Adaptive boundary layer sliding mode control for multi-input-multi-output systems

Tiffany Lodge

Follow this and additional works at: https://louis.uah.edu/uah-theses

Recommended Citation
https://louis.uah.edu/uah-theses/465

This Thesis is brought to you for free and open access by the UAH Electronic Theses and Dissertations at LOUIS. It has been accepted for inclusion in Theses by an authorized administrator of LOUIS.
ADAPTIVE BOUNDARY LAYER SLIDING MODE CONTROL FOR MULTI-INPUT-MULTI-OUTPUT SYSTEMS

Tiffany Lodge

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in The Department of Mechanical and Aerospace Engineering to The Graduate School of The University of Alabama in Huntsville

May 2023

Approved by:
Dr. Farbod Fahimi, Research Advisor/Committee Chair
Dr. Naga Venkat Adurthi, Committee Member
Dr. Chang-kwon Kang, Committee Member
Dr. Keith Hollingsworth, Department Chair
Dr. Shankar Mahalingam, College Dean
Dr. Jon Hakkila, Graduate Dean
Abstract

ADAPTIVE BOUNDARY LAYER SLIDING MODE CONTROL FOR MULTI-INPUT-MULTI-OUTPUT SYSTEMS

Tiffany Lodge

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

The Department of Mechanical and Aerospace Engineering
The University of Alabama in Huntsville
May 2023

When a nonlinear system is controlled using the conventional sliding mode control, the system input will create a high-frequency chatter, which can damage the system’s actuators. The most common method used to prevent chattering is the introduction of a constant boundary layer, but if the disturbance is high, then chattering can still happen in the system input. In this thesis, a sliding mode controller with an adaptive boundary layer is proposed for multi-input-multi-output systems that prevent chattering, even if the disturbance is high. To demonstrate the effectiveness of the proposed controller in suppressing chatter, a robotic arm was modeled in Matlab’s Simulink. Results show that when a disturbance is added, the controller with the constant boundary layer is chattering, whereas the control input for the adaptive boundary layer is still smooth.
Acknowledgements

Writing a thesis has challenged my knowledge in the field of controls, but I am thankful I had an exemplary professor and advisor, Dr. Farbod Fahimi, who adequately prepared me for the challenges. He not only was there to guide me through the process of writing this thesis, but also increased my knowledge in sliding mode control and robotics. He was a key factor in helping me to complete this research.

I’d like to also thank my committee members, Chang-kwon Kang and Naga Venkat Adurthi, because they are the first to give helpful criticism before this thesis is published. Also, for giving me the opportunity to continue advancing in my degree.

Lastly, I would like to thank my boyfriend Aidan Miller for encouraging me to keep pushing through my master’s when the courses became more vigorous; and thank you to my parents, Cathy Lodge and Brian Lodge, for their service in the armed forces which made it possible for me to pursue a graduate degree.
Table of Contents

Abstract ............................................................. ii

Acknowledgements .................................................. iv

Table of Contents ................................................... vii

List of Figures ....................................................... viii

List of Tables ......................................................... xi

Chapter 1. Introduction ............................................. 1

Chapter 2. Single-Input Single Output (SISO) System ........... 6

2.1 Constant Boundary Layer ................................. 6

2.1.1 Problem Definition ...................................... 6

2.1.2 Derivation of the Control Law ......................... 8

2.1.3 Stability Analysis ......................................... 10

2.2 Time-Varying Boundary Layer ............................. 14

2.2.1 Outside Boundary Layer (|s| ≥ ϕ) ...................... 15

2.2.2 Inside Boundary Layer (|s| < ϕ) ......................... 23
# Chapter 3. Multi-Input-Multi-Output (MIMO) System

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Constant Boundary Layer</td>
<td>27</td>
</tr>
<tr>
<td>3.1.1 Problem Definition</td>
<td>27</td>
</tr>
<tr>
<td>3.1.2 Derivation of the Control Law</td>
<td>28</td>
</tr>
<tr>
<td>3.1.3 Stability Analysis</td>
<td>30</td>
</tr>
<tr>
<td>3.2 Varying Boundary Layer</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 Outside Boundary Layer $(</td>
<td>s_i</td>
</tr>
<tr>
<td>3.2.2 Inside the Boundary Layer $(</td>
<td>s_i</td>
</tr>
</tbody>
</table>

# Chapter 4. Simulation and Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Definition of Values</td>
<td>43</td>
</tr>
<tr>
<td>4.2 Assumptions Made</td>
<td>45</td>
</tr>
<tr>
<td>4.3 Desired Motion Functions</td>
<td>45</td>
</tr>
<tr>
<td>4.4 Controller Gains</td>
<td>47</td>
</tr>
<tr>
<td>4.5 Nominal Values of Dynamic Model Parameters</td>
<td>48</td>
</tr>
<tr>
<td>4.6 Results with No Disturbance (Case 1)</td>
<td>49</td>
</tr>
<tr>
<td>4.6.1 Case 1a</td>
<td>49</td>
</tr>
<tr>
<td>4.6.2 Case 1b</td>
<td>53</td>
</tr>
<tr>
<td>4.6.3 Case 1c</td>
<td>57</td>
</tr>
<tr>
<td>4.7 Results with High Disturbance (Case 2)</td>
<td>62</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Theoretical Sliding Mode Control ........................................... 2
1.2 Practical implementation of Sliding Mode Control (no boundary layer) .................. 2
1.3 Practical implementation of SCM with fixed boundary layer with negligible disturbance ..................................................... 3
1.4 Practical implementation of SMC with fixed boundary layer with high disturbance .......................................................... 4
2.1 If \(|s| > \phi\), \(s\) approaches the boundary layer ......................... 17
2.2 Adaptive boundary layer ......................................................... 25
2.3 Adaptive boundary layer with high disturbance .......................... 26
4.1 RR-Robot ............................................................................. 44
4.2 Case 1a: \(s_1\)-trajectory vs Time ............................................. 50
4.3 Case 1a: \(s_2\)-trajectory vs Time ............................................. 50
4.4 Case 1a: \(\tau_1\) vs Time ............................................................. 51
4.5 Case 1a: \(\tau_2\) vs Time ............................................................. 51
4.6 Case 1a: \(e = \theta_1 - \theta_1^d\) vs Time ......................................... 52
4.7 Case 1a: \(e = \theta_2 - \theta_2^d\) vs Time ......................................... 52
4.8 Case 1a: \(\theta_1\) vs Time ............................................................. 53
4.9 Case 1a: \(\theta_2\) vs Time ............................................................. 53
4.10 Case 1b: \(s_1\)-trajectory vs Time ............................................ 54
4.11 Case 1b: \(s_2\)-trajectory vs Time ............................................ 54
4.12 Case 1b: \(\tau_1\) vs Time ............................................................. 55
4.13 Case 1b: \(\tau_2\) vs Time ............................................................. 55
4.14 Case 1b: $e = \theta_1 - \theta_1^d$ vs Time ........................................... 56
4.15 Case 1b: $e = \theta_2 - \theta_2^d$ vs Time ........................................... 56
4.16 Case 1b: $\theta_1$ vs Time .................................................. 57
4.17 Case 1b: $\theta_2$ vs Time .................................................. 57
4.18 Case 1c: $s_1$-trajectory vs Time .......................................... 58
4.19 Case 1c: $s_2$-trajectory vs Time .......................................... 58
4.20 Case 1c: $\tau_1$ vs Time .................................................... 59
4.21 Case 1c: $\tau_2$ vs Time .................................................... 59
4.22 Case 1c: $e = \theta_1 - \theta_1^d$ vs Time ......................................... 60
4.23 Case 1c: $e = \theta_2 - \theta_2^d$ vs Time ......................................... 60
4.24 Case 1c: $\theta_1$ vs Time .................................................... 61
4.25 Case 1c: $\theta_2$ vs Time .................................................... 61
4.26 Case 2a: $s_1$-trajectory vs Time ........................................... 63
4.27 Case 2a: $s_2$-trajectory vs Time ........................................... 63
4.28 Case 2a: $\tau_1$ vs Time .................................................... 64
4.29 Case 2a: $\tau_2$ vs Time .................................................... 64
4.30 Case 2a: $e = \theta_1 - \theta_1^d$ vs Time ......................................... 65
4.31 Case 2a: $e = \theta_2 - \theta_2^d$ vs Time ......................................... 65
4.32 Case 2a: $\theta_1$ vs Time .................................................... 66
4.33 Case 2a: $\theta_2$ vs Time .................................................... 66
4.34 Case 2b: $s_1$-trajectory vs Time ........................................... 67
4.35 Case 2b: $s_2$-trajectory vs Time ........................................... 67
4.36 Case 2b: $\tau_1$ vs Time .................................................... 68
4.37 Case 2b: $\tau_2$ vs Time .................................................... 68
4.38 Case 2b: $e = \theta_1 - \theta_1^d$ vs Time ......................................... 69
4.39 Case 2b: $e = \theta_2 - \theta_2^d$ vs Time ......................................... 69
4.40 Case 2b: \( \theta_1 \) vs Time ........................................ 70
4.41 Case 2b: \( \theta_2 \) vs Time ........................................ 70
4.42 Case 2c: \( s_1 \)-trajectory vs Time ............................ 71
4.43 Case 2c: \( s_2 \)-trajectory vs Time ............................ 71
4.44 Case 2c: \( \tau_1 \) vs Time ....................................... 72
4.45 Case 2c: \( \tau_2 \) vs Time ....................................... 72
4.46 Case 2c: \( e = \theta_1 - \theta_1^d \) vs Time ......................... 73
4.47 Case 2c: \( e = \theta_2 - \theta_2^d \) vs Time ......................... 73
4.48 Case 2c: \( \theta_1 \) vs Time ....................................... 74
4.49 Case 2c: \( \theta_2 \) vs Time ....................................... 74
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Parameter values</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>Gain values</td>
<td>48</td>
</tr>
<tr>
<td>4.3</td>
<td>Nominal parameter values</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>Disturbance</td>
<td>62</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

Sliding mode control is a common method to design a robust controller for nonlinear systems. This controller is simple to use and is helpful if there are unknown dynamic parameters. This method of control implements a discontinuous control signal that forces the system onto a surface and then is confined to this surface [1, 2]. This surface will help drive the errors in the system to zero. Once on the sliding surface, the controller should not be effected by parametric uncertainties. Conventional sliding mode control uses a switching function to keep the system confined to the surface, so when the dynamics start to pass the sliding manifold, the switching function will push the state trajectory back to the surface. This high-frequency switching control signal causes the dynamics to oscillate around the surface and this is called chattering [3, 4].

Figure 1.1 shows how the control law should theoretically drive the surface variable $s$ to zero and keep it at zero for the given $K$. However, in practical implementation, $s$ does not exactly land on zero, and will overshoot the surface. Therefore, sgn$(s)$ will be either 1 or -1 (non zero), repeating the overshoot of the surface as the errors vanish, ending up indefinitely oscillating around the origin as can be seen in Figure 1.2.
Figure 1.1: Theoretical Sliding Mode Control

Figure 1.2: Practical implementation of Sliding Mode Control (no boundary layer)
The main control challenge with this method is to smooth out this chattering because it will cause damage to actuators [5, 6, 7]. There are many methods that have been used to reduce chattering which can be seen in [8, 9, 6]. A solution that is mostly used to prevent the high frequency in the input is to add a constant boundary layer [10, 11, 12]. This boundary layer will change the switching function into a continuous structure so that once the dynamics of the system reaches the surface, it will stay there and no longer oscillate around the surface, Figure 1.3. Although the application of a constant boundary layer provides a solution for cases of low disturbance, in the case of high disturbance, the constant boundary layer will no longer be able to suppress the chatter, Figure 1.4. Another solution is to add an adaptive boundary layer to the sliding mode control so that when high disturbance is present the chattering can be prevented [13, 14, 15].

**Figure 1.3:** Practical implementation of SCM with fixed boundary layer with negligible disturbance
When high disturbance is added to the fixed boundary layer, $s$ will be able to go outside the boundary layers, Figure 1.4, which will cause chattering.

**Figure 1.4:** Practical implementation of SMC with fixed boundary layer with high disturbance

Another method of reducing chattering is using higher-order sliding mode control. In the case of conventional sliding mode control the first time derivative of the sliding surface is taken to help ensure that the surface goes to zero. The higher-order sliding mode control method will take higher order time derivatives of the sliding surface to ensure the surface goes to zero. The main disadvantage of higher-order sliding mode control is that it “directly depends on the initial conditions of the system which may not be accurately known” [16]. When disturbance is added to a higher-order sliding mode controller, chattering becomes higher than when using a conventional sliding mode controller [17]. Another dis-
advantage of using higher-order sliding mode control is that the higher derivative of the states, e.g. the acceleration or jerk, are needed for feedback. Calculating these derivatives is a challenge due to the existence of sensor noise [18, 19].

In this thesis, a controller will be derived based on the conventional sliding mode control, followed by how to derive and implement a constant boundary layer and adaptive boundary layer [10]. Robotic manipulators have highly nonlinear dynamics. For that reason, they have been used by researchers as a great benchmark case for demonstrating a controller’s performance [20, 21, 22, 23]. Also, in this thesis, to adequately compare the different controllers a robotic arm is simulated in Matlab/Simulink with the conventional sliding mode control with no boundary layer, a constant boundary layer, and an adaptive boundary layer. The results show that when no disturbance is present, the controller with no boundary layer has chattering and causes a large amount of error in the system. The controllers with a constant boundary layer and adaptive boundary layer are equally sufficient in preventing chattering. When the disturbance is added to the controllers the constant boundary layer contains chattering, but the adaptive boundary layer is still able to suppress the chatter.
Chapter 2. Single-Input Single Output (SISO) System

This chapter is to gain a better understanding of how to form a control law for a system using conventional sliding mode control. Additionally, different methods to smooth the system input are exhibited. Single-input-Single-Output systems are simplistic in derivations, which makes them easier to understand and control.

Section 1.1 explains how to form the control law and find the discontinuity gain. The constant boundary layer is the simplest method to prevent chattering in the system input. The constant boundary layer is also easy to implement once the control law is found. Section 1.2 explains how to derive a time-varying boundary layer. The constant boundary layer is not able to prevent chattering in the case of disturbance, whereas the time-vary boundary layer is able to adapt to the changes induced by disturbance.

2.1 Constant Boundary Layer

2.1.1 Problem Definition

Consider a general second-order equation representing the actual system.

\[ \ddot{x} = f + bu \]  

(2.1)
where $u$ is the control input, $b$ is the control gain which is unknown but of known bounds and its best estimate is $\hat{b}$, $x$ is the (scalar) output of interest, and the dynamics $f$ are not exactly known but best estimated as $\hat{f}$. The bound of uncertainty in $f$ is defined as:

$$|\hat{f} - f| \leq F$$  \hspace{1cm} (2.2)

where $F$ is some known constant. The known bounds of the control gain $b$ are defined as follows.

$$0 < b_{min} \leq b \leq b_{max}$$  \hspace{1cm} (2.3)

Assume that the nominal value of $b$, is defined as: $\hat{b} = (b_{min}b_{max})^{1/2}$.

Inverting Eq. (2.3) yields:

$$\frac{1}{b_{max}} \geq \frac{1}{\hat{b}} \geq \frac{1}{b_{min}}.$$  

Since $\frac{1}{\hat{b}}$ is undefined this term will disappear. We have:

$$\frac{1}{b_{max}} \leq \frac{1}{\hat{b}} \leq \frac{1}{b_{min}}.$$  

Now each term is multiplied by $\hat{b}$.

$$\frac{\hat{b}}{b_{max}} \leq \frac{\hat{b}}{\hat{b}} \leq \frac{\hat{b}}{b_{min}}$$
Then the value of $\hat{b}$ is plugged into the equation above.

$$\frac{\sqrt{b_{\min}b_{\max}}}{b_{\max}} \leq \frac{\hat{b}}{b} \leq \frac{\sqrt{b_{\min}b_{\max}}}{b_{\min}}$$

The equation is simplified into the form:

$$\sqrt{\frac{b_{\min}}{b_{\max}}} \leq \frac{\hat{b}}{b} \leq \sqrt{\frac{b_{\max}}{b_{\min}}}.$$ 

Set $\beta = \sqrt{\frac{b_{\max}}{b_{\min}}}$.

$$\beta^{-1} \leq \frac{\hat{b}}{b} \leq \beta \quad \beta^{-1} \leq \frac{b}{\hat{b}} \leq \beta \quad (2.4)$$

The nominal system is defined based on $\hat{f}$ and $\hat{b}$.

$$\ddot{x} = \hat{f} + \hat{b}u \quad (2.5)$$

### 2.1.2 Derivation of the Control Law

In this section, a control law for $u$ must be derived to drive the system in Eq. (2.1) on a desired trajectory despite the bounded uncertainties defined in Eqs. (2.2) and (2.4). Note that the control law is derived based on the nominal model, i.e. by using Eq. (2.5).

In order to have the system track a desired trajectory defined by $x^d(t)$, we define a sliding surface:

$$s = \dot{e} + \lambda e. \quad (2.6)$$
The motivation for the above definition is as follows. If a control law can drive \( s \) to zero, then, for \( \lambda > 0 \), \( e \) exponentially approaches zero as time passes. In order to use the dynamics of the system for derivation of \( u \), \( \ddot{x} \) needs to appear in the surface given by Eq. (2.6). This can be done by taking the derivative of Eq. (2.6).

\[
\dot{s} = \dot{e} + \lambda \dot{e}
\]

Now, \( \dot{e} = \ddot{x} - \ddot{x}^d \) is plugged in to yield:

\[
\dot{s} = \ddot{x} - \ddot{x}^d + \lambda \dot{e}.
\]

\( \ddot{x} \) is substituted from Eq. (2.5) into the above equation to get:

\[
\dot{s} = \hat{f} + \hat{b}u - \ddot{x}^d + \lambda \dot{e}. \tag{2.7}
\]

The following equation is defined to make sure that \( s \) vanishes as time passes:

\[
\dot{s} = -K \text{sgn}(s) \tag{2.8}
\]

where \( K > 0 \). Substituting Eq. (2.8) in Eq. (2.7) and solving for \( u \) yields:

\[
u = \hat{b}^{-1}(\dot{u} - K \text{sgn}(s)). \tag{2.9}\]
Where the notation $\hat{u}$ is used for simplicity and is defined as:

$$\hat{u} = -\dot{f} + \ddot{x}^d - \lambda \dot{e}. \quad (2.10)$$

The term $\text{sgn}$ represents the sign function:

$$\text{sgn}(s) = \begin{cases} 
1 & \text{if } s > 0 \\
0 & \text{if } s = 0 \\
-1 & \text{if } s < 0 
\end{cases}.$$

The above control law will be applied to the actual system presented in Eq. (2.1).

### 2.1.3 Stability Analysis

Note that the control law is derived based on the nominal model in Eq. (2.5). This nominal model is slightly different than Eq. (2.1), which represents the actual system. Now the question arises whether the control law derived based on the nominal model in Eq. (2.5) can stabilize the actual system represented by Eq. (2.1). The answer is found by stability analysis, which is proposed as the following theorem.

**Theorem:** Consider an uncertain dynamic system defined by Eq. (2.1).

$$\ddot{x} = f + bu$$
Consider a nominal dynamic model of the same dynamic system represented by Eq. (2.5).

\[ \ddot{x} = \hat{f} + \hat{b}u \]

The uncertainty in the \( f \) term is bounded as defined by Eq. (2.2).

\[ |f - \hat{f}| < F \]

The uncertainty in \( b \) is bounded as defined by Eq. (2.4):

\[ \beta^{-1} \leq \frac{\hat{b}}{b} \leq \beta \quad \beta^{-1} \leq \frac{b'}{b} \leq \beta \]

where \( \beta = \sqrt{\frac{b_{\max}}{b_{\min}}} \). We claim that the control law proposed in Eq. (2.9)

\[ u = \hat{b}^{-1}(\hat{u} - K\text{sgn}(s)) \]

drives the surface variable \( s \) to zero despite the bounded uncertainties, provided that the controller gain \( K \) is calculated as follows:

\[ K = \beta(F + \eta) + (\beta - 1)|\hat{u}| \quad (2.11) \]

where \( \eta > 0 \).

**Proof:** Consider a positive semi-definite Lyapunov function defined as follows:

\[ v = \frac{1}{2}s^2. \]
We prove that the time rate of the Lyapunov function is negative, leading to the decay of the Lyapunov function to zero, at which point $s$ vanishes. The rate of the Lyapunov function is calculated as follows.

$$
\dot{v} = s \dot{s} = s[f + bu - \ddot{x}^d + \lambda \dot{e}]
$$

Notice that $\dot{s}$ in the above equation is found with the actual system which is:

$$
\dot{s} = f + bu - \ddot{x}^d + \lambda \dot{e}.
$$  \hspace{1cm} (2.12)

Plug in the value of $u$ from Eq. (2.9), then plug in for the value of $\hat{u}$.

$$
\dot{v} = s[f + b\hat{b}^{-1}(\hat{u} - K \text{sgn}(s)) - \ddot{x}^d + \lambda \dot{e}]
$$

$$
\dot{v} = s[f - b\hat{b}^{-1}\hat{f} + b\hat{b}^{-1}(\ddot{x}^d - \lambda \dot{e}) - b\hat{b}^{-1}K \text{sgn}(s) - \ddot{x}^d + \lambda \dot{e}]
$$

The equation can be simplified into the form:

$$
\dot{v} = s[f - b\hat{b}^{-1}\hat{f} - b\hat{b}^{-1}K \text{sgn}(s) + (1 - b\hat{b}^{-1})(-\ddot{x}^d + \lambda \dot{e})].
$$

The terms in the right-hand side of the above equation are maximized by using their absolute value so that the worst-case scenario for $\dot{v}$ is used.

$$
\dot{v} \leq |s||f - b\hat{b}^{-1}\hat{f} + (1 - b\hat{b}^{-1})(-\ddot{x}^d + \lambda \dot{e})| - b\hat{b}^{-1}K|s|
$$

12
Note that \( f = f - \hat{f} + \hat{f} \).

\[
\dot{v} \leq |s||f - \hat{f} + (1 - \hat{b}\hat{b}^{-1})\hat{\dot{f}} + (1 - \hat{b}\hat{b}^{-1})(-\ddot{x} + \lambda \dot{e})| - \hat{b}\hat{b}^{-1}K|s|
\]

\[
\dot{v} \leq |s||(f - \hat{f}) + (1 - \hat{b}\hat{b}^{-1})(\hat{\dot{f}} - \ddot{x} + \lambda \dot{e})| - \hat{b}\hat{b}^{-1}K|s|
\]

The maximum case for \( f - \hat{f} \) is \( F \) (see Eq. (2.2)). The maximum case for \( (1 - \hat{b}\hat{b}^{-1}) \) is \( (1 - \beta^{-1}) \) and the minimum case for \( \hat{b}\hat{b}^{-1}K \) is \( \beta^{-1}K \) (see Eq. (2.4)).

\( \hat{u} = \hat{f} - \ddot{x} + \lambda \dot{e} \) so we have:

\[
\dot{v} \leq |s||F + (1 - \beta^{-1})\hat{u} - \beta^{-1}K|s|.
\]

Substitute \( K \) from Eq. (2.11) in the above equation and cancel out like terms.

\[
\dot{v} \leq |s||F + (1 - \beta^{-1})\hat{u}| - |s|(F + \eta) - |s|(1 - \beta^{-1})|\hat{u}|
\]

\[
\dot{v} \leq |s|F + |s|(1 - \beta^{-1})|\hat{u}| - |s|F - |s|\eta - |s|(1 - \beta^{-1})|\hat{u}|
\]

\[
\dot{v} \leq -\eta|s|
\]

As the above equation implies, \( \dot{v} \) is negative for all \( s \), which means that \( v \) vanishes, at which point \( s \) vanishes too. This concludes the proof of the theorem. Now that we know what the discontinuity gain, \( K \), is let us add in a boundary layer, \( \phi \), and replace \( K \text{sgn}(s) \) with \( K \text{sat}(\frac{s}{\phi}) \) to help with the chattering.
2.2 Time-Varying Boundary Layer

Consider systems that are subjected to large disturbances in short periods of time. When controlling such systems using the sliding mode control, one must use a high gain value for $K$ by increasing $\eta$ (see Eq. (2.11)), such that the sliding mode controller can reject the disturbance. A high value of $K$ makes $\dot{s}$ large (see Eq. (2.8)). When $\dot{s}$ is large, there is a high possibility that $s$ (which is moving fast) overshoots the boundary layer, nullifying the purpose of the boundary layer, and causing the control input to chatter.

To prevent the chatter caused by the above scenario, we propose an adaptive $K$ (named $\bar{K} = K − G\dot{\phi}$, $G > 0$) along with a variable boundary layer that varies at the rate of $\dot{\phi}$. When a disturbance is received, the boundary layer expands ($\dot{\phi} > 0$), reducing $\bar{K}$; hence preventing chatter. When the disturbance is passed, the boundary layer shrinks ($\dot{\phi} < 0$) increasing $\bar{K}$, and consequently $\dot{s}$; hence accelerating the reduction of $s$.

Now, the question is, what mathematical law must we use for changing the boundary layer thickness $\phi$, and what the gain $G$ is to obtain a stable system response while avoiding chatter? In the following, the stability of the system, i.e. the response of $s$ is studied for two cases: when $s$ is outside the boundary layer ($|s| \geq \phi$); and when $s$ is inside the boundary layer ($|s| < \phi$).

It is desired for $s$ to approach the boundary layer when it is outside of the boundary layer ($|s| \geq \phi$), and to remain inside the boundary layer after entering it ($|s| < \phi$).
2.2.1 Outside Boundary Layer \((|s| \geq \phi)\)

**Remark:** For \(s\) to approach the boundary layer, the following condition must be true.

\[
\frac{1}{2} \frac{d}{dt} s^2 \leq (\dot{\phi} - \eta)|s| \tag{2.13}
\]

**Proof:** Consider the case \(\phi > 0\). The above equation is expanded to:

\[
s\dot{s} \leq (\dot{\phi} - \eta)|s|.
\]

Since \(s \geq \phi\), \(s\) is positive meaning that \(s = |s| > 0\) so an \(s\) can be divided out.

\[
\dot{s} \leq (\dot{\phi} - \eta)
\]

Now \(\dot{\phi}\) can be moved over to the left-hand side.

\[
\dot{s} - \dot{\phi} \leq -\eta
\]

\[
\frac{d}{dt}(s - \phi) \leq -\eta \tag{2.14}
\]

Note that \(s \geq \phi > 0\), i.e \(s - \phi\) is positive. Equation (2.14) implies that the rate of \(s - \phi\) is negative, monotonically decreasing \(s - \phi\); to eventually land \(s\) on \(\phi\) (the boundary layer).

Now consider the case of \(s \leq -\phi < 0\). Equation (2.13) is expanded to:

\[
s\dot{s} \leq (\dot{\phi} - \eta)|s|.
\]
Since $s < -\phi < 0$, $s$ is negative meaning that $-s = |s|.$

\[ s\dot{s} \leq -(\dot{\phi} - \eta)s \]

Now it can be clearly seen that an $s$ can be divided out. However, since $s < 0$, the sign of the inequality will flip.

\[ \dot{s} \geq -(\dot{\phi} - \eta) \]

Bring $\dot{\phi}$ to the left-hand side to get:

\[ \dot{s} - (-\dot{\phi}) \geq \eta \]

\[ \frac{d}{dt}(s - (-\phi)) \geq \eta. \quad \text{(2.15)} \]

Note that $s \leq -\phi < 0$, i.e. $s - (-\phi)$ is negative. Equation (2.15) implies that the rate of $s - (-\phi)$ is positive, monotonically increasing $s - (-\phi)$; to eventually land $s$ on $-\phi$ (the boundary layer). Figure 2.1 shows that Eq. (2.14) and Eq. (2.15) would cause $s$ to eventually land on the boundary layer.
This concludes the remark and it can be seen that $s$ is approaching the boundary layer from outside the boundary layer.

*Theorem:* If $K$ is revised to $\bar{K} = K - G\dot{\phi}$, and $\dot{\phi}$ and $G$ are determined by the following equations, then, $s$ approaches the boundary layer in finite time.

\[
\dot{\phi} = \begin{cases} 
\beta^d K(x^d) - \lambda \phi & \text{if } \beta^d K(x^d) \geq \lambda \phi \\
\frac{K(x^d)}{\beta^d} - \frac{\lambda \phi}{(\beta^d)^2} & \text{if } \beta^d K(x^d) \leq \lambda \phi
\end{cases}
\]  \hspace{1cm} (2.16)

\[
G = \begin{cases} 
\beta^{-1} & \text{if } \beta^d K(x^d) \geq \lambda \phi \\
\beta & \text{if } \beta^d K(x^d) \leq \lambda \phi
\end{cases}
\]  \hspace{1cm} (2.17)

Where $x^d$ is the desired trajectory and $\beta^d = \beta(x^d)$. 

\[\text{Figure 2.1: If } |s| > \phi, s \text{ approaches the boundary layer}\]
Proof: Note that, in order to apply the boundary layer, the law for reducing s is as follows.

\[ \dot{s} = -\bar{K}(x)\text{sat}(\frac{s}{\phi}) \]

where \( \text{sat} \) represents the saturation function which is defined as follows.

\[
\text{sat}(\frac{s}{\phi}) = \begin{cases} 
\text{sgn}(\frac{s}{\phi}) & \text{if } |s| \geq \phi \\
\frac{s}{\phi} & \text{if } |s| \leq \phi
\end{cases}
\]

Since \( \phi > 0 \) does not effect the sign of \( s \), we have \( \text{sgn}(\frac{s}{\phi}) = \text{sgn}(s) \). This is the same as Eq. (2.8) which gives the same control law as Eq. (2.9). Use Eq. (2.12) and plug in the value for \( u \) from Eq. (2.9) to get:

\[ \dot{s} = f + b\hat{b}^{-1}(\hat{u} - \bar{K}(x)\text{sgn}(s)) - \ddot{x}d + \lambda \dot{e}. \]

Plugging in for \( \hat{u} \) as defined in Eq. (2.10) will give the below equation.

\[ \dot{s} = f + b\hat{b}^{-1}(-\hat{f} + \ddot{x}d - \lambda \dot{e} - \bar{K}(x)\text{sgn}(s)) - \ddot{x}d + \lambda \dot{e} \]

Note that \( f = f - \hat{f} + \hat{f} \).

\[ \dot{s} = f - \hat{f} + \hat{f} + b\hat{b}^{-1}(-\hat{f} + \ddot{x}d - \lambda \dot{e}) - b\hat{b}^{-1}\bar{K}(x)\text{sgn}(s) - \ddot{x}d + \lambda \dot{e} \]
The equation can be simplified into the following form:

\[
\dot{s} = f - \hat{f} + (1 - \hat{b}\hat{b}^{-1})(\hat{f} - \dddot{x}^d + \lambda \hat{c}) - \hat{b}\hat{b}^{-1} \bar{K}(x) \text{sgn}(s). \tag{2.18}
\]

Equation (2.13) is repeated here for convenience.

\[
\dot{s} \leq (\dot{\phi} - \eta)|s| \tag{2.19}
\]

Note that Eq. (2.19) is in the form of an inequality. Here, Eq. (2.18) is multiplied by \(s\), then, its right hand side is maximized to form an equality that could be matched with Eq. (2.19).

\[
\dot{s} \leq |s||f - \hat{f}| + |1 - \hat{b}\hat{b}^{-1}||\hat{f} - \dddot{x}^d + \lambda \hat{c}||s| - \hat{b}\hat{b}^{-1} \bar{K}(x) \text{sgn}(s)s
\]

Recall: \(|f - \hat{f}| = F\).

\[
\dot{s} \leq |s|F + |1 - \hat{b}\hat{b}^{-1}||\hat{f} - \dddot{x}^d + \lambda \hat{c}||s| - \hat{b}\hat{b}^{-1} \bar{K}(x)|s|
\]

\[
\dot{s} \leq |s|\frac{\hat{b}\hat{b}^{-1}}{b\hat{b}^{-1}}F + |1 - \hat{b}\hat{b}^{-1}||\hat{f} - \dddot{x}^d + \lambda \hat{c}||s| - \hat{b}\hat{b}^{-1} \bar{K}(x)|s|
\]

Pull a \(\hat{b}\hat{b}^{-1}\) out of the first two terms and simplify equation:

\[
\dot{s} \leq \hat{b}\hat{b}^{-1}(|s|\frac{1}{\hat{b}\hat{b}^{-1}}F + |1|) - 1||\hat{f} - \dddot{x}^d + \lambda \hat{c}||s| - \hat{b}\hat{b}^{-1} \bar{K}(x)|s|
\]

\[
\dot{s} \leq \hat{b}\hat{b}^{-1}(|s|b\hat{b}^{-1}bF + |b^{-1}b - 1||\hat{f} - \dddot{x}^d + \lambda \hat{c}||s| - \hat{b}\hat{b}^{-1} \bar{K}(x)|s|
\]
\[ ss \leq \hat{b} b^{-1}(b^{-1} \hat{b} F + |b^{-1} \hat{b} - 1||\hat{f} - \hat{x}^d + \lambda \hat{\epsilon}|)|s| - \hat{b} b^{-1} \hat{K}(x)|s|. \]

In order to achieve convergence to the surface, Eq. (2.13), or equivalently Eq. (2.19) must be true. Therefore, we enforce the right hand of the above equality to be equal to \((\dot{\phi} - \eta)|s|\).

\[
\hat{b} b^{-1}(b^{-1} \hat{b} F + |b^{-1} \hat{b} - 1||\hat{f} - \hat{x}^d + \lambda \hat{\epsilon}|)|s| - \hat{b} b^{-1} \hat{K}(x)|s| = (\dot{\phi} - \eta)|s| \quad (2.20)
\]

Here, we use the definition for \(K\), presented in Eq. (2.11), to simplify the above equation. Equation (2.11) is rewritten here for reference. However, the worst case value of \(\beta = b^{-1} \hat{b}\) from Eq. (2.4) used.

\[
K(x) = b^{-1} \hat{b}(F + \eta) + |b^{-1} \hat{b} - 1||\hat{f} - \hat{x}^d + \lambda \hat{\epsilon}|
\]

This can be rearranged to the form:

\[
K(x) - b^{-1} \hat{b} \eta = b^{-1} \hat{b} F + |b^{-1} \hat{b} - 1||\hat{f} - \hat{x}^d + \lambda \hat{\epsilon}|.
\]

The right-hand side of the above equation appears in Eq. (2.20). We replace that with the left-hand side of the above equation.

\[
\hat{b} b^{-1}(K(x) - b^{-1} \hat{b} \eta)|s| - \hat{b} b^{-1} \hat{K}(x)|s| = (\dot{\phi} - \eta)|s|
\]
We solve the above equation for $\bar{K}(x)$ to find:

$$
\bar{K}(x) = K(x) - b^{-1}\dot{b}\dot{\phi}. \quad (2.21)
$$

*Case 1:* When $\dot{\phi} < 0$, the boundary layer is shrinking. In this case, we would like to increase the value of $\bar{K}$. Therefore, we select the maximum value of $b^{-1}\dot{b}$, which is equal to $\beta$ (see Eq. (2.4)).

$$
\bar{K}(x) = K(x) - \beta\dot{\phi}
$$

This means that $\bar{K}$, derived from Eq. (2.21) is determined using the following relation:

$$
\bar{K}(x) = K(x) - \beta\dot{\phi} \quad \text{if} \quad \dot{\phi} < 0. \quad (2.22)
$$

This implies that:

$$
G = \beta \quad \text{if} \quad \dot{\phi} < 0. \quad (2.23)
$$

*Case 2:* When $\dot{\phi} > 0$, the boundary layer is expanding. In this case, we would like to decrease the value of $\bar{K}$, but we tend to reduce the amount of decrease. Therefore, we select minimum value of $b^{-1}\dot{b}$, which is equal to $\frac{1}{\beta}$ (see Eq. (2.4)).

$$
\bar{K}(x) = K(x) - \frac{\dot{\phi}}{\beta}
$$
This means that $\bar{K}$, derived from Eq. (2.21) is determined using the following relation:

$$
\bar{K}(x) = K(x) - \frac{\dot{\phi}}{\beta} \quad \text{if} \quad \dot{\phi} > 0.
$$

(2.24)

This implies that:

$$
G = \frac{1}{\beta} \quad \text{if} \quad \dot{\phi} > 0.
$$

(2.25)

To find $\dot{\phi}$, the balance condition in Eq. (2.35) is used. From Eq. (2.35), solve for $\bar{K}(x^d)$ to yield:

$$
\bar{K}(x^d) = \frac{\lambda \phi}{\beta d}.
$$

(2.26)

To derive $\dot{\phi}$, we rewrite equations (2.22) and (2.24) for when $x = x^d$.

$$
\bar{K}(x^d) = K(x^d) - \beta^d \dot{\phi} \quad \text{if} \quad \dot{\phi} < 0
$$

(2.27)

$$
\bar{K}(x^d) = K(x^d) - \frac{\dot{\phi}}{\beta d} \quad \text{if} \quad \dot{\phi} > 0.
$$

(2.28)

Plug Eq. (2.26) into Eqs. (2.27) and (2.28) to find $\dot{\phi}$’s desired behaviour:

$$
\frac{\lambda \phi}{\beta d} = K(x^d) - \beta^d \dot{\phi} \quad \text{if} \quad \dot{\phi} < 0
$$

(2.29)

$$
\frac{\lambda \phi}{\beta d} = K(x^d) - \frac{\dot{\phi}}{\beta d} \quad \text{if} \quad \dot{\phi} > 0.
$$

(2.30)

Rearrange Eqs. (2.29) and (2.30) to have $\dot{\phi}$ on left-hand side.

$$
\dot{\phi} = \frac{K(x^d)}{\beta d} - \frac{\lambda \phi}{(\beta d)^2} \quad \text{if} \quad \dot{\phi} \leq 0 \quad \text{or} \quad \beta^d K(x^d) \leq \lambda \phi
$$

(2.31)
\[ \dot{\phi} = \beta^d K(x^d) - \lambda \phi \quad \text{if} \quad \dot{\phi} \geq 0 \quad \text{or} \quad \beta^d K(x^d) \geq \lambda \phi \quad (2.32) \]

Equations (2.23), (2.25), (2.31), and (2.32) conclude the proof of the theorem.

### 2.2.2 Inside Boundary Layer (\(|s| < \phi\))

In this section, it is proved that after \(s\) enters the boundary layer, its response is bounded, and the bound of the response can be arbitrarily tuned by increasing the controller gain \(\lambda\).

**Theorem:** If \(K\) is replaced by \(\bar{K} = K - G\dot{\phi}\), \(s\) will be bounded inside the boundary layer (\(|s| < \phi\)).

**Proof:** To reduce the chatter replace \(K \text{sgn}(s)\) in Eq. (2.8) with \(\bar{K} \text{sat}(\frac{s}{\phi})\).

\[ \dot{s} = -\bar{K} \text{sat}(\frac{s}{\phi}) \quad (2.33) \]

Plug in Eq. (2.33) for \(\dot{s}\) into Eq. (2.9) to get the following:

\[ u = \tilde{b}^{-1}(\tilde{u} - \bar{K}(x) \text{sat}(\frac{s}{\phi})). \]

Plug this new control law into Eq. (2.12) to yield:

\[ \dot{s} = f + b\tilde{b}^{-1}(\tilde{u} - \bar{K}(x) \frac{s}{\phi}) - \ddot{x}^d + \lambda \dot{e} \]
where \( \text{sat}(\frac{s}{\phi}) \) is replaced by \( \frac{s}{\phi} \) because \( s \) is in the boundary layer \((|s| < \phi)\).

Plugging in for \( \hat{u} \) as defined in Eq. (2.10) will give the below equation.

\[
\dot{s} = f + b\hat{b}^{-1}(-\hat{f} + \ddot{x}^d - \lambda \dot{\epsilon} - \bar{K}(x)\frac{s}{\phi}) - \ddot{x}^d + \lambda \dot{\epsilon}
\]

Note that \( f = f - \hat{f} + \hat{f} \).

\[
\dot{s} = f - \hat{f} + \hat{f} + b\hat{b}^{-1}(-\hat{f} + \ddot{x}^d - \lambda \dot{\epsilon}) - b\hat{b}^{-1}\bar{K}(x)\frac{s}{\phi} - \ddot{x}^d + \lambda \dot{\epsilon}
\]

The equation can be simplified into the following form:

\[
\dot{s} = f - \hat{f} + (1 - b\hat{b}^{-1})(\hat{f} - \ddot{x}^d + \lambda \dot{\epsilon}) - b\hat{b}^{-1}\bar{K}(x)\frac{s}{\phi}.
\]

Note that, inside the boundary layer \( x \) and \( x^d \) are close so \( \bar{K}(x) \approx \bar{K}(x^d) \), \( f - \hat{f} = \Delta f(x^d) \approx \Delta f(x) \), and \( \dot{\epsilon} \approx 0 \).

\[
\dot{s} = \Delta f(x^d) + (1 - b(x^d)\hat{b}(x^d)^{-1})(\hat{f}(x^d) - \ddot{x}^d) - b(x^d)\hat{b}^{-1}(x^d)\bar{K}(x^d)\frac{s}{\phi}.
\]

\[
\dot{s} = -b(x^d)\hat{b}^{-1}(x^d)\bar{K}(x^d)\frac{s}{\phi} + (H(x^d) + O(\epsilon)) \tag{2.34}
\]

Here \( H(x^d) + O(\epsilon) \) is the total effect of uncertainties in the system which is bounded. In the above equation, the coefficient of \( s \) is negative. The above equation represents a stable first-order filter with the input \( H(x^d) + O(\epsilon) \) and the output \( s \). Since the input of the stable filter is bounded, its output \( s \) is bounded.

The coefficient of \( s \) in Eq. (2.34) will determine the rate at which \( s \) will decay. To
balance the rate of decay of $s$ and that of $e$ we will set the coefficient of $s$ equal to $\lambda$.

$$\frac{\bar{K}(x^d)}{\phi}(\frac{b(x^d)}{\bar{b}(x^d)})_{max} = \lambda$$

Simplify this once more using Eq. (2.4) and the balance condition is:

$$\frac{\bar{K}(x^d)}{\phi} \beta^d = \lambda.$$  \hspace{1cm} (2.35)

Note that a higher $\lambda$, which is now the coefficient of $s$ in (2.34), reduces the bound of $s$. Figure 2.2 shows that that $s$ will stay inside the boundary layer once entering the boundary layer. Also when disturbance is added to the adaptive boundary layer controller, $s$ will still stay inside the boundary layer because the boundary layer will adapt to keep $s$ inside as shown in Figure 2.3. This concludes the proof of the theorem.

![Figure 2.2: Adaptive boundary layer](image)

Figure 2.2: Adaptive boundary layer
Figure 2.3: Adaptive boundary layer with high disturbance
Chapter 3. Multi-Input-Multi-Output (MIMO) System

This chapter is used to bring the system from the previous (SISO System) chapter to multiple dimensions. The majority of systems are multi-input-multi-output systems. Similar methods to prevent chattering will be used in this chapter.

Section 2.1 explains how to derive the control law for MIMO systems and how to include a constant boundary layer. Section 2.2 explains how to derive the time-varying boundary layer because as explained in Chapter 1, the constant boundary layer is not able to suppress chattering in the case of high disturbance.

3.1 Constant Boundary Layer

3.1.1 Problem Definition

Consider a general second-order equation representing the actual system.

\[ \ddot{x} = \vec{f} + b \vec{u} \]  

(3.1)

where \( \vec{u} \) is the control input with dimensions \( n \times 1 \), \( b \) is the control gain which is unknown but of known bounds and its best estimate is \( \hat{b} \) with dimensions \( n \times n \), \( \vec{x} \) is the output of interest with dimensions \( n \times 1 \), and the dynamics \( \vec{f} \) are not
exactly known but best estimated as $\hat{f}$ with dimensions $n \times 1$. The bound of uncertainty in $\tilde{f}$ is defined as:

$$|\hat{f} - \tilde{f}| \leq \tilde{F}. \quad (3.2)$$

Known bounds of the control gain $b$

$$b\hat{b}^{-1} - I = \Delta \quad (3.3)$$

where

$$|\Delta_{ij}| \leq D_{ij} \text{ for } i, j = 1, \ldots, n.$$

The nominal system is defined as:

$$\ddot{x} = \hat{f} + \hat{b}u. \quad (3.4)$$

A control law for $\tilde{u}$ needs to be derived to drive the system in Eq. (3.1) on a desired trajectory despite the bounded uncertainties defined in Eqs. (3.2) and (3.3).

### 3.1.2 Derivation of the Control Law

The nominal model in Eq. (3.4) will be used to derive the control law. To have the system track a desired trajectory defined by $\tilde{x}^d(t)$, we define a sliding manifold:

$$\tilde{s} = \tilde{e} + \lambda \tilde{e} \quad (3.5)$$
where $\lambda$ is a diagonal $n \times n$ matrix and $\vec{e}$ is an $n \times 1$ vector. For $\lambda_{ii} > 0$, where $\lambda_{ii}$ is the $i$-th diagonal element of the matrix $\lambda$. The error, $\vec{e}$, will exponentially approach zero as time passes if $\vec{s}$ is driven to zero. By taking the derivative of Eq. (3.5), the dynamics of the system for $\vec{u}$ can be used, because $\ddot{x}$ will appear in the surface.

$$\dot{\vec{s}} = \vec{e} + \lambda \vec{e}.$$ 

Now, $\vec{e} = \vec{x} - \vec{x}^d$ is plugged in to yield:

$$\dot{\vec{s}} = \vec{x} - \vec{x}^d + \lambda \vec{e}.$$ 

Substitute in $\ddot{x}$ from Eq. (3.4) into the above equation to give:

$$\dot{\vec{s}} = \ddot{x} + \hat{b} \vec{u} - \vec{x}^d + \lambda \vec{e}. \quad (3.6)$$

The following equation is defined to ensure that $\vec{s}$ vanishes as time passes.

$$\dot{\vec{s}} = -K \text{sgn}(\vec{s}) \quad (3.7)$$

where $K$ is a positive definite diagonal $n \times n$ matrix with positive diagonal elements. Substituting Eq. (3.7) into Eq. (3.6) and solving for $\vec{u}$ will yield:

$$\vec{u} = \hat{b}^{-1}(\ddot{\vec{s}} - K \text{sgn}(\vec{s})) \quad (3.8)$$
where \( \hat{u} \) is used for simplicity and is defined as:

\[
\hat{u} = -\hat{f} + \ddot{x}^d - \lambda \ddot{e}.
\] (3.9)

Equation (3.8) will be the control law applied to the actual system presented in Eq. (3.1).

### 3.1.3 Stability Analysis

Notice that the control law was derived based on the nominal model in Eq. (3.4). The nominal model is marginally different than the actual system in Eq. (3.1). We apply stability analysis, to see whether the control law derived based on the nominal model in Eq. (3.4) can stabilize the actual system depicted in Eq. (3.1).

**Theorem:** Consider an uncertain MIMO dynamic system defined by Eq. (3.1).

\[
\ddot{x} = \ddot{f} + b\ddot{u}
\]

Consider a nominal dynamic model of the same dynamic system represented by Eq. (3.4).

\[
\ddot{x} = \hat{\ddot{x}} + \hat{b}\ddot{u}
\]

The uncertainty in the \( \ddot{f} \) term is bounded as defined in Eq. (3.2).

\[
|\ddot{f} - \ddot{f}| \leq \bar{F}
\]
The uncertainty in $b$ is bounded as defined by Eq. (3.3).

$$\Delta = bb^{-1} - I$$

The bound of the elements of the matrix $\Delta$ is defined as a matrix $D$, whose elements are defined as follows:

$$|\Delta_{ij}| < D_{ij} \text{ for } i, j = 1, \ldots, n. \quad (3.10)$$

We claim that the control law proposed in Eq. (3.8)

$$\vec{u} = b^{-1}(\hat{\vec{u}} - K\text{sgn}(\vec{s}))$$

drives the surface variable $\vec{s}$ to zero despite the bounded uncertainties, provided that the controller gain $K$ is calculated as follows.

$$\vec{K} = (I - D)^{-1} \left[ \vec{F} + D|\hat{\vec{u}}| + \vec{\eta} \right] \quad (3.11)$$

where $\vec{\eta}$ is of the dimensions of $n \times 1$ with all positive elements. $\vec{K}$ is a $n \times 1$ vector containing the diagonal values of the $K$ matrix.

Proof: Consider a positive semi-definite Lyapunov function defined as follows:

$$v = \frac{1}{2} s^T s.$$
We prove that the time rate of the Lyapunov function is negative, leading to the decay of the Lyapunov function to zero, at which point $\dot{s}$ vanishes. The rate of the Lyapunov function is calculated as follows.

\[
\dot{v} = s^T \dot{s} = s^T \left[ f + bu - \ddot{x}^d + \lambda \dot{e} \right]
\]

Notice that the above equation uses $\dot{s}$ with the actual system from Eq. (3.1) rather than the nominal model from Eq. (3.4), where $\dot{s}$ for the actual system is defined below.

\[
\dot{s} = f + bu - \tilde{x}^d + \lambda \dot{e} \tag{3.12}
\]

Plug in the value of $\bar{u}$ from Eq. (3.8), then plug in for the value of $\hat{u}$ from Eq. (3.9).

\[
\dot{v} = s^T \left[ f + b\hat{b}^{-1}(\hat{u} - K\text{sgn}(\bar{s})) - \tilde{x}^d + \lambda \dot{e} \right]
\]

Plug in the value for $b\hat{b}^{-1}$ from Eq. (3.3).

\[
\dot{v} = s^T \left[ f + (I + \Delta)(-f + \tilde{x}^d - \lambda \dot{e} - K\text{sgn}(\bar{s})) - \tilde{x}^d + \lambda \dot{e} \right]
\]

Expand out the equation to see the terms that can be canceled out.

\[
\dot{v} = s^T \left[ f + \Delta(-f + \tilde{x}^d - \lambda \dot{e}) - f + \tilde{x}^d - \lambda \dot{e} - (I + \Delta)K\text{sgn}(\bar{s}) - \tilde{x}^d + \lambda \dot{e} \right]
\]

32
The equation can be simplified into the following:

\[
\dot{v} = \dot{s}^T [\hat{f} + \Delta (-\hat{f} + \ddot{x} - \lambda \hat{e}) - \hat{f} - (I + \Delta)K \text{sgn}(\hat{s})].
\]

The terms in the right-hand side of the above equation are maximized by using their absolute value so that the worst-case scenario for \(\dot{v}\) is used.

\[
\dot{v} \leq |\dot{s}^T| |f - \hat{f}| + |\dot{s}^T| |\Delta| - |\dot{f} + \ddot{x} - \lambda \hat{e}| - |\dot{s}^T|(I + \Delta)k
\]

The maximum case for \(|f - \hat{f}|\) is \(\hat{F}\) and \(\hat{u} = -\hat{f} - \ddot{x} + \lambda \hat{e}\). The maximum case for \(\Delta\) is \(D\). To increase \(K\) for the worst case scenario, its coefficient is reduced by using \((I - D)\) instead of \((I + \Delta)\).

\[
\dot{v} \leq |\dot{s}^T| \hat{F} + |\dot{s}^T| D |\hat{u}| - |\dot{s}^T|(I - D) \hat{K}
\]

Substitute \(\hat{K}\) from Eq. (3.11) into the above equation.

\[
\dot{v} \leq |\dot{s}^T| \hat{F} + |\dot{s}^T| D |\hat{u}| - |\dot{s}^T|(I - D) \left[(I - D)^{-1} \left[\hat{F} + D |\hat{u}| + \eta\right]\right]
\]

\[
\dot{v} \leq |\dot{s}^T| \hat{F} + |\dot{s}^T| D |\hat{u}| - |\dot{s}^T| \hat{F} - |\dot{s}^T| D |\hat{u}| - |\dot{s}^T| \eta
\]

\[
\dot{v} \leq |\dot{s}^T|(-\eta)
\]

As the above equation implies, \(\dot{v}\) is negative for all \(\hat{s}\), which means that \(v\) vanishes, at which point \(\hat{s}\) vanishes too. This concludes the proof of the theorem. Now that
we know what the discontinuity gain, \( K \) is, let us add in a boundary layer, \( \phi \). To reduce the chatter, replace \( K \text{sgn}(\vec{s}) \) with \( K \text{sat}(\phi^{-1}\vec{s}) \), where \( \phi \) is a \( n \times n \) positive-definite diagonal matrix.

3.2 Varying Boundary Layer

Remark: Consider the \( n \times 1 \) surface vector \( \vec{s} \), the \( n \times 1 \) boundary layer vector \( \vec{\phi} \) (\( \phi_i > 0 \) for \( i = 1, \ldots, n \)), and the gain vector \( \vec{\eta} \) (\( \eta_i > 0 \) for \( i = 1, \ldots, n \)). For all \( s_i \) (\( i = 1, \ldots, n \)) to approach their corresponding boundary layer \( \phi_i \), the following condition must be true.

\[
\vec{s}^T \dot{\vec{s}} \leq |\vec{s}^T| (\dot{\vec{\phi}} - \vec{\eta})
\]

Proof: To force \( s_i \) approach to \( \phi_i \), when \( s_i > \phi_i > 0 \). i.e. \( s_i - \phi_i > 0 \), we must enforce:

\[
\frac{d}{dt}(s_i - \phi_i) \leq -\eta_i
\]

where \( \eta_i > 0 \). We simplify the above equation to

\[
\dot{s}_i - \dot{\phi}_i \leq -\eta_i
\]

\[
\dot{s}_i \leq (\dot{\phi}_i - \eta_i).
\]

Multiply the above equation by \( s_i > 0 \)

\[
s_i \dot{s}_i \leq s_i (\dot{\phi}_i - \eta_i).
\]
Since $s_i > \phi_i > 0$, therefore $|s_i| = s_i$, we have

$$s_i \dot{s}_i \leq |s_i| (\dot{\phi}_i - \eta_i). \quad (3.13)$$

To force $s_i$ approach to $-\phi_i$, when $s_i < -\phi_i < 0$, i.e. $s_i - (-\phi_i) < 0$, we must enforce:

$$\frac{d}{dt}(s_i - (-\phi_i)) \geq \eta_i$$

where $\eta_i > 0$. We simplify the above equation

$$\dot{s}_i - (-\dot{\phi}_i) \geq \eta_i$$

$$\dot{s}_i \geq -(\dot{\phi}_i - \eta_i).$$

Multiply the above equation by $s_i < 0$

$$s_i \dot{s}_i \leq -s_i (\dot{\phi}_i - \eta_i).$$

Since $s_i < -\phi_i < 0$, therefore $-s_i = |s_i|

$$s_i \dot{s}_i \leq |s_i| (\dot{\phi}_i - \eta_i). \quad (3.14)$$

Note that Eqs. (3.13) and (3.14) are the same. Therefore for $|s_i| \geq \phi_i$ to approach the boundary layer, it is sufficient that

$$s_i \dot{s}_i \leq |s_i| (\dot{\phi}_i - \eta_i).$$

35
We can sum up both of the above equations for \( i = 1, \ldots, n \), covering all \( s_i \)'s.

\[
\sum_{i=1}^{n} s_i \dot{s}_i \leq \sum_{i=1}^{n} |s_i| (\dot{\phi}_i - \eta_i)
\]

The above equation is written in vector form.

\[
\vec{s}^T \dot{\vec{s}} \leq |\vec{s}^T| (\frac{\dot{\vec{s}}}{\vec{\phi} - \vec{\eta}})
\]

This concludes the proof of the remark.

3.2.1 Outside Boundary Layer (\(|s_i| > \phi_i \text{ for } i = 1, \ldots, n\))

**Theorem:** If \( K \) is revised to \( \bar{K} \) where \( \bar{K} \) is defined in the following equation then \( \bar{s} \) approaches the boundary layer in finite time.

\[
\bar{K}(\vec{x}) = K(\vec{x}) - (I - D)^{-1} \dot{\vec{\phi}}
\]  

(3.15)

Where \( \dot{\vec{\phi}} \) is a \( n \times 1 \) vector with positive elements.

**Proof:** The following law is used to reduce \( \bar{s} \) to a zero vector.

\[
\dot{\bar{s}} = \bar{K}(\vec{x})\text{sat}(\phi^{-1} \bar{s})
\]

Where the diagonal \( n \times n \) matrix \( \phi \) holds the elements of the vector \( \vec{\phi} \) in its diagonal values. Since \( \phi_i > 0 \) for \( i = 1, \ldots, n \), we have \( \text{sgn}(\phi^{-1} \bar{s}) = \text{sgn}(\bar{s}) \). Here \( \dot{s} \) of the actual system will be used from Eq. (3.12), then, the value of the control
law \( \bar{u} \) will be plugged in from Eq. (3.8).

\[
\dot{s} = \bar{f} + b b^{-1} (\hat{u} - K \text{sgn}(\hat{s})) - \ddot{x} + \lambda \dot{\hat{e}}
\]

Plug in for the value of \( \hat{u} \) from Eq. (3.9).

\[
\dot{s} = \bar{f} + b b^{-1} (-\hat{f} + \ddot{x} - \lambda \dot{\hat{e}} - K \text{sgn}(\hat{s})) - \ddot{x} + \lambda \dot{\hat{e}}
\]

Plug in the value for \( b b^{-1} \) from Eq. (3.3).

\[
\dot{s} = \bar{f} + (I + \Delta) (-\hat{f} + \ddot{x} - \lambda \dot{\hat{e}} - K \text{sgn}(\hat{s})) - \ddot{x} + \lambda \dot{\hat{e}}
\]

Expand out the equation to see the terms that can be canceled out.

\[
\dot{s} = \bar{f} + \Delta (-\hat{f} + \ddot{x} - \lambda \dot{\hat{e}}) - \hat{f} + \ddot{x} - \lambda \dot{\hat{e}} - (I + \Delta) K \text{sgn}(\hat{s}) - \ddot{x} + \lambda \dot{\hat{e}}
\]

The equation can be simplified into the following:

\[
\dot{s} = \bar{f} - \hat{f} + \Delta (-\hat{f} + \ddot{x} - \lambda \dot{\hat{e}}) - (I + \Delta) K \text{sgn}(\hat{s}).
\]

Plug \( \hat{s} \) into the positive semi-definite Lyapunov function defined as follows:

\[
v = \frac{1}{2} \ddot{s}^T \ddot{s}.
\]

(3.16)
We prove that the time rate of the Lyapunov function is negative, which means that the Lyapunov function will decay to zero, at which point \( \vec{s} \) vanishes. The rate of decay is calculated as follows:

\[
\dot{v} = \vec{s}^T \dot{\vec{s}} = \vec{s}^T \left[ \vec{f} - \hat{\vec{f}} + \Delta(-\dot{\vec{f}} + \vec{x}d - \lambda \dot{\vec{e}}) - (I + \Delta)\tilde{K}(\vec{x})\text{sgn}(\vec{s}) \right].
\]

The terms on the right-hand side of the above equation are maximized by using the absolute value so that the worst-case scenario for \( \dot{v} \) is used.

\[
\dot{v} \leq |\vec{s}^T|\vec{f} - \hat{\vec{f}}| + |\vec{s}^T|\Delta| - \hat{\vec{f}} + \vec{x}d - \lambda \dot{\vec{e}}| - |\vec{s}^T|(I + \Delta)\tilde{K}(\vec{x})
\]

The maximum case for \( |\vec{f} - \hat{\vec{f}}| \) is \( \vec{F} \) and simplify \( | - \hat{\vec{f}} + \vec{x}d - \lambda \dot{\vec{e}}| \) to \( \hat{\vec{u}} \). Also, to increase the value of \( \tilde{K} \), the coefficient of \( \tilde{K} \) is reduced. This is done by replacing the term \( (I + \Delta) \) by \( (I - D) \).

\[
\dot{v} \leq |\vec{s}^T|\vec{F} + |\vec{s}^T|D|\hat{\vec{u}}| - |\vec{s}^T|(I - D)\tilde{K}(\vec{x})
\]

Substitute \( \tilde{K} \) from Eq. (3.15) then plug in \( K \) from Eq. (3.11).

\[
\dot{v} \leq |\vec{s}^T|\vec{F} + |\vec{s}^T|D|\hat{\vec{u}}| - |\vec{s}^T|(I - D)[K(\vec{x}) - (I - D)^{-1}\dot{\vec{\phi}}]
\]

\[
\dot{v} \leq |\vec{s}^T|\vec{F} + |\vec{s}^T|D|\hat{\vec{u}}| - |\vec{s}^T|(I - D)(I - D)^{-1}\left[\vec{F} + D|\hat{\vec{u}}| + \vec{\eta}\right] - (I - D)^{-1}\dot{\vec{\phi}}
\]

\[
\dot{v} \leq |\vec{s}^T|\vec{F} + |\vec{s}^T|D|\hat{\vec{u}}| - |\vec{s}^T|\vec{F} - |\vec{s}^T|D|\hat{\vec{u}}| - |\vec{s}^T|\vec{\eta} + |\vec{s}^T|\dot{\vec{\phi}}
\]
\[
\dot{v} \leq -|\tilde{s}|^T \tilde{\eta} + |\tilde{s}|^T \hat{\phi}
\]
\[
\dot{v} \leq |\tilde{s}|^T |(\hat{\phi} - \tilde{\eta})|
\]
This shows that \( \tilde{s} \) monotonically approaches the boundary layer in finite time and concludes the proof of this theorem.

3.2.2 Inside the Boundary Layer \((|s_i| < \phi_i \text{ for } i = 1, \ldots, n)\)

The section proves that after \( \tilde{s} \) enters the boundary layer, its response is bounded, and the bound of the response can be arbitrarily tuned by increasing the controller gain \( \lambda \).

**Theorem:** If \( K(\tilde{x}) \) is replaced by \( \bar{K}(\tilde{x}) = K(\tilde{x}) - (I - \Delta)^{-1} \hat{\phi} \), then, \( \tilde{s} \) will be bounded inside the boundary layer.

**Proof:** To reduce the chatter, we replace \( K_{\text{sgn}}(\tilde{s}) \) with \( K_{\text{sat}}(\phi^{-1} \tilde{s}) \) when inside the boundary layer \( \text{sat}(\phi^{-1} \tilde{s}) = \phi^{-1} \tilde{s} \). So plug \( K(\phi^{-1} \tilde{s}) \) in Eq. (3.7) for \( K_{\text{sgn}}(\tilde{s}) \).

\[
\dot{\tilde{s}} = -\bar{K}(\phi^{-1} \tilde{s})
\]

The new control law inside the boundary layer will be:

\[
\hat{u} = \hat{b}^{-1}(\hat{u} - \bar{K}(\tilde{x})(\phi^{-1} \tilde{s})). \quad (3.17)
\]
Plug this control law in for $\vec{u}$ in Eq. (3.12).

$$\dot{s} = f + \hat{b}b^{-1}(\hat{u} - K(\phi^{-1}s)) - \ddot{x}d + \lambda \dot{e}$$

Plug in for the value of $\hat{u}$ from Eq. (3.9).

$$\dot{s} = f + \hat{b}b^{-1}(-\hat{f} + \ddot{x}d - \lambda \dot{e} - K(\phi^{-1}s)) - \ddot{x}d + \lambda \dot{e}$$

Plug in the value for $b\hat{b}^{-1}$ from Eq. (3.3).

$$\dot{s} = f + (I + \Delta)(-\hat{f} + \ddot{x}d - \lambda \dot{e} - K(\phi^{-1}s)) - \ddot{x}d + \lambda \dot{e}$$

Here, let’s decompose $\Delta$ into two parts, a diagonal part, and an off-diagonal part.

$$\Delta = \Delta_d + \Delta_{od}$$

Plug in this new $\Delta$ into the above equation.

$$\dot{s} = f + (I + \Delta_d)(-\hat{f} + \ddot{x}d - \lambda \dot{e} - K(\phi^{-1}s)) + \Delta_{od}(-\hat{f} + \ddot{x}d - \lambda \dot{e} - K(\phi^{-1}s)) - \ddot{x}d + \lambda \dot{e}$$

Foil terms help find terms that can cancel in the equation.

$$\dot{s} = f - (I + \Delta_d)K(\phi^{-1}s) + (-\hat{f} + \ddot{x}d - \lambda \dot{e}) + \Delta_d(-\hat{f} + \ddot{x}d - \lambda \dot{e}) + \Delta_{od}(-\hat{f} + \ddot{x}d - \lambda \dot{e}) - \Delta_{od}K(\phi^{-1}s) - \ddot{x}d + \lambda \dot{e}$$
The equation can be simplified and put into the following form.

\[
\dot{s} = -(I + \Delta_d) \bar{K}(\bar{x})(\phi^{-1}s) + (\bar{f} - \tilde{f}) + \Delta_d(-\tilde{f} + \bar{x}^d - \lambda \tilde{e}) \\
+ \Delta_{od}(-\tilde{f} + \bar{x}^d - \lambda \tilde{e}) - \Delta_{od} \bar{K}(\bar{x})(\phi^{-1}s)
\]

Note that, inside the boundary layer \(\bar{x}\) and \(\bar{x}^d\) are close so \(\bar{K}(\bar{x}) \approx \bar{K}(\bar{x}^d)\), \(\bar{f} - \tilde{f} = \Delta f(\bar{x}^d) \approx \Delta f(\bar{x})\), and \(\tilde{e} \approx 0\).

\[
\dot{s} = -(I + \Delta_d) \bar{K}(\bar{x}^d)(\phi^{-1}s) + \Delta f(\bar{x}^d) + \Delta_d(-\tilde{f} + \bar{x}^d) + \Delta_{od}(-\tilde{f} + \bar{x}^d) - \Delta_{od} \bar{K}(\bar{x}^d)(\phi^{-1}s)
\]

For simplicity, we denote \(\Delta_d(-\tilde{f} + \bar{x}^d) + \Delta_{od}(-\tilde{f} + \bar{x}^d) - \Delta_{od} \bar{K}(\bar{x}^d)(\phi^{-1}s)\) as \(H(\bar{x}^d)\) since these terms all represent uncertainties in the system.

\[
\dot{s} = -(I + \Delta_d) \bar{K}(\bar{x}^d)(\phi^{-1}s) + (\Delta f(\bar{x}^d) + H(\bar{x}^d))
\]

(3.18)

Here \(\Delta f(\bar{x}^d) + H(\bar{x}^d)\) is the total effect of uncertainties in the system which is bounded. In the above equation, the coefficient of \(s\) is negative. The above equation represents a stable first-order filter with the input \(\Delta f(\bar{x}^d) + H(\bar{x}^d)\) and the output is \(\tilde{s}\). Since the input of the stable filter is bounded, its output \(\tilde{s}\) is bounded. The rate of decay of \(\tilde{s}\) inside the boundary layer is equal to the coefficient of \(\tilde{s}\). We take this to be equal to \(\lambda\) to balance the rate of decay of \(\tilde{s}\) and that of \(\tilde{e}\). The balance condition will be:

\[
(I + \Delta_d) \bar{K}(\bar{x}^d) \phi^{-1} = \lambda.
\]
We force $\vec{s}$ to decrease faster by increasing $\bar{K}(\vec{x}^d)$. For that reason, use $D_{d_{ii}} \geq |\Delta_{ii}|$, and reduce the coefficient of $\bar{K}(\vec{x}^d)$ by replacing $(I + \Delta_d)$ with $(I - D_d)$.

$$(I - D_d)\bar{K}(\vec{x}^d)\phi^{-1} = \lambda$$

(3.19)

This concludes the proof of the theorem. Note that increasing $\lambda$ increases the coefficient of $\vec{s}$ in Eq. (3.18), which reduces the bound of $\vec{s}$. 

42
Chapter 4. Simulation and Results

This chapter is focused on simulating a robotic arm and comparing the performance of sliding mode controllers with no boundary layer, constant boundary layer, and adaptive boundary layer. Robotic arms are used in the majority of modern manufacturing processes including automotive assembly lines, the production of sensitive medical devices, and more recently building construction. The adaptive boundary controller can be used to increase the quality at which these products can be made and even where steady maneuvers are needed.

4.1 Definition of Values

Let us consider the revolute-revolute (RR) robot arm, Figure 4.1. \( L_1 = 1\text{m} \) is the length of the first arm and \( L_2 = 1\text{m} \) is the length of the second arm. \( \theta_1 \) and \( \theta_2 \) are the angular positions of arm one and arm two, respectively. The center of mass for arm one and arm two is located at \( c_1 \) and \( c_2 \), respectively. The value of each parameter can be found in Table 4.1.
**Figure 4.1:** RR-Robot

**Table 4.1:** Parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>1 m</td>
</tr>
<tr>
<td>$L_2$</td>
<td>1 m</td>
</tr>
<tr>
<td>$m_1$</td>
<td>10 kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>11 kg</td>
</tr>
<tr>
<td>$g$</td>
<td>$9.81 \frac{m}{s^2}$</td>
</tr>
<tr>
<td>$\theta_1^d$</td>
<td>$\frac{\pi}{2} \left( 1 - \cos \frac{2\pi t}{T} \right)$ rad</td>
</tr>
<tr>
<td>$\theta_2^d$</td>
<td>$\frac{\pi}{2} \left( 1 - \cos \frac{2\pi t}{T} \right)$ rad</td>
</tr>
</tbody>
</table>
4.2 Assumptions Made

There are two assumptions made for the model simulated in simulink. The first assumption is that the only parameter that is not completely known is the mass of the two robotic links. The second assumption made is that the mass moment of inertia is negligible.

4.3 Desired Motion Functions

The equations of motion of the robotic arm is defined by the following equation:

$$\ddot{\tau} = M\ddot{\theta} + \dot{V} + \ddot{G}$$  \hspace{1cm} (4.1)

where the mass matrix , $M$, is symmetric. The $\dot{V}$ contains the centrifugal and Coriolis terms, $\ddot{G}$ contains the gravity terms, and $\ddot{\tau}$ is the control torque. $\ddot{\theta}$ is defined as:

$$\ddot{\theta} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$  \hspace{1cm} (4.2)
The derivations to find the equations of motion can be found in chapter 6 of [21].

The mass matrix, $\vec{V}$, and $\vec{G}$ are defined as follows:

$$M = \begin{bmatrix}
L_2^2 m_2 + 2 L_1 L_2 m_2 (\cos \theta_2) + L_1^2 (m_1 + m_2) & L_2^2 m_2 + L_1 L_2 m_2 (\cos \theta_2) \\
L_2^2 m_2 + L_1 L_2 m_2 (\cos \theta_2) & L_2^2 m_2
\end{bmatrix} \quad (4.3)$$

$$\vec{V} = \begin{bmatrix}
-m_2 L_1 L_2 (\sin \theta_2) \ddot{\theta}_2 - 2 m_2 L_1 L_2 (\sin \theta_2) \dot{\theta}_1 \dot{\theta}_2 \\
m_2 L_1 L_2 (\sin \theta_2) \ddot{\theta}_1
\end{bmatrix} \quad (4.4)$$

$$\vec{G} = \begin{bmatrix}
m_2 L_2 g (\cos (\theta_1 + \theta_2)) + (m_1 + m_2) L_1 g (\cos \theta_1) \\
m_2 L_2 g (\cos (\theta_1 + \theta_2))
\end{bmatrix}. \quad (4.5)$$

Equation (4.1) can be rearranged to see the desired system equation by getting $\ddot{\theta}$ on the left hand-side.

$$M \ddot{\theta} = -\vec{V} - \vec{G} + \vec{r}$$

$$\ddot{\theta} = M^{-1}(-\vec{V} - \vec{G}) + M^{-1} \vec{r}. \quad (4.6)$$
Now it is easier to see this equation is in the form:

\[ \ddot{\theta} = \vec{f} + b\vec{u} \]  

which is similar to Eq. (3.1). Some relations can be made by comparing Eq. (4.6) to Eq. (4.7).

\[ \vec{f} = M^{-1}(\vec{V} - \vec{G}) \]  

\[ b = M^{-1} \]  

\[ \vec{u} = \vec{r} \]

4.4 Controller Gains

The controller gains are found by trail and error, but for the best results \( \lambda \) is tuned first to get the desired and actual values of \( \ddot{\theta} \) as close as possible before tuning \( \vec{r} \). The gains used in each controller are defined in Table 4.2.
Table 4.2: Gain values

<table>
<thead>
<tr>
<th></th>
<th>No boundary layer</th>
<th>Constant Boundary layer</th>
<th>Adaptive Boundary Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>1 0</td>
<td>1 0</td>
<td>20 0</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>0 1</td>
<td>0 20</td>
</tr>
<tr>
<td>$\vec{\eta}$</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$\vec{\phi}$</td>
<td>–</td>
<td>0.02</td>
<td>$\phi_1(t)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.02</td>
<td>$\phi_2(t)$</td>
</tr>
</tbody>
</table>

4.5 Nominal Values of Dynamic Model Parameters

Not all the parameters of the dynamic model are known, so these values are estimated and called nominal values. These nominal values are used in the calculation of $\hat{f}$ and $\hat{b}$. The mass of the robotic arm links are estimated and the values can be found in Table 4.3. When forming the nominal system all other parameters are the same values seen in Table 4.1.
Table 4.3: Nominal parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{m}_1$</td>
<td>10 kg</td>
</tr>
<tr>
<td>$\hat{m}_2$</td>
<td>10 kg</td>
</tr>
</tbody>
</table>

4.6 Results with No Disturbance (Case 1)

In this section the following three simulation cases are presented and discussed.

- Case 1a: Model with no disturbance and controller with no boundary layer.
- Case 1b: Model with no disturbance and controller with a fixed boundary layer.
- Case 1c: Model with no disturbance and controller with an adaptive boundary layer

4.6.1 Case 1a

Figure 4.2 and Figure 4.3 show the $s$-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. These figures show that $s_1$ and $s_2$ present chattering. The chattering in $s_2$ is larger than that in $s_1$ meaning that the second link will show more vibration as the robotic arm moves to the desired position. This makes sense because the first link will
have chattering and present some vibrations when moving, which will add more vibration to the second link.

Figure 4.4 and Figure 4.5 show the driving torques of the links. $\tau_1$ is the driving torque of the first link, and $\tau_2$ is the driving torque of the second link. Both link’s driving torques show significant chattering. This is the control law that would be fed to the joint actuators. This shows why chattering can cause
damage to actuators, because the actuators would need to generate a fast response where there are spikes.

Figure 4.6 and Figure 4.7 show the errors between the angular position and desired angular position. The wanted result is for the error to be negligible. The error figures show that the error in $\theta_1$ has the largest error at 8 seconds. The error in $\theta_2$ can be seen in Figure 4.7 has larger error than $\theta_1$ between 5 and 7
Figure 4.6: Case 1a: $e = \theta_1 - \theta_1^d$ vs Time

seconds. The model was simulated for 10 seconds and within this time the errors in $\theta_1$ and $\theta_2$ remain very close to zero.

Figure 4.7: Case 1a: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.8 and Figure 4.9 show the angular position and desired angular position. The results should show that the angular position is equal to the desired angular position. Both Figure 4.8 and Figure 4.9 show the real value of $\theta_1$ and $\theta_2$ are very close to their desired values.
4.6.2 Case 1b

Now, we consider Case 1b: Model with no disturbance and controller with a fixed boundary layer.

Figure 4.10 and Figure 4.11 show the $s$-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. A desired result of these figures is for $s_1$ and $s_2$ to be negligible. These figures show that
Figure 4.10: Case 1b: $s_1$-trajectory vs Time

Figure 4.11: Case 1b: $s_2$-trajectory vs Time

$s_1$ and $s_2$ are bounded by a fixed boundary layer, staying very close to zero, and present no chattering. This also shows that $\bar{s}$ started inside the boundary layers and is staying within them.

Figure 4.12 and Figure 4.13 show the driving torques of the joints. $\tau_1$ is the driving torque of the first link, and $\tau_2$ is the driving torque of the second link. These figures show that no chattering, so the joint actuators would not be damaged with this controller.
Figure 4.12: Case 1b: $\tau_1$ vs Time

Figure 4.13: Case 1b: $\tau_2$ vs Time

Figure 4.14 and Figure 4.15 show the errors between the angular position and desired angular position. The desired result is for the error to be negligible. Figure 4.15 shows that the error in $\theta_2$ is very small. Figure 4.14 shows the error in $\theta_1$ is even smaller than $\theta_2$’s error.
Figure 4.14: Case 1b: $e = \theta_1 - \theta_1^d$ vs Time

Figure 4.15: Case 1b: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.16 and Figure 4.17 show the angular position and desired angular position. The results should show that the angular position is equal to the desired angular position. Both Figure 4.16 and Figure 4.17 show the real value of $\theta_1$ and $\theta_2$ are very close to their desired values.
4.6.3 Case 1c

Now, we consider Case 1c: Model with no disturbance and controller with an adaptive boundary layer.

Figure 4.18 and Figure 4.19 show the s-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. These figures show that $s_1$ and $s_2$ are bounded by an adaptive boundary layer, staying
very close to zero, and present no chattering. This also shows that $\vec{s}$ started inside the boundary layers and when $s_1$ and $s_2$ fluctuate the boundary layer fluctuates with it to keep $s_1$ and $s_2$ inside the boundary layer.

Figure 4.20 and Figure 4.21 show the driving torques of the joints. $\tau_1$ is the driving torque of the first joint, and $\tau_2$ is the driving torque of the second joint. These figures show that no chattering, so the joint actuators would not be damaged with this controller.
Figure 4.20: Case 1c: $\tau_1$ vs Time

Figure 4.21: Case 1c: $\tau_2$ vs Time

Figure 4.22 and Figure 4.23 show the errors between the angular position and desired angular position. The desired result is for the error to be negligible. Figure 4.23 shows that the error in $\theta_2$ is very small. Figure 4.22 shows the error in $\theta_1$ is even smaller than $\theta_2$’s error.
Figure 4.22: Case 1c: $e = \theta_1 - \theta_1^d$ vs Time

Figure 4.23: Case 1c: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.24 and Figure 4.25 show the angular position and desired angular position. The results should show that the angular position is equal to the desired angular position. Both Figure 4.24 and Figure 4.25 show the real value of $\theta_1$ and $\theta_2$ are extremely close to their desired values.
All cases above where able to have the real values of the angular position match with the desired values. The figures in case 1a show some chattering. The controllers with a boundary layer have a lot smoother outputs. The controllers in cases 1b and 1c are equally able to stop the chattering, the main difference that can be seen between the fixed boundary layer and the adaptive boundary layer is in the error. By comparing Figure 4.14 to Figure 4.22, one can see that the error when using the adaptive boundary layer is more than half of the error.
when using the fixed boundary layer. This can also be seen when comparing Figure 4.15 to Figure 4.23. The results show that when no distribution is added to the simulation, both cases 1b and 1c (SMC with fixed boundary layer and SMC with variable boundary layer) are able to stop chattering, and case 1a (SMC with no boundary layer) has chattering.

4.7 Results with High Disturbance (Case 2)

In this section, we present the following three simulation cases in which disturbance is added. The disturbance added to each model is defined in Table 4.4.

<table>
<thead>
<tr>
<th>Case</th>
<th>Disturbance</th>
<th>Max disturbance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2a</td>
<td>No Boundary Layer</td>
<td>50 25</td>
</tr>
<tr>
<td>2b</td>
<td>Constant Boundary Layer</td>
<td>50 25</td>
</tr>
<tr>
<td>2c</td>
<td>Adaptive Boundary Layer</td>
<td>50 25</td>
</tr>
</tbody>
</table>

This disturbance is added to the dynamics $f$, so now (4.8) will be replaced with:

$$\vec{f} = M^{-1}(-\vec{V} - \vec{G} + \text{Disturbance}).$$

(4.11)

- Case 2a: Model with disturbance and controller with no boundary layer.
- Case 2b: Model with disturbance and controller with a fixed boundary layer.
- Case 2c: Model with disturbance and controller with an adaptive boundary layer
4.7.1 Case 2a

First, we present case 2a, where a disturbance is added to the model and the controller has no boundary layer.

![s1-trajectory vs Time](image1)

**Figure 4.26:** Case 2a: $s_1$-trajectory vs Time

![s2-trajectory vs Time](image2)

**Figure 4.27:** Case 2a: $s_2$-trajectory vs Time

Figure 4.26 and Figure 4.27 show the $s$-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. These figures show that $s_1$ and $s_2$ are staying very close to zero, and present chattering.
Notice that the chatter has increased compared to Figure 4.2 and Figure 4.3. This is due to the presence of the high disturbance.

Figure 4.28: Case 2a: $\tau_1$ vs Time

Figure 4.29: Case 2a: $\tau_2$ vs Time

Figure 4.28 and Figure 4.29 show the driving torques of the joints. $\tau_1$ is the driving torque of the first joint, and $\tau_2$ is the driving torque of the second joint. These figures show a lot of chattering, so the joint actuators would be damaged with this controller.
Figure 4.30: Case 2a: $e = \theta_1 - \theta_1^d$ vs Time

Figure 4.31: Case 2a: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.30 and Figure 4.31 show the errors between the angular position and desired angular position. Figure 4.22 shows that $\theta_1$ has the largest error around 6 seconds. Figure 4.23 shows that $\theta_2$ has the largest error also around 6 seconds. With disturbance added into the simulation, the joints of each link are reflecting errors at the same time.
Figure 4.32: Case 2a: $\theta_1$ vs Time

Figure 4.33: Case 2a: $\theta_2$ vs Time

Figure 4.32 and Figure 4.33 show the angular position and desired angular position. The results should show that the angular position is equal to the desired angular position. Both Figure 4.32 and Figure 4.33 show the real value of $\theta_1$ and $\theta_2$ are very close to their desired values.
4.7.2 Case 2b

Now, let’s consider case 2b, in which the disturbance is added to the model and the controller has a fixed boundary layer.

![Figure 4.34: Case 2b: $s_1$-trajectory vs Time](image)

![Figure 4.35: Case 2b: $s_2$-trajectory vs Time](image)

Figure 4.34 and Figure 4.35 show the $s$-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. These figures show that $s_1$ and $s_2$ are bounded by a fixed boundary layer, and staying
close to zero. Notice that $s_1$ and $s_2$ started inside the boundary layers but are going outside the boundary layers where the disturbance is the highest.

Figure 4.36: Case 2b: $\tau_1$ vs Time

Figure 4.37: Case 2b: $\tau_2$ vs Time

Figure 4.36 and Figure 4.37 show the driving torques of the joints. $\tau_1$ is the driving torque of the first joint, and $\tau_2$ is the driving torque of the second joint. These figures show some chattering. However, even this small amount
of chattering will damage the actuators in the long run, after so many hours of repetitive work by the robotic arm.

Figure 4.38: Case 2b: $e = \theta_1 - \theta_1^d$ vs Time

Figure 4.39: Case 2b: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.38 and Figure 4.39 show the errors between the angular position and desired angular position. The errors in $\theta_1$ and $\theta_2$ have increased a lot with disturbance added. This increase in error will cause the real values for the angular position to be slightly off from the desired values.
Figure 4.40: Case 2b: $\theta_1$ vs Time

Figure 4.41: Case 2b: $\theta_2$ vs Time

Figure 4.40 and Figure 4.41 show the angular position and desired angular position. The results should show that the angular position is equal to the desired angular position. Both Figure 4.40 and Figure 4.41 show the real value of $\theta_1$ and $\theta_2$ are still close to their desired values, but the real values are slightly off from the desired values due to the errors in $\theta_1$ and $\theta_2$. 
4.7.3 Case 2c

Now, we present case 2c, in which the disturbance is added to the model and the controller has an adaptive boundary layer.

Figure 4.42: Case 2c: $s_1$-trajectory vs Time

Figure 4.43: Case 2c: $s_2$-trajectory vs Time

Figure 4.42 and Figure 4.43 show the $s$-trajectory of the robotic arm. Here, $s_1$-trajectory is for the first link, and $s_2$-trajectory is for the second link. These figures show that $s_1$ and $s_2$ are bounded by an adaptive boundary layer, stay
close to zero, and present no chattering. Notice that $s_1$ and $s_2$ started inside the boundary layers and are staying inside the boundary layers where the disturbance is the highest.

Figure 4.44: Case 2c: $\tau_1$ vs Time

Figure 4.45: Case 2c: $\tau_2$ vs Time

Figure 4.44 and Figure 4.45 show the driving torques of the joints. $\tau_1$ is the driving torque of the first joint, and $\tau_2$ is the driving torque of the second
joint. These figures show no chattering, which is good for preventing damage to the actuators.

![Figure 4.46](image1)

**Figure 4.46:** Case 2c: $e = \theta_1 - \theta_1^d$ vs Time

![Figure 4.47](image2)

**Figure 4.47:** Case 2c: $e = \theta_2 - \theta_2^d$ vs Time

Figure 4.46 and Figure 4.47 show the errors between the angular position and desired angular position. The errors in $\theta_1$ and $\theta_2$ are still very small even with disturbance added.
Figure 4.48: Case 2c: $\theta_1$ vs Time

Figure 4.49: Case 2c: $\theta_2$ vs Time

Figure 4.48 and Figure 4.49 show the angular position and desired angular position. The results should show that the angular position is extremely close to the desired angular position. Both Figure 4.48 and Figure 4.49 show the real value of $\theta_1$ and $\theta_2$ are still extremely close to their desired values.

All three controllers with disturbance added are still able to get the real values to closely match the desired values. Notice that in cases 2a and 2b the control torque has chattering. The controller with the fixed boundary layer is able
to reduce the chatter. However, the repetitive use of the robot, even with small chatter in the joint inputs will cause an eventual joint damage. The controller with the adaptive boundary layer is the only controller to completely suppress the chattering. Since the boundary layer thickness $\phi$ is able to increase when $s_1$ and $s_2$ fluctuate the $s$-trajectory will not be able to go outside the adaptive boundary layer. Interestingly, the error is the adaptive boundary layer Figure 4.46 and Figure 4.47 is not less than the error in the controller with no boundary layer, Figure 4.30 and Figure 4.31. This is caused by the sat function not being able to switch as fast as the sgn function, so there is a small performance trade-off in theory, but since in most cases the conventional sliding mode controller will present chattering and damage actuators in real life application, the adaptive boundary layer controller is the only viable practical controller.
Chapter 5. Conclusion

A conventional sliding mode controller was derived for a general second-order SISO system and a constant boundary layer and time-varying boundary layer were derived to help prevent chattering within the system’s input. This was also done for a general-second order MIMO system. A RR robot was simulated in Matlab/Simulink to demonstrate the use of sliding mode control with no boundary layer, with constant boundary layers, and with an adaptive boundary layer. Figure 4.2 and Figure 4.25 show that the constant boundary layer and the adaptive boundary layer were equal in preventing the chattering with no disturbance. Figures Figure 4.26-Figure 4.49 show that when the disturbance is added, the constant boundary layer is not able to prevent the chattering. However, the adaptive boundary layer is able to suppress chattering even when the disturbance is present.
References


