing boundary layer should be applied to the controller to reduce the amount of chatter during the system response for implementation of the control law on an actual system.

5.2 Nonlinear Sliding Mode Controller with Moving Boundary Layer

The following sections show the results of applying the model-free sliding mode controller with a smoothing boundary layer to a first-order and a second-order nonlinear system to test the feasibility of the proposed method and to reduce the control effort chattering.

5.2.1 First-Order Nonlinear Example with Moving Boundary Layer

The first-order nonlinear system to be controlled was chosen once gain to be:

\[ \dot{x} - 5x^2 = u \]  
(73)

where \( x \) and \( \dot{x} \) are the state measurement variables of the system and \( u \) is the input to the system.

Using the control law defined in Eq. (54) and the smoothing boundary layer defined in Eq. (55), the model-free control law is derived as following:

\[ u = \dot{x}_d - \dot{x} - \lambda (x - x_d) + u_{n-1} - (K - \dot{\phi}) \text{sat}\left(\frac{s}{\dot{\phi}}\right) \]  
(74)

with

\[ \dot{\phi} = -\lambda \phi + \eta \]  
(75)

and

\[ \eta = K \]  
(76)

Using the derived control law, sliding condition, smoothing boundary layer, and switching gain a control system was developed and programmed in Simulink and MATLAB for the first-order linear system described above. A fixed-sampling time of 0.0001 seconds was once again used and the system was simulated for 30 seconds. The value of \( \lambda \) was set to 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 97 displays the Simulink diagram of the sliding mode controller with a moving boundary layer for the first-order nonlinear system.
Figure 97: First-Order Nonlinear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer

Figure 98 displays the first-order nonlinear system to be controlled.

Figure 98: Open Loop System
Figure 99 displays the updated “signum” or “sat” function for the model-free control law.

![Diagram of the updated “signum” or “sat” function](image)

**Figure 99: Signum or Sat Function**

Figure 100 displays the desired tracking responses for the closed-loop system.

![Diagram of desired tracking responses](image)

**Figure 100: Desired Tracking**

Figure 101 displays the position tracking error for the first-order nonlinear system with a moving boundary layer. The figure shows the closed-loop tracking is adequate having a maximum tracking error ranging from -6e-5 to 6e-5. The moving boundary layer produced a completely smooth tracking error response. Figure 102 displays the position comparison of the desired and actual responses for the first-order nonlinear system with a moving boundary layer. The difference between the responses is negligible, showing the controller is robust.

![Diagram of position tracking error](image)
Figure 101 displays the velocity tracking error for the first-order nonlinear system with a moving boundary layer. The figure shows the controller is adequate producing a maximum velocity tracking error ranging from −6e-4 to 6e-4. The moving boundary layer produces a smooth response. Figure 104 displays the velocity comparison of the desired and actual responses for the first-order nonlinear system with a moving boundary layer. The difference between the responses is small, showing the robust nature of the control law.
Figure 105 displays the boundary layer and sliding surface for the first-order nonlinear system with a moving boundary layer. The figure proves the sliding surface remained within the boundary at all times thus satisfying the updated sliding condition and indicating closed-loop stability in the Lyapunov sense. Figure 106 displays the control effort for the first-order nonlinear system with a moving boundary layer. The moving boundary layer produces a control effort with no chattering and one that can be implemented on a real-world system.

Figure 107 displays the phase plane for the first-order nonlinear system with a moving boundary layer and Figure 108 displays the phase plane of the sliding system for the first-order nonlinear system with a moving boundary layer and proves the Lyapunov stability of the closed-loop system. The moving boundary layer produced a completely smooth response for both planes.
The model-free sliding mode controller with a moving boundary layer produces smooth system responses of the first-order nonlinear system with adequate closed-loop tracking performance and thus can be implemented in a real-world system. The closed-loop system was shown to be stable in the Lyapunov sense since the sliding condition was satisfied at all time.
5.2.2 Higher-Order Nonlinear Example with Moving Boundary Layer

The higher-order nonlinear system to be controlled was once again chosen to be:

\[ m \ddot{x} + c \dot{x}^2 + kx = u \]  \hspace{1cm} (77)

where \( m \) is the mass of the system, \( c \) is the damping coefficient of the system, \( k \) is the spring constant of the system, \( u \) is the input to the system, and \( x, \dot{x}, \) and \( \ddot{x} \) are the state measurement variables of the system. The mass, damping coefficient, and spring constant values are 2 (mass units), 0.8 (force units)/(length units)/sec, and 2 (force units)/(length units), respectively.

Using the control law defined in Eq. (54) and the smoothing boundary layer defined in Eq. (55), the model-free control law is derived as following:

\[ u = \dot{x}_d - \dot{x} - \lambda (x - x_d) + u_{n-1} - (K - \phi) \text{sat}\left(\frac{s}{\phi}\right) \]  \hspace{1cm} (78)

with

\[ \dot{\phi} = -\lambda \phi + \eta \]  \hspace{1cm} (79)

and

\[ \eta = K \]  \hspace{1cm} (80)

Using the derived control law, sliding condition, smoothing boundary layer, and switching gain a control system was developed in Simulink and MATLAB for first-order linear system described above. The sampling time used was 0.0001 seconds and the system was simulated for 30 seconds. The value of \( \lambda \) used was 20 (rad/sec) and the value of \( \eta \) used was 0.1. The desired tracking of the system was \( x_d(t) = \sin(\pi/2) \). Figure 109 displays the Simulink diagram of the sliding mode controller with a moving boundary layer for the higher-order nonlinear system.
Figure 109: Higher-order Nonlinear Simulink Diagram of Sliding Mode Controller with Moving Boundary Layer

Figure 110 displays the higher-order nonlinear system to be controlled.

Figure 110: Open Loop System
Figure 111 displays the updated “signum” or “sat” function for the controller.

![Signum or Sat Function](image1.png)

Figure 111: Signum or Sat Function

Figure 112 displays the desired tracking responses for the closed-loop system.

![Desired Tracking](image2.png)

Figure 112: Desired Tracking

Figure 113 displays the position tracking error for the higher-order nonlinear system with a moving boundary layer. The figure indicates adequate tracking performance is achieved with a maximum position tracking error ranging from $-7e-6$ to $8e-6$. The moving boundary layer produced a completely smooth response. Figure 114 displays the position comparison of the desired and actual responses for the higher-order nonlinear system with a moving boundary layer. The difference between the responses is small, indicating once again the controller is robust.
Figure 113: Position Tracking Error for Higher-Order Nonlinear System with Moving Boundary Layer

Figure 114: Position Comparison of Desired and Actual Responses for Higher-Order Nonlinear System with Moving Boundary Layer

Figure 115 displays the velocity tracking error for the higher-order nonlinear system with a moving boundary layer. The figure shows the velocity tracking error is small having a maximum velocity tracking error ranging from -2.25e-5 to 2.25e-5. Also, the moving boundary layer produces a "smooth" response. Figure 116 displays the velocity comparison of the desired and actual responses for the higher-order nonlinear system with a moving boundary layer. The difference between the responses is negligible, proving the controller is robust for the velocity comparison.
Figure 117 displays the acceleration tracking error for the higher-order nonlinear system with a moving boundary layer. The maximum acceleration tracking error is $-3.5 \times 10^{-4}$ to $4.5 \times 10^{-4}$ indicating adequate tracking performance is achieved. The moving boundary layer produced a “smooth” response for the acceleration error responses as well. Figure 118 displays the acceleration comparison of the desired and actual responses for the higher-order nonlinear system with a moving boundary layer. The difference between the responses is negligible, proving the controller is robust.

Figure 119 displays the boundary layer and sliding surface for the higher-order nonlinear system with a moving boundary layer. The figure proves the sliding surface remained within the boundary at all times thus satisfying the updated sliding condition proving the closed-loop system is stable in the Lyapunov sense. Figure 120 displays the control effort for the higher-order nonlinear system with a moving boundary layer. The moving boundary layer produces a control effort that does not chatter and is realistic for implementation in a real-world system.
Figure 119: Boundary Layer for Higher-Order Nonlinear System with Moving Boundary Layer

Figure 120: Control Effort for Higher-Order Nonlinear System with Moving Boundary Layer

Figure 121 displays the phase plane for the higher-order nonlinear system with a moving boundary layer and Figure 122 displays the phase plane of the sliding system for the higher-order nonlinear system with a moving boundary layer and proves the Lyapunov stability of the closed-loop system.

Figure 121: Phase Plane for Higher-Order Nonlinear System with Moving Boundary Layer

Figure 122: Phase Plane of Sliding System for Higher-Order Nonlinear System with Moving Boundary Layer

The model-free sliding mode controller with a moving boundary layer produces “smooth” system responses of the higher-order nonlinear system with a moving boundary layer similar in performance to the nonlinear first-order system.
5.3 Discussion of Results

The following sections discuss the results of the application of the model-free sliding mode controller to nonlinear systems. The first-order nonlinear system is compared against the higher-order nonlinear system with and without a moving boundary layer applied. The effects of the moving boundary layer on the first-order nonlinear system and on the higher-order nonlinear system are also discussed. Finally, the model-free sliding mode controller is compared against model-based sliding mode controllers regarding their ability to control nonlinear systems.

5.3.1 First-Order Nonlinear System vs. Higher-Order Nonlinear System

The following sections compare the first-order nonlinear system and higher-order nonlinear system with and without a moving boundary layer applied.

5.3.1.1 Without Moving Boundary Layer

Comparing the position tracking error of the first-order nonlinear system, Figure 75, and the position tracking error of the higher-order nonlinear system, Figure 87, both responses experienced small error with chatter present. The first-order nonlinear system experienced more chatter in respect to the higher-order system. The extreme low tracking error is reflected in the position comparison of the actual and desired responses of both the first and higher-order nonlinear systems as shown in Figure 76 and Figure 88. There is a negligible difference between the actual and desired responses of both the first and higher-order nonlinear systems.

Comparing the velocity tracking error of the first-order nonlinear system, Figure 77, and the position tracking error of the higher-order nonlinear system, Figure 89, both responses experienced significant chatter. However, in comparison to the velocity tracking error of the first-order nonlinear system, the velocity tracking error of the higher-order nonlinear system experienced very little error. The velocity tracking error for the higher-order nonlinear system was between -3.5e-6 and 3.5e-6, whereas the velocity tracking error for the first-order nonlinear system was between -0.1 and 0.1. The large difference in error is reflected in the velocity comparison of the actual and desired responses for the first and higher-order nonlinear systems as shown in Figure 78 and Figure 90. The actual velocity response of the first-order nonlinear system experienced significant chatter, making a large difference in comparison to the desired
response whereas there is a negligible difference between the actual and desired responses of the higher-order nonlinear system.

The acceleration tracking error for the higher-order nonlinear system, Figure 91, behaved similarly to the velocity tracking error for the first-order nonlinear system, Figure 77. Both experienced higher error than other responses of their respective systems and significant chatter. The acceleration comparison of the desired and actual responses for the higher-order nonlinear system, Figure 92, also behaved similarly to the velocity comparison of the desired and actual responses for the first-order nonlinear system, Figure 78. Both actual responses experienced significant chatter, producing large differences in comparison to the desired responses. The pattern indicates that as the order of the system increases, the highest state response of the system will experience more error and chatter in comparison to the other state responses.

Both the sliding condition of the first-order nonlinear system, Figure 79, and the sliding condition of the higher-order nonlinear system, Figure 93, experienced a significant amount of chatter. The high volume of chatter makes determining if the sliding condition is constantly satisfied difficult for both the first and higher-order nonlinear systems. However, the responses are very close to zero so that the sliding condition can be assumed to be satisfied. Also, both the control effort of the first-order nonlinear system, Figure 80, and the control effort of the higher-order nonlinear system, Figure 94, experienced a significant amount of chatter.

Comparing the phase plane of the first-order nonlinear system, Figure 81, and the phase plane of the higher-order nonlinear system, Figure 95, the phase plane of the first-order nonlinear system experienced significant chatter. Whereas the phase plane of the higher-order nonlinear system experienced a smooth response. However, both the phase plane for the sliding system of the first-order nonlinear system, Figure 82, and the phase plane for the sliding system of the higher-order nonlinear system, Figure 96, experienced significant chatter.

5.3.1.2 With Moving Boundary Layer

Comparing the position tracking error of the first-order nonlinear system with a moving boundary layer, Figure 101, and the position tracking error of the higher-order nonlinear system with a moving boundary layer, Figure 113, both responses experienced small error with no chatter. The small error is reflected in the position comparison of the actual and desired
responses of both the first and higher-order nonlinear systems with a moving boundary layer, Figure 102 and Figure 114. There is a negligible difference between the actual and desired responses of both the first and higher-order nonlinear systems with a moving boundary layer.

Comparing the velocity tracking error of the first-order nonlinear system with a moving smoothing boundary layer, Figure 103, and the velocity tracking error of the higher-order nonlinear system with a moving boundary layer, Figure 115, both responses experienced small error with no chatter. The relatively small error is reflected in the velocity comparison of the actual and desired responses of both the first and higher-order nonlinear systems with a moving boundary layer, Figure 104 and Figure 116. There is a negligible difference between the actual and desired responses of both the first and higher-order nonlinear systems with a moving boundary layer.

The acceleration tracking error of the higher-order nonlinear system with a moving boundary layer, Figure 117, continued the trend. The tracking error response had small error with no chatter. The relatively small error is reflected in the acceleration comparison of the actual and desired responses of higher-order nonlinear system with a moving boundary layer, Figure 118. There is a negligible difference between the actual and desired response of the higher-order nonlinear system with a moving boundary layer.

Both the sliding surface of the first-order nonlinear system with a moving boundary layer, Figure 105, and the sliding surface of the higher-order nonlinear system with a moving boundary layer, Figure 119, remained within the boundary limits proving that the closed-loop system is stable in the Lyapunov sense since the updated sliding condition is satisfied. Also, both the control effort of the first-order nonlinear system with a moving boundary layer, Figure 106, and the control effort of the higher-order nonlinear system with a moving boundary layer, Figure 120, experienced no chatter.

Both the phase plane of the first-order nonlinear system with a moving boundary layer, Figure 107, and the phase plane of the higher-order nonlinear system with a moving boundary layer, Figure 121, experienced no chatter. Also, both the phase plane for the sliding system of the first-order nonlinear system with a moving boundary layer, Figure 108, and the phase plane for the sliding system of the higher-order nonlinear system with a moving boundary layer, Figure 122,
experienced no chatter. The smoothing boundary layer removed all chatter from the system responses and enabled the robust control of the model-free sliding mode controller.

### 5.3.2 Impacts of a Moving Boundary Layer

Applying a smoothing boundary layer removed all chatter from the system responses. A detailed explanation of the comparison for both the first-order nonlinear systems and higher-order nonlinear systems is provided in the following sections.

#### 5.3.2.1 First-order Nonlinear System

Comparing the position tracking error of the first-order nonlinear systems, Figure 75 and Figure 101, clearly shows the moving smoothing boundary layer removes all chatter from the position tracking error. Also, there is a negligible difference between the position comparison of the actual and desired values for the first-order nonlinear systems, Figure 76 and Figure 102.

The moving boundary layer did have a large impact on the velocity tracking error and velocity comparison of the actual and desired values for the first-order nonlinear systems. Comparing the velocity tracking error of the first-order nonlinear systems, Figure 77 and Figure 103, it clearly shows the moving boundary layer removes all chatter and reduced the velocity tracking error to a negligible amount. The result is demonstrated in the velocity comparisons of the first-order nonlinear systems, Figure 78 and Figure 104. Figure 78 displays significant chatter in the system response, creating a large error between the desired and actual response whereas in Figure 104, the smoothing boundary layer removed the chatter, making the difference between the desired and actual response negligible.

Examining Figure 80 and Figure 106, it is clear the smoothing boundary layer also removes the chatter on the control effort. Observing the phase planes for the system of the first-order nonlinear systems, Figure 81 and Figure 107, and the phase planes for the sliding system of the first-order nonlinear systems, Figure 82 and Figure 108, the smoothing boundary layer removes all chatter for both phase planes creating clearly defined phase planes. The smoothing boundary layer removed all chatter from the system responses and enables the robust control of the model-free sliding mode controller.
5.3.2.2 Higher-Order Nonlinear System

Comparing the position tracking error of the higher-order nonlinear systems, Figure 87 and Figure 113, indicates the moving boundary layer removes all chatter from the position tracking error. Also, there is a negligible difference between the position comparison of the actual and desired values for the higher-order nonlinear systems, Figure 88 and Figure 114, showing adequate tracking performance is achieved. Comparing the velocity tracking error of the higher-order nonlinear systems, Figure 89 and Figure 115, shows the moving boundary layer removed all chatter and reduced the velocity tracking error. Also, there is a negligible difference between the velocity comparison of the actual and desired values for the higher-order nonlinear systems, Figure 90 and Figure 116.

The moving boundary layer has a significant impact on the acceleration tracking error and acceleration comparison of the actual and desired values for the higher-order nonlinear systems. Comparing the acceleration tracking error of the higher-order nonlinear systems, Figure 91 and Figure 117, clearly shows the moving boundary layer removed all chatter and reduced the acceleration tracking error to a negligible amount. Once again the result is demonstrated in the acceleration comparisons of the higher-order nonlinear systems, Figure 92 and Figure 118. Figure 92 displays significant chatter in the system response, creating a large error between the desired and actual response whereas in Figure 118, the smoothing boundary layer removed the chatter, which forces the difference between the desired and actual response to be negligible.

Comparing the sliding condition of the higher-order nonlinear systems, Figure 93 and Figure 119, the benefits of the smoothing boundary layer are clear. Determining if the sliding condition was satisfied in Figure 93 is difficult due to the high amount of chatter, but once again is identically zero and thus the sliding condition is satisfied proving the stability nature of the closed-loop system. Whereas in Figure 119, the chatter was removed by using the smoothing boundary layer and the sliding condition is clearly satisfied. Examining Figure 94 and Figure 120, it is clear the smoothing boundary layer also removed the chatter on the control effort. Observing the phase planes for the system of the higher-order nonlinear systems, Figure 95 and Figure 121 the smoothing boundary layer had little effect on tracking performance. However, observing the phase portraits for the sliding system of the higher-order nonlinear systems, Figure 96 and Figure 122, the smoothing boundary layer removes all chatter creating a clearly defined
phase plane. The smoothing boundary layer removed all chatter from the system responses and enabled the robust control of the model-free sliding mode controller.

5.3.3 Comparison to Model-Based Sliding Mode Controller

The following sections compare the model-based controller and model-free controller regarding their ability to control nonlinear systems. The effectiveness of a moving boundary layer is also discussed for both controller methods.

5.3.3.1 Without Moving Boundary Layer

Comparing the position tracking error of the model-based controller, Figure 1, to the position tracking error of the model-free controller, Figure 87, the model-free controller was more accurate, but produced more chatter. However, comparing the position comparisons of the model-based and model-free controllers, Figure 2 and Figure 88 respectively, the error is negligible for both controllers.

Comparing the velocity tracking error of the model-based controller, Figure 3, to the velocity tracking error of the model-free controller, Figure 89, the model-free controller was more accurate, but produced more chatter. However, comparing the velocity comparisons of the model-based and model-free controllers, Figure 4 and Figure 90 respectively, the error is negligible for both controllers.

Comparing the acceleration tracking error of the model-based controller, Figure 5 to the acceleration tracking error of the model-free controller, Figure 91, the model-free controller was more accurate, but produced more chatter. However, comparing the accelerations comparisons of the model-based and model-free controllers, Figure 6 and Figure 92 respectively, both controllers experienced significant tracking error.

Comparing the sliding condition of the model-based controller, Figure 7 to the sliding condition of the model-free controller, Figure 93, the model-based controller satisfied the sliding condition at all times, whereas the model-free controller did not. However, since the small error in the model-free controller can be contributed to numeration error and can be considered negligible. Also, the control efforts of the model-based and model-free controllers, Figure 8 and Figure 94 respectively, experienced significant chatter.
Comparing the phase planes of the model-based and model-free controllers, Figure 9 and Figure 95 respectively, both controllers produced a smooth response. Whereas the phase planes for the sliding system of the model-based and model-free controllers, Figure 10 and Figure 96 respectively, both controllers experienced significant chatter.

In conclusion, both the model-based and model-free controllers produced accurate tracking for the position and velocity comparisons, with the model-free controller having more accurate tracking. However, both controllers experienced significant chatter on the acceleration comparisons and control efforts. Also, the controllers produced a smooth response for the phase planes, proving the Lyapunov stability of the closed-loop system.

5.3.3.2 With Moving Boundary Layer

Comparing the position tracking error of the model-based controller, Figure 11, to the position tracking error of the model-free controller, Figure 113, the model-free controller was significantly more accurate. However, comparing the position comparisons of the model-based and model-free controllers, Figure 12 and Figure 114 respectively, the error is negligible for both controllers.

Comparing the velocity tracking error of the model-based controller, Figure 13, to the velocity tracking error of the model-free controller, Figure 115, the model-free controller was significantly more accurate. However, comparing the velocity comparisons of the model-based and model-free controllers, Figure 14 and Figure 116 respectively, the error is negligible for both controllers.

Comparing the acceleration tracking error of the model-based controller, Figure 15 to the acceleration tracking error of the model-free controller, Figure 117, the model-free controller was significantly more accurate. However, comparing the accelerations comparisons of the model-based and model-free controllers, Figure 16 and Figure 118 respectively, the error is negligible for both controllers.

Comparing the sliding surface of the model-based controller, Figure 17 to the sliding surface of the model-free controller, Figure 119, both sliding surfaces remained within the boundary. The error in the model-free controller was significantly smaller in comparison to the model-based
controller. Also, the control efforts of the model-based and model-free controllers, Figure 18 and Figure 120 respectively, experienced smooth responses.

In conclusion, both the model-based and model-free controllers produced accurate tracking for the position, velocity, and acceleration comparisons, with the model-free controller having significantly more accurate tracking. Both controllers also produced smooth responses for the control effort. The moving boundary layer proved to be effective for both controllers, however, the model-free controller produced more accurate responses.
6 CONCLUSIONS

A new model-free sliding mode control scheme based solely on observable measurements was developed. Lyapunov’s Direct Method was used to prove asymptotically stability during the reaching phase for any a nonlinear or linear type system. The method was applied to a linear and nonlinear first-order system along with a linear and nonlinear second-order system. The results produced significant chattering, which caused higher tracking error than desired. A time varying smoothing boundary layer was then applied to the controller form to remove chattering from the results. The new controller was applied to a linear and nonlinear first-order system along with a linear and nonlinear second-order system. The new controller was able to successfully remove chatter from all system responses including the control effort. The results proved the method produces a robust control system with precise tracking, smooth controller effort, and satisfies the sliding condition thus proving closed-loop stability in the Lyapunov sense. The model-free sliding mode controller was shown to be an effective method for the control of both linear and nonlinear systems.

The model-free method is more beneficial over model-based methods since the model-free method does not require knowledge of the system to be controlled as opposed to model-based methods. The model-free controller also proved to be more effective in comparison to a model-based method in regards to controller nonlinear systems with and without a moving boundary layer.

6.1 Suggestions for Future Work

The next step in the research of a model-free sliding mode controller would be to include an input gain on \( u \) in the formulation of the model-free control law and test the effectiveness of the new controller on linear and nonlinear systems. Measurement noise should also be added to the controller to determine how measurement noise affects the tracking performance. The controller should also be tested at different sampling rates along with different inputs to determine if a different sampling rate or a different input would affect the stability of the system. Discrete effects could be included using a zero-order hold. A controller for a multiple-input-multiple-output system should also be derived and used to control each output. Parametric variations should also be considered for the desired tracking. Once all the circumstances for instability have
been determined, a boundary map should then be created indicating the cases which produce the instability.

From there the model-free sliding mode controller can be applied to flight control systems used for Unmanned Aircraft Systems (UASs). Once the simulation phase is completed for UAS applications, an experimental verification can be shown on an actual UAS flight vehicle to validate the model-free control method approach.

For example, the Federal Aviation Administration announced Griffiss International Airport will be authorized to test commercial UAS, one of six sites nationwide. The success of the application is credited to Griffiss’ close proximity to other UAS testing sites and military bases who utilize UAS. Rome Laboratory and a technology park are also located at Griffiss, which encouraged many technology related businesses to locate in the area. Now that testing of UAS is allowed at Griffiss, the businesses will be able to become involved with the development and implementation of UAS. Griffiss will now play a key role in the FAA’s goal of integrating UAS into the national airspace by 2015. [12] The FAA UAS test site can be used to test the performance of UAS using a model-free sliding control system.

Results from testing at the site will help to develop regulations to allow integration of UAS into the NAS. The testing will include determining safety of flying UASs with the National Airspace System (NAS) and analyzing the performance of the pilots control over the UAS with instrument only flight. Testing will also ensure the pilot has constant control over the UAS and cannot be overtaken by another source. [13]

6.2 Applications to Society

The control scheme proposed in this work can later be applied to Unmanned Aircraft System control systems, which will aid in their integration into the NAS. There are many potential applications for UAS’s if there were to be integrated into the NAS ranging from security and protection to civilian and industrial use. These applications include border security, search and rescue, farming, fire protection, plus many applications. UASs can be extremely beneficial to society if they were integrated into the NAS. Also, integrating UAS into the NAS will result in a large beneficial economic impact and employment impact according to the Association for Unmanned Vehicle System International [30]. The proposed model-free control system will aid
in the easy integration of UAS into the NAS because of its robust performance and simple implementation.
7 REFERENCES


8  APPENDIX

The following sections displays the MATLAB code used for each system. The Simulink Diagrams called for each system can be found above in their respective sections.

8.1  First-Order Linear System

% Simulation Parameters
tf=30;  % final simulation time
Ts=0.0001;  % sampling time
t=0:Ts:tf;t=t';  % time for input variables
signum_sw=1;  % 1 for sgn(s), 0 for sat(s/phi)

% SMC Parameters
lambda=20;  % slope of sliding surface
ita=0.1;  % small positive constant
phi=0.1;  % boundary layer thickness

sim('First_Order')

%define new variables
xtil=x-x_d;  % position tracking error
xtildot=xdot-xdot_d;  % velocity tracking error
sc=-ita*abs(s);  % sliding condition

% Plot
figure('Position',[10,550,550,410])
plot(t,xtil);title('Position Tracking Error')
xlabel('Time (s)');ylabel('x-x_d')

figure('Position',[580,550,550,410])
plot(t,x,t,x_d);title('Position Comparison')
xlabel('Time (s)');ylabel('x and x_d')

figure('Position',[1150,550,550,410])
plot(t,xtildot);title('Velocity Tracking Error')
xlabel('Time (s)');ylabel('xdot-xdot_d')

figure('Position',[10,60,550,410])
plot(t,xdot,t,xdot_d);title('Velocity Comparison')
xlabel('Time (s)');ylabel('xdot and xdot_d')

figure('Position',[10,550,550,410])
plot(t,u);title('Control Effort')
xlabel('Time (s)');ylabel('u')

figure('Position',[1150,550,550,410])
plot(t,s.*sdot,t,sc);title('Sliding Condition')
xlabel('Time (s)');ylabel('ssdot, sc');legend('s*sdot','ita*abs(s)')

figure('Position',[10,60,550,410])
plot(s,sdot);title('Phase Plane for the Sliding System')
8.2 Higher-Order Linear System

% Simulation Parameters
% final simulation time
% sampling time
% time for input variables
% mass
% damping coefficient
% spring constant
% 1 for sgn(s), 0 for sat(s/phi)

% SMC Parameters
% slope of sliding surface
% small positive constant
% boundary layer thickness

sim('MCK')

% define new variables
% position tracking error
% velocity tracking error
% acceleration tracking error
% sliding condition

% Plot
figure('Position', [10, 550, 550, 410])
plot(t, xtil); title('Position Tracking Error')
xlabel('Time (s)'); ylabel('x-x_d')

figure('Position', [580, 550, 550, 410])
plot(t, x, x_d); title('Position Comparison')
xlabel('Time (s)'); ylabel('x and x_d')

figure('Position', [1150, 550, 550, 410])
plot(t, xtildot); title('Velocity Tracking Error')
xlabel('Time (s)'); ylabel('xdot-xdot_d')

figure('Position', [10, 60, 550, 410])
plot(t, xdot, xdot_d); title('Velocity Comparison')
xlabel('Time (s)'); ylabel('xdot and xdot_d')

figure('Position', [580, 60, 550, 410])
plot(t, xtildotdot); title('Acceleration Tracking Error')
xlabel('Time (s)'); ylabel('xdotdot-xdotdot_d')

figure('Position', [1150, 60, 550, 410])
plot(t, xdotdot, xdotdot_d); title('Acceleration Comparison')
xlabel('Time (s)'); ylabel('xdotdot and xdotdot_d')
figure('Position', [10, 550, 550, 410])
plot(t, u); title('Control Effort')
xlabel('Time (s)'); ylabel('u')

figure('Position', [1150, 550, 550, 410])
plot(t, s.*sdot, t, sc); title('Sliding Condition')
xlabel('Time (s)'); ylabel('ssdot, sc'); legend('s*sdot', '-ita*abs(s)')

figure('Position', [10, 60, 550, 410])
plot(s, sdot); title('Phase Plane for the Sliding System')
xlabel('s'); ylabel('sdot')

figure('Position', [580, 60, 550, 410])
plot(x, xdot); title('Phase Plane for the System')
xlabel('x'); ylabel('xdot')

8.3 First-Order Linear System with Moving Boundary Layer

% Simulation Parameters
% final simulation time
% sampling time
% time for input variables

% SMC Parameters
% slope of the sliding surface
% small positive constant

sim('First_Order_moving_boundary')

% define new variables
% position tracking error
% velocity tracking error

% Plot
figure('Position', [10, 550, 550, 410])
plot(t, xtil); title('Position Tracking Error')
xlabel('Time (s)'); ylabel('x - x_d')

figure('Position', [580, 60, 550, 410])
plot(t, x, x_d); title('Position Comparison')
xlabel('Time (s)'); ylabel('x and x_d')

figure('Position', [1150, 550, 550, 410])
plot(t, xtildot); title('Velocity Tracking Error')
xlabel('Time (s)'); ylabel('xdot - xdot_d')

figure('Position', [10, 60, 550, 410])
plot(t, xdot, t, xdot_d); title('Velocity Comparison')
xlabel('Time (s)'); ylabel('xdot and xdot_d')

figure('Position', [10, 550, 550, 410])
plot(t, u); title('Control Effort')
xlabel('Time (s)'); ylabel('u')
8.4 Higher-Order Linear System with Moving Boundary Layer

% Simulation Parameters
tf=30; % final simulation time
Ts=0.0001; % sampling time
t=0:Ts:tf; % time for input variables
t=t'; % time
m=2; % mass
c=0.8; % damping coefficient
k=2; % spring constant

% SMC Parameters
lambda=20; % slope of sliding surface
ita=0.1; % small positive constant

sim('MCK_moving_boundary')

% define new variables
xtil=x-x_d; % position tracking error
xtildot=xdot-xdot_d; % velocity tracking error
xtildotdot=xdotdot-xdotdot_d; % acceleration tracking error

% Plot
figure('Position',[10,550,550,410])
plot(t,xtil);title('Position Tracking Error')
xlabel('Time (s)');ylabel('x-x_d')

figure('Position',[580,550,550,410])
plot(t,x,t,x_d);title('Position Comparison')
xlabel('Time (s)');ylabel('x and x_d')

figure('Position',[1150,550,550,410])
plot(t,xtildot);title('Velocity Tracking Error')
xlabel('Time (s)');ylabel('xdot-xdot_d')

figure('Position',[10,60,550,410])
plot(t,xdot,t,xdot_d);title('Velocity Comparison')
xlabel('Time (s)');ylabel('xdot and xdot_d')

figure('Position',[580,60,550,410])
plot(t,xtildotdot);title('Acceleration Tracking Error')
xlabel('Time (s)');ylabel('xdotdot-xdotdot_d')
8.5 First-Order Nonlinear System

% Simulation Parameters
tf=30; % final simulation time
Ts=0.0001; % sampling time
t=0:Ts:tf;t=t'; % time for input variables
signum_sw=1; % 1 for sgn(s), 0 for sat(s/phi)

% SMC Parameters
lambda=20; % slope of sliding surface
ita=0.1; % small positive constant
phi=0.1; % boundary layer thickness

sim('First_Order_Nonlinear')

define new variables
xt=x-x_d; % position tracking error
xtildot=xdot-xdot_d; % velocity tracking error
sc=-ita*abs(s); % sliding condition

% Plot
figure('Position',[10,550,550,410])
plot(t,xt);title('Position Tracking Error')
xlabel('Time (s)');ylabel('x-x_d')

figure('Position',[580,550,550,410])
plot(t,x,t,x_d);title('Position Comparison')
xlabel('Time (s)');ylabel('x and x_d')

figure('Position',[1150,550,550,410])
plot(t,xtildot);title('Velocity Tracking Error')
xlabel('Time (s)');ylabel('xdot-xdot_d')
8.6 Higher-Order Nonlinear System

% Simulation Parameters
% final simulation time
% sampling time
% time for input variables
% mass
% damping coefficient
% spring constant
% 1 for sgn(s), 0 for sat(s/phi)

% SMC Parameters
% slope of sliding surface
% small positive constant
% boundary layer thickness

sim('MCK_Nonlinear')

%define new variables
% position tracking error
% velocity tracking error
% acceleration tracking error
% sliding condition

% Plot
figure('Position', [10, 60, 550, 410])
plot(t, xtil); title('Position Tracking Error')
xlabel('Time (s)'); ylabel('x-x_d')

figure('Position', [580, 60, 550, 410])
plot(t, x, t, x_d); title('Position Comparison')
xlabel('Time (s)'); ylabel('x and x_d')

figure('Position', [1150, 550, 550, 410])
8.7 First-Order Nonlinear System with Moving Boundary Layer

% Simulation Parameters
tf=30; \textit{\%} final simulation time
Ts=0.0001; \textit{\%} sampling time
t=0:Ts:tf;t=t'; \textit{\%} time for input variables

% SMC Parameters
lambda=20; \textit{\%} slope of sliding surface
ita=0.1; \textit{\%} small positive constant

sim('First_Order_Nonlinear_moving_boundary')

%define new variables
xtil=x-x_d; \textit{\%} position tracking error
xtildot=xdot-xdot_d; \textit{\%} velocity tracking error

% Plot
figure('Position', [10,550,550,410])
plot(t,xtil);title('Position Tracking Error')
xlabel('Time (s)');ylabel('x-x_d')

figure('Position', [580,550,550,410])
plot(t,xtildot);title('Velocity Tracking Error')
xlabel('Time (s)');ylabel('xdot-xdot_d')

figure('Position', [1150,550,550,410])
plot(t,xtildotdot);title('Acceleration Tracking Error')
xlabel('Time (s)');ylabel('xdotdot-xdotdot_d')

figure('Position', [10,550,550,410])
plot(t,u);title('Control Effort')
xlabel('Time (s)');ylabel('u')

figure('Position', [1150,550,550,410])
plot(t,s.*sdot,t,sc);title('Sliding Condition')
xlabel('Time (s)');ylabel('ssdot, sc');legend('s*sdot','-ita*abs(s)')

figure('Position', [10,550,550,410])
plot(s,sdot);title('Phase Plane for the Sliding System')
xlabel('s');ylabel('sdot')

figure('Position', [580,550,550,410])
plot(x,xdot);title('Phase Plane for the System')
xlabel('x');ylabel('xdot')
8.8 Higher Order Nonlinear System with Moving Boundary Layer

% Simulation Parameters
% final simulation time
% sampling time
% time for input variables
% mass
% damping coefficient
% spring constant

% SMC Parameters
% slope of sliding surface
% small positive constant

sim('MCK_Nonlinear_moving_boundary')

% define new variables
% position tracking error
% velocity tracking error
% acceleration tracking error

% Plot
figure('Position',[10,550,550,410])
plot(t,xtildot);title('Position Tracking Error')
xlabel('Time (s)');ylabel('x-x_d')

figure('Position',[580,550,550,410])
plot(x,xdot);title('Position for the System')
xlabel('x');ylabel('xdot')
plot(t,x,t,x_d);title('Position Comparison')
xlabel('Time (s)');ylabel('x and x_d')

figure('Position', [1150,550,550,410])
plot(t,xtildot);title('Velocity Tracking Error')
xlabel('Time (s)');ylabel('xdot-xdot_d')

figure('Position', [10,60,550,410])
plot(t,xdot,t,xdot_d);title('Velocity Comparison')
xlabel('Time (s)');ylabel('xdot and xdot_d')

figure('Position', [580,60,550,410])
plot(t,xtildotdot);title('Acceleration Tracking Error')
xlabel('Time (s)');ylabel('xdotdot-xdotdot_d')

figure('Position', [1150,60,550,410])
plot(t,xdotdot,t,xdotdot_d);title('Acceleration Comparison')
xlabel('Time (s)');ylabel('xdotdot and xdotdot_d')

figure('Position', [10,550,550,410])
plot(t,u);title('Control Effort')
xlabel('Time (s)');ylabel('u')

figure('Position', [10,60,550,410])
plot(s,sdot);title('Phase Plane for the Sliding System')
xlabel('s');ylabel('sdot')

figure('Position', [580,60,550,410])
plot(x,xdot);title('Phase Plane for the System')
xlabel('x');ylabel('xdot')

figure('Position', [1150,60,550,410])
plot(t,abs(phi),t,s,t,-1*abs(phi));title('Boundary Layer')
xlabel('Time (s)');ylabel('Response');legend('phi','s','-phi')