Avestia Publishing
Journal of Machine Intelligence and Data Science (JMIDS)
Volume 5, Year 2024
ISSN: 2564-3282
DOI: 10.11159/jmids.2024.015

Sliding Mode Control: Boundary Estimation and Hyperplane Transformation for MIMO Systems

Aashrita Mandalapu[1], Joshua Coleman[1], Agamemnon Crassidis[1]
[1]Rochester Institute of Technology, Department of Mechanical Engineering
1 Lomb Memorial Drive, Rochester, New York, USA 14623
am2313@rit.edu; jcc2126@rit.edu; alceme@rit.edu

***Abstract** - Due to the correlation between increasing system complexity and performance requirements, the need for globally robust control systems is more and more apparent. This paper proposes two extensions to a novel globally robust control system called sliding mode control (SMC). First, a new approach to estimating the boundaries of the influence matrix for a system without its model is developed. Next, the use of hyper-plane transformations in SMC is discussed. Both techniques were individually paired with sliding mode controllers and simulated against a nonlinear test system. In the future, these methods will be added to model free sliding mode control (MFSMC) to broaden the applicability of this type of controller.*

Keywords: Sliding Mode Control, Globally Robust Control, Estimation, Hyper-plane transformation, non-square MIMO system, Model Free Sliding Mode Control.

1. Introduction

As technology advances, systems modelling has become increasingly difficult. Furthermore, today's complicated, high performing systems may contain more nonlinearities. As a result, traditional control methods, such as PID control, are often used at the cost of efficiency and performance. Modern Lyapunov-based controllers are more robust, which is important when stability is desired even with modelling assumptions and uncertainties. In many cases, however, the number of uncertainties makes using most Lyapunov-based controllers impractical. For example, a sliding mode controller used to control a system with high uncertainties would require a large controller gain, and, as a result, a high control output.

To feasibly control these systems, multiple globally robust controllers have been developed. These controllers generally use two methods to achieve stability no matter the uncertainty: limiting the form of the error in the system or estimating the maximum uncertainty. One example of the first method was given in [1]. The LQR-based controller was proven to cause stability so long as the uncertainties were in the image of the control influence matrix. On the other hand, the controller in [2] would result in stability issues when a certain parameter was larger than a linear combination of the uncertainty and its derivative. An estimator was created to keep the parameter above the linear combination. Both controllers were tested and performed adequately but are limited in their scope (linear systems for [1] and second-order SISO systems for [2]). These two controllers illustrate the need for a broadly applicable globally robust controller; such a controller would guarantee stability no matter the system form.

For under-actuated systems many different control methods have been developed using SMC. [3] compared four different types of SMCs used for underactuated systems. No estimation techniques are presented for tuning the parameters and chattering was also present causing issues in the real-time implementation of the designed controls strategies. [4] uses a combination of LQR and SMC wherein the states in the underactuated system are split within two different loops. While the proposed design is effective for the system defined, disturbance is present at various time intervals. Also, it might not work well in the

Date Received: 2024-02-09
Date Revised: 2024-09-06
Date Accepted: 2024-10-04
Date Published: 2024-11-04

presence of uncertainties or when the system model is not accurate. The concept introduced in [5] follows complex mathematical formulations which could result in challenges during practical implementation.

A controller with the potential to be broadly globally robust was introduced in [6] and developed in [7] and [8]. The controller method – known as Model Free Sliding Mode Control (MFSMC) – solved the issue of large uncertainties in the system by avoiding a system model entirely. Instead, it relied on the unity gain equation and assumptions which are as follows:

$$\vec{\boldsymbol{x}}^{(\boldsymbol{n})} = \vec{\boldsymbol{x}}^{(\boldsymbol{n})} + \boldsymbol{B}[\vec{\boldsymbol{u}} - \vec{\boldsymbol{u}}_{\boldsymbol{k-1}}] + \vec{\boldsymbol{\varepsilon}} \tag{1}$$

$$\vec{\boldsymbol{\varepsilon}} = \boldsymbol{B}[\vec{\boldsymbol{u}}_{\boldsymbol{k-1}} - \vec{\boldsymbol{u}}] \tag{2}$$

$$\hat{\boldsymbol{\varepsilon}} = \hat{\mathbf{B}}[\vec{\mathbf{u}}_{\mathbf{k-2}} - \vec{\mathbf{u}}_{\mathbf{k-1}}] \tag{3}$$

$$|\vec{\boldsymbol{\varepsilon}}| < (1+\sigma_u)|\hat{\boldsymbol{\varepsilon}}| \tag{4}$$

$$\boldsymbol{B}_{\boldsymbol{ij,min}} < \boldsymbol{B}_{\boldsymbol{ij}} < \boldsymbol{B}_{\boldsymbol{ij,max}} \tag{5}$$

Where $\vec{\boldsymbol{x}}^{(\boldsymbol{n})} \epsilon \boldsymbol{R}^{\boldsymbol{n}\times\boldsymbol{1}}$ wherein n represents the system states, **B** is the control or input gain matrix and is given by $\boldsymbol{B}_{\boldsymbol{ij}} \epsilon \boldsymbol{R}^{\boldsymbol{n}\times\boldsymbol{m}}$ with m representing the number of inputs, $\vec{\boldsymbol{u}} \epsilon \boldsymbol{R}^{\boldsymbol{m}\times\boldsymbol{1}}$ is the control input and σ_u is the boundary constant in MFSMC. Sliding mode control techniques are then applied and the resultant controller is of the form:

$$\vec{\mathrm{u}} = \hat{\mathbf{B}}^{-\mathbf{1}}\left[-\tilde{\mathbf{x}} - \hat{\boldsymbol{\varepsilon}} - \left(\dot{\vec{\mathrm{s}}} - \tilde{\mathbf{x}}^{(\boldsymbol{n})}\right) - \vec{\kappa} \circ sgn(\vec{\mathbf{s}})\right] \tag{6}$$

$$\vec{\kappa} = |\boldsymbol{\beta} - I|\left(|\tilde{\boldsymbol{x}}^{(\boldsymbol{n})}| + |\dot{\vec{\mathbf{s}}} - \tilde{\mathbf{x}}^{(\boldsymbol{n})}|\right) + |\boldsymbol{\beta}(1+\sigma_u) - \boldsymbol{I}||\hat{\boldsymbol{\varepsilon}}| + \boldsymbol{\beta}\vec{\eta} \tag{7}$$

Where $\vec{\mathbf{s}}$ is the sliding condition, $\vec{\kappa}$ is variable boundary layer discontinuous switching gain, $\boldsymbol{\beta}$ is the square root of $\mathbf{B_{ij,max}}/\mathbf{B_{ij,min}}$, $\boldsymbol{I}$ is the identity matrix, $\vec{\eta}$ is discontinuous switching gain. A boundary layer $\vec{\Phi}$ was also introduced to reduce dithering. A derivation in a sliding mode context is given in [9].

These papers demonstrated MFSMC was applicable to all square systems (systems in which the number of control inputs is equal to the number of non-derivative states) so long as the bounds of the influence matrix were known. [10] proposed the use of an estimator for the parameter $\vec{\eta}$, which would avoid the need to know the bounds on the input matrix. Due to efficiency concerns, [11] adapted the estimator to find the matrix directly. While the controller-estimator combination stabilized several test systems, the performance of the estimator was sub-par in some circumstances.

Currently work is being performed to solve both limitations. In this work, both a method for estimating the bounds on the influence matrix and the use of a hyper-plane transformation for the control of non-square systems are presented. Results of using these techniques in a sliding mode context are presented. In the future, these methods will be adapted to MFSMC.

2. Sliding Mode Control

2.1. Boundary Estimation

Using a traditional estimator in MFSMC is more difficult than it may first seem – without a model, finding a suitable regression equation is not straightforward. The method in [11] was adapted using a least squares estimator with bounded gain forgetting. Rather than deriving the estimator from a relationship involving the influence matrix, [11] created a least-squares with bounded gain forgetting estimator using the sliding condition. The condition was developed as a part of sliding mode control:

$$\left(\dot{\vec{\Phi}} - \vec{\eta}\right) \circ |\vec{s}| \geq \vec{s} \circ \dot{\vec{s}} \tag{8}$$

$$\vec{\boldsymbol{s}} = \left[\frac{d}{dt} + \boldsymbol{\Lambda}\right]^{n-1} \tilde{\boldsymbol{x}} \tag{9}$$

Where $\boldsymbol{\Lambda}$ is positive definite matrix, $\tilde{\boldsymbol{x}}$ is the difference between the current states $\vec{\mathbf{x}}$ and their desired values $\vec{\mathbf{x}}_{\boldsymbol{d}}$ and "∘" is an element-wise product. When Eq. (8) is satisfied, the system is stable. The estimate is multiplied by a factor to estimate the matrix's bounds. The new technique similarly uses the sliding condition. Given a system of the form:

$$\vec{\boldsymbol{x}}^{(\boldsymbol{n})} = \vec{\boldsymbol{f}}\left(\vec{\boldsymbol{x}}^{(\boldsymbol{i})},\boldsymbol{t}\right) + \boldsymbol{B}\left(\vec{\boldsymbol{x}}^{(\boldsymbol{i})},\boldsymbol{t}\right)\vec{\boldsymbol{u}} \tag{10}$$

the estimator's goal is to find the influence matrix bounds. Start by substituting Eq. (10) into Eq. (9) and rearranging to obtain:

$$sgn(\vec{\boldsymbol{s}}) \circ \boldsymbol{B}\vec{\boldsymbol{u}} \geq sgn(\vec{\boldsymbol{s}}) \circ \left[\vec{\boldsymbol{x}}_{\boldsymbol{d}}^{(\boldsymbol{n})} - \vec{\boldsymbol{f}} - \left(\dot{\vec{\boldsymbol{s}}} - \tilde{\boldsymbol{x}}^{(\boldsymbol{n})}\right)\right] + \left(\dot{\vec{\Phi}} - \vec{\eta}\right) \tag{11}$$

Next, multiply both sides by a factor $\vec{\alpha}$ which is 1 when the sliding condition is not satisfied and -1 in all other situations. Note doing so flips the inequality. The factor generalizes the inequality to all states of the sliding condition.

At this point, the derivation diverges for coupled and decoupled systems. First, consider the case when the system is decoupled (**B** is diagonal). If $\vec{\mathrm{b}}$ is a vector of the diagonal elements of **B** ($\mathbf{B} = diag(\vec{\mathrm{b}})$), the left side reduces to:

$$\begin{aligned}\vec{\alpha} \circ sgn(\vec{\boldsymbol{s}}) \circ \boldsymbol{B}\vec{\boldsymbol{u}} &= \vec{\alpha} \circ sgn(\vec{\boldsymbol{s}}) \circ \vec{\boldsymbol{b}} \circ \vec{\boldsymbol{u}} \\ &= diag(\vec{\alpha} \circ sgn(\vec{\boldsymbol{s}}) \circ \vec{\boldsymbol{u}})\vec{\boldsymbol{b}}\end{aligned} \tag{12}$$

The final step is to define the following values:

$$\boldsymbol{A} = |diag(\vec{\alpha} \circ sgn(\vec{s}) \circ \vec{\boldsymbol{u}})| \tag{13}$$

$$\vec{q} = \vec{\alpha} \circ \left(\left|\dot{\vec{\Phi}} - \vec{\eta} + sgn(\vec{s}) \circ \left(\vec{\boldsymbol{x}}_{\boldsymbol{d}}^{(\boldsymbol{n})} - (\dot{\vec{\boldsymbol{s}}} - \widetilde{\vec{\boldsymbol{x}}}^{(\boldsymbol{n})})\right)\right| + \overline{\boldsymbol{f}} \circ (sgn(\vec{\boldsymbol{s}}) \tag{14}$$

$$\overline{\boldsymbol{f}} = \begin{cases} min(|\vec{\boldsymbol{f}}_{\boldsymbol{min}}|,|\vec{\boldsymbol{f}}_{\boldsymbol{max}}|), & \boldsymbol{B}_{\boldsymbol{min}} \text{ is estimated} \\ max(|\vec{\boldsymbol{f}}_{\boldsymbol{min}}|,|\vec{\boldsymbol{f}}_{\boldsymbol{max}}|), & \boldsymbol{B}_{\boldsymbol{max}} \text{ is estimated} \end{cases} \tag{15}$$

Using these definitions, the influence matrix bounds may be estimated as:

$$\boldsymbol{B}_{\boldsymbol{k|k}} = max\left(\boldsymbol{B}_{\boldsymbol{k|k-1}}, \boldsymbol{B}_{\boldsymbol{k-1|k-1}}\right) \tag{16}$$

$$\boldsymbol{B}_{\boldsymbol{k|k-1}} = diag(\boldsymbol{A}^{+}\vec{\boldsymbol{q}}) \tag{17}$$

Where $\boldsymbol{A}^{+}$ is the pseudoinverse of **A** and $\boldsymbol{B}_{\boldsymbol{k|k}}$ is the current bounds estimate. $\boldsymbol{B}_{\boldsymbol{k|k-1}}$ is the exact solution to the equation. However, if any of the current values of **B** are less than their corresponding maxima, some of the estimated values will be too low. For that reason, the actual estimate $\boldsymbol{B}_{\boldsymbol{k|k}}$ is found by comparing the equation's solution with the last estimates. A similar approach may be taken to derive an estimator for coupled systems. In testing, these definitions – which are derived from the maximizations of each side of the inequality – performed better than alternatives.

2.2. Hyper-plane Transformation

Using the MFSMC approach perfect tracking and stability for square MIMO systems have been obtained but additional methods are needed to guarantee tracking for underactuated (non-square) MIMO systems. A challenge in handling non-square systems is the non-invertibility of the input gain matrix, making the formulation of the control law impossible. A potential solution involves employing a coordinate transformation of the system. Through the transformation, the originally "non-square" matrix can effectively be "squared," overcoming the non-invertibility limitation and facilitating the formulation of the control law.

Consider the following system in the matrix form:

$$\boldsymbol{x} = \boldsymbol{f} + \boldsymbol{B}\boldsymbol{u} \tag{18}$$

Where $\boldsymbol{x}$ represents the system states matrix, **f** represents the system functions in the matrix form, [**B**] is the non-square matrix and **u** is the control input matrix. A transformation matrix, [**T**] is considered for converting into the hyperplane, y:

$$\vec{\boldsymbol{y}} = [\boldsymbol{T}]\vec{\boldsymbol{x}} \tag{19}$$

Where the dimensions of matrix [**T**] = the dimensions of $[\boldsymbol{B}]'$. Eq. 19 can be rewritten as:

$$\boldsymbol{y} = [\boldsymbol{T}]\boldsymbol{f} + [\boldsymbol{T}][\boldsymbol{B}]\boldsymbol{u} \tag{20}$$

and the product of **[[T][B]]** is now square and invertible. The matrix **[T]** can be thought of as a weighing matrix. Since the system under consideration is underactuated, and states cannot display perfect tracking simultaneously, **[T]** can be used to track certain outputs "more heavily" than others.

By applying SMC for a varying boundary layer, the control law is given by Eq. (21):

$$\begin{aligned}\boldsymbol{u} = (\boldsymbol{TB})^{-\boldsymbol{1}}[\boldsymbol{T}(-\hat{\boldsymbol{f}} + \ddot{\boldsymbol{x}}_{\boldsymbol{d}} - \lambda\boldsymbol{T}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\boldsymbol{d}}) \\ - k^{*}(x)sat(\boldsymbol{s}/\varphi))]\end{aligned} \tag{21}$$

Where $\hat{\boldsymbol{f}}$ is the best estimate of $\boldsymbol{f}$. Now:

$$k^{*}(x) = k(x) - \dot{\varphi} \tag{22}$$

Where $k^{*}(x)$ is the variable boundary layer discontinuous switching gain. $k(x)$ is given by:

$$k(x) = \boldsymbol{F}(\boldsymbol{x}) + \eta \tag{23}$$

Where $\boldsymbol{F}(\boldsymbol{x})$ is:

$$\boldsymbol{F}(\boldsymbol{x}) = |\boldsymbol{T}(\boldsymbol{f} - \hat{\boldsymbol{f}})| \tag{24}$$

The boundary layer equation is given by:

$$\dot{\varphi} + \lambda\,\varphi = k(x) \tag{25}$$

$$\varphi(0) = k\big(x_d(0)\big)/\lambda \tag{26}$$

To apply the model-free SMC method to an underactuated MIMO system, knowledge of the size of the **[B]** matrix of the system is required to formulate the transformation matrix **[T]**. Once that is known, the model-free SMC scheme can be developed in the y coordinate system, in a similar manner to the derivation in square MIMO systems, and **[T]** is used to relate y to x, and vice versa.

3. Results

3.1. Simulation Results for Boundary Estimator in SMC

Simulation results for the nonlinear system represented by Eqs. (27) and (28) performed for SMC with and without boundary estimation are given in Figures 1 and 2.

$$\ddot{x}_1 = -a_1(t)\dot{x}_1{}^2\cos(2x_1)\,x_2 + b_{11}(t)u_1 \tag{27}$$

$$\ddot{x}_2 = -a_2(t)\dot{x}_2{}^2\dot{x}_1 x_2 + b_{22}(t)u_2 \tag{28}$$

Unsurprisingly, the controller with known bounds reaches the desired states faster overall (though state 2 of the estimated controller beats that of the regular SMC). The discrepancy is due to the time spent estimating the boundaries. Figure 2 shows the boundary estimates are above the maximum values. While the amount of estimation would seem to cause inefficiencies, Figure 3 indicates the estimated SMC has a lower control output.

Figure 3. Desired (dotted) and simulated (solid) state trajectories for SMC with boundary estimation on a coupled system.

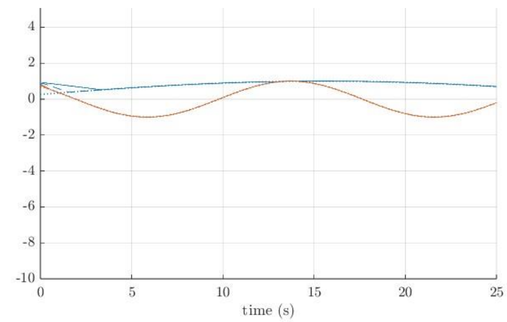


Figure 1. Tracking performance with and without boundary estimator for a constant B system.

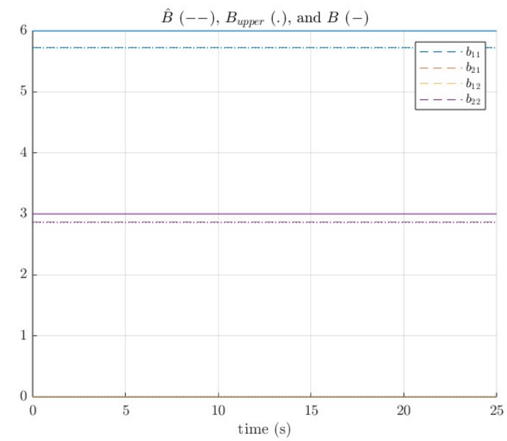
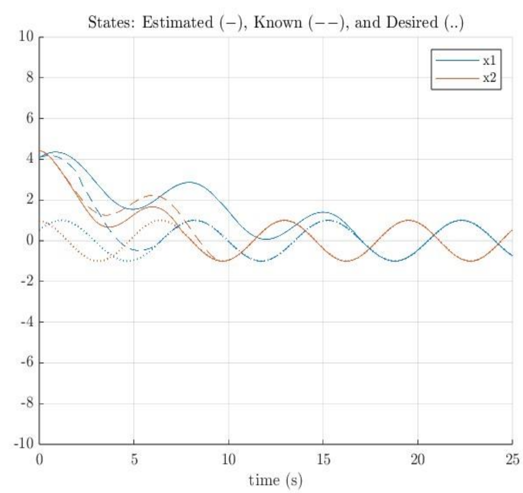


Figure 2. Best guess, estimated, and actual B values for a decoupled, constant B system.



Figures 4 and 5 show the SMC with boundary estimation's performance, this time against a coupled system. In Figure 5, the estimator is giving values of $\boldsymbol{\delta}$, a parameter related to **B** by:

$$\boldsymbol{\delta} = \boldsymbol{B}\widehat{\boldsymbol{B}}^{-1} - I \tag{29}$$

The test system is the same as in Eqs. (27) and (28), except with $b_{12}(t)$ and $b_{21}(t)$ terms. Again, state 1 is sluggish, but the system is still stable.

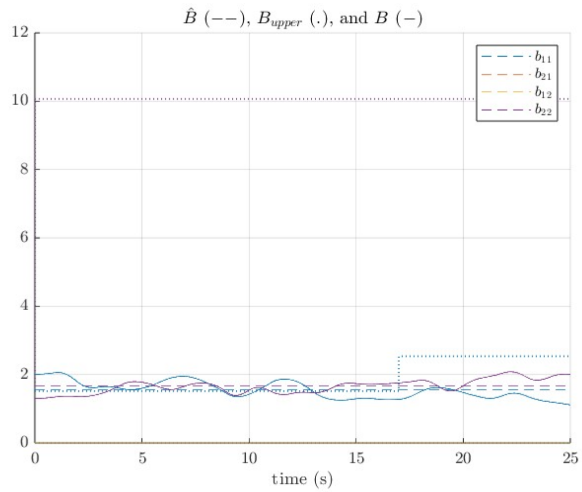


Figure 4. Desired (dotted), SMC (dashed) and SMC with boundary estimation (solid) state trajectories.

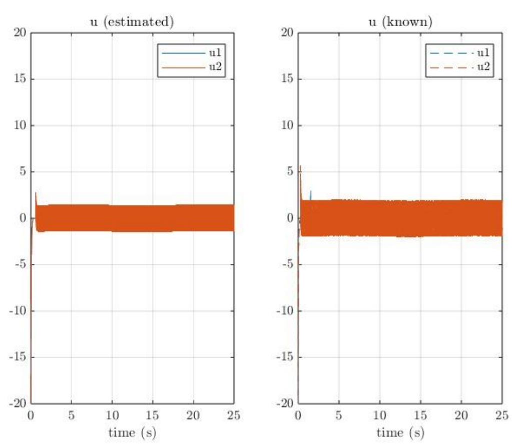


Figure 5. Real (solid), best guess (dashed) and estimated maximum (dotted) values of B

3.2. Simulation Results for Hyper Plane Transformation in SMC

Simulation for traditional SMC with hyperplane transformation including boundary layer for the given system below was carried out in Simulink with fixed-step ode5 (Dormand-Prince) solver for 10 seconds.

$$\ddot{x}_1 + a_1(t)[x_1 + \dot{x}_1]\cos(x_2) = u(t) \tag{30}$$

$$\ddot{x}_2 + a_2(t)[x_2 + \dot{x}_2 - \dot{x}_1]|x_1| = 0 \tag{31}$$

The parameter values considered for the simulation are given in table 1.

Table 1 Parameters and their values used in the simulation

Parameters	Value
Bounds of a_1	1 - 2
Bounds of a_2	3 - 4
Transformation Matrix, [T]	[1 0.01]
Lambda, λ	20
Ita, $\vec{\eta}$	0.1
Phi, $\vec{\Phi}$	0.1

Figures 6(a and b) and 7(a and b) from the simulation result plots, represent the closed-loop responses of the states of the system (x_1 , $\dot{x}_1$, x_2 , $\dot{x}_2$) and the desired signals. There is close to perfect tracking for x_1 and $\dot{x}_1$ with the desired signals but tracking of x_2 and $\dot{x}_2$ is lacking. This is because of the values set in the transformation matrix, **[T]**, x_1 and $\dot{x}_1$ is more heavily tracked than x_2 and $\dot{x}_2$. Based on the requirement, **[T]** matrix can be set to either track x_1 and $\dot{x}_1$ or x_2 and $\dot{x}_2$.

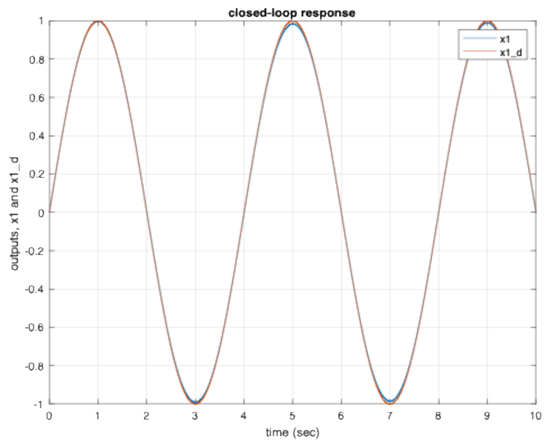


Figure 6a. Closed loop response of x_1 vs x_{1_d}

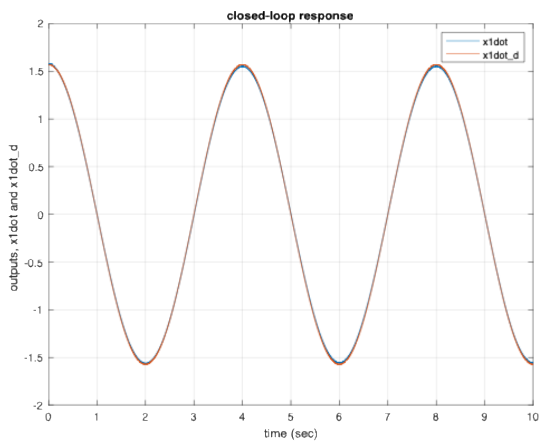


Figure 6b. Closed loop response of $\dot{x_1}$ vs $\dot{x_1}_d$

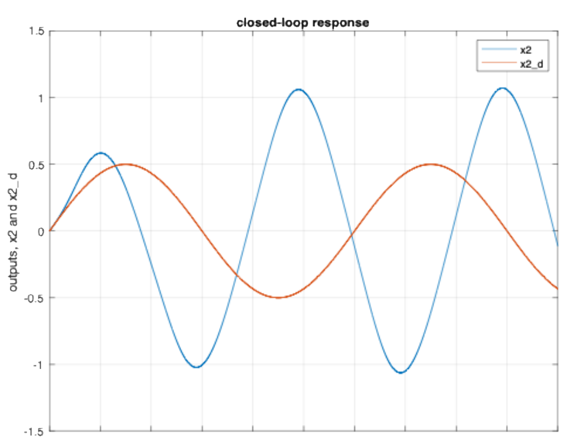


Figure 7a. Closed loop response of x_2 vs x_{2_d}

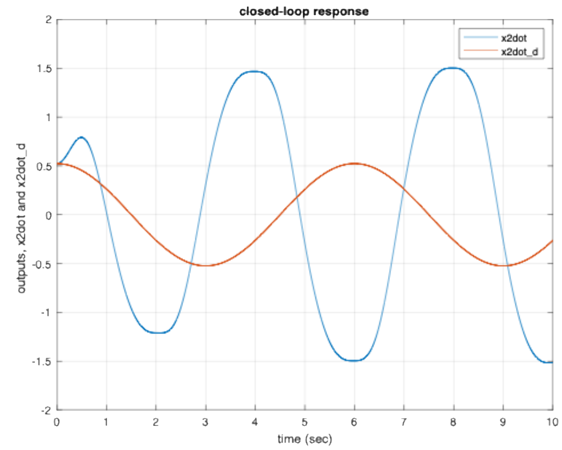


Figure 7b. Closed loop response of $\dot{x_2}$ vs $\dot{x}_{2_d}$

While the original signal tracking of the system is based on the **[T]** matrix, it can be seen from Figure 8a the transformed systems signal tracking is close to its desired value. The controller effort for the system is displayed in Figure 8b. Figure 9a shows that the system's sliding condition is satisfied which again proves the system is stable in the tracking sense. From Figure 9b, the S-trajectories lie within the defined boundary layer further proving the sliding condition is satisfied.

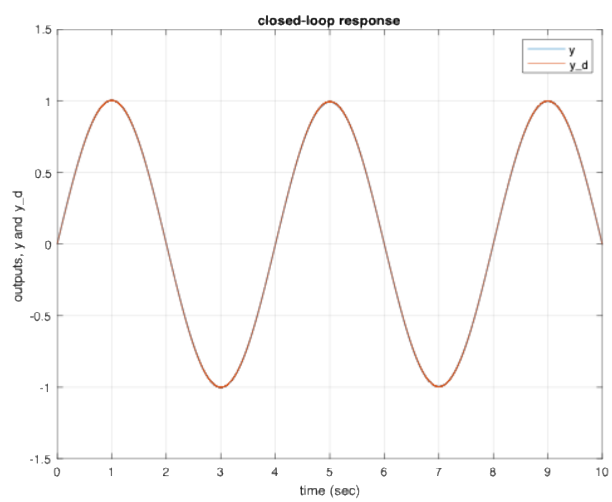


Figure 8a. Closed loop response of $\dot{y}$ vs $\dot{y}_d$

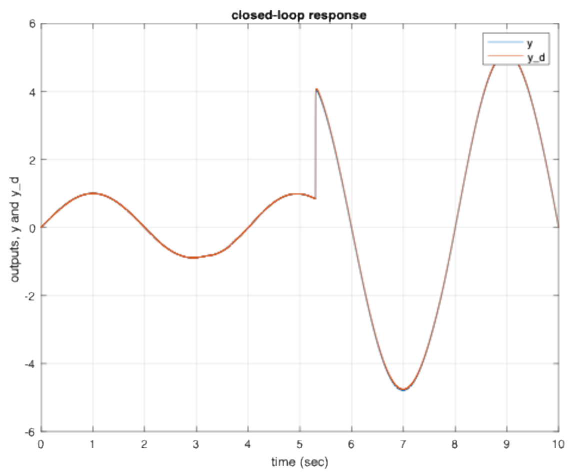


Figure 8b. Closed loop response of y_1 vs y_{1_d}

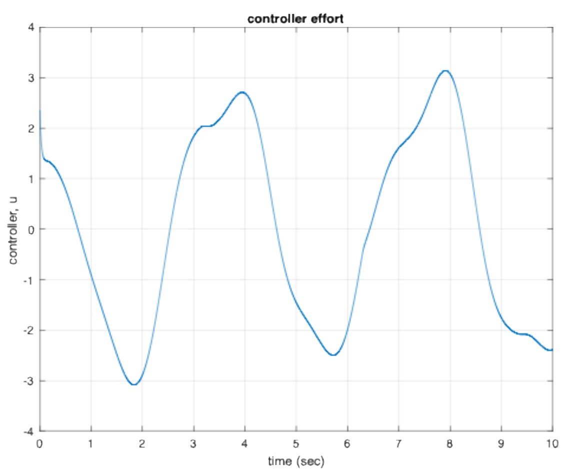


Figure 8c. Controller effort "u"

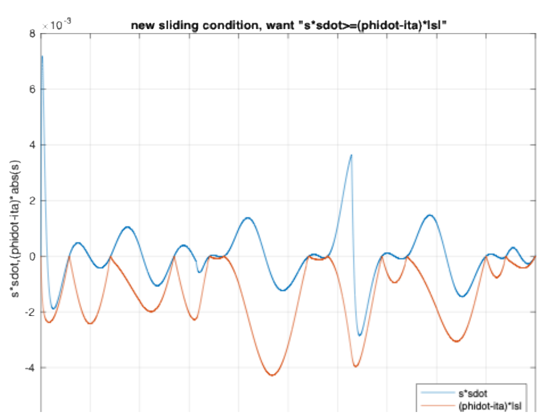


Figure 9a. Sliding condition satisfied

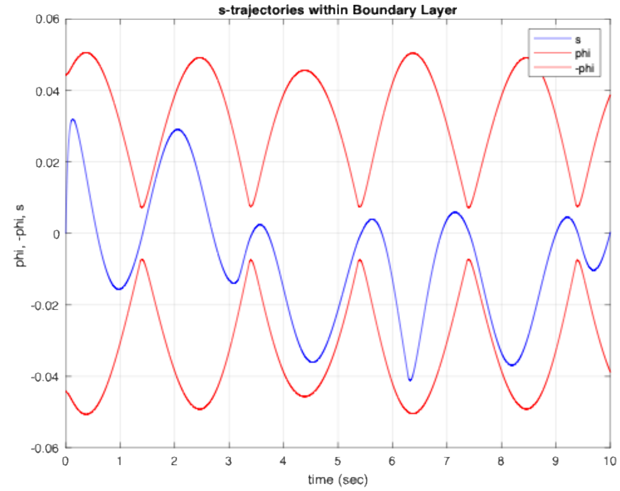


Figure 9b. Boundary layer condition satisfied

4.Conclusion

This paper introduced two methods: One, the integration of MFSMC approach with an online parameter (boundary) estimation method for complex nonlinear systems which allows for dynamic updates to the control law based on evolving system characteristics. The introduced approach includes a boundary layer to limit chattering and has precise tracking along with proven stability. The second method is the integration of hyperplane transformation with a traditional SMC for non-square MIMO systems with the inclusion of boundary layer. The tracking in this method depends on the values chosen in the transformation matrix. The next steps would be to apply both methods to model-free control and evaluate their performance. Upon obtaining the results for model-free control then this type of control approach can be applied to any physical systems making it more robust and stable.

References

[1] H.-P. Pang and G.-Y. Tang, "Global Robust Optimal Sliding Mode Control for a Class of Uncertain Linear Systems," presented at the Chinese Control and Decision Conference, 2008, pp. 3509–3512.

[2] Z. Huang and C. Sun, "Adaptive Global Robust Tracking Control for Uncertain Dynamic Systems," *Int. J. Control Autom. Syst.*, vol. 21, no. 4, pp. 1070–1079, Apr. 2023.

[3] S. Din, Q. Khan, F. Rehman, and R. Akmeliawanti, "A Comparative Experimental Study of Robust Sliding Mode Control Strategies for Underactuated

Systems," IEEE Access, Vol. 5, pp. 10068-10080, 2017

[4] R E. I. Mohamed and A. Karaarslan," Stabilization of Quadcopter Using the Combination of LQR And SMC Methods," International Marmara Sciences Congress Proceedings Book, 2022

[5] B Lu, Y Fang and Ning Sun," Continuous Sliding Mode Control Strategy for a Class of Nonlinear Underactuated Systems," IEEE transactions on Automatic Control, vol. 63, no. 10, 2018

[6] A. Mizov, "A Model-Free Control Algorithm Derived Using the Sliding Model Control Method," Rochester Institute of Technology, 2015.

[7] R. M. Reis, "A New Model-Free Sliding Mode Control Method with Estimation of Control Input Error," Rochester Institute of Technology, 2016.

[8] F. El Tin, "A Model-Free Control System Based on the Sliding Mode Control Method with Applications to Multi-Input-Multi-Output Systems," Rochester Institute of Technology, 2017.

[9] J.-J. E. Slotine and W. Li, "Chapter 7: Sliding Control," in *Applied Nonlinear Control*, Prentice Hall, 1991, pp. 277–310.

[10] M. S. Islam, "A Model-Free Control System Based on the Sliding Mode Control with Automatic Tuning Using as On-Line Parameter Estimation Approach," Rochester Institute of Technology, 2020.

[11] N. Hutson, "Model-Free Sliding Mode Control in the Lateral and Direction Dynamics of an Aircraft," Rochester Institute of Technology, 2023.