Proceedings of the 12th International Conference of Control Systems, and Robotics (CDSR 2025)
July 13, 2025 - July 15, 2025| Imperial College London Conference Center, London, United Kingdom
Paper No. 134
DOI: 10.11159/cdsr25.134

Sliding Mode Control of Non-square Systems with Non-unitary Input Gain

Aashrita Mandalapu[1], Sam Silliman[2], Agamemnon Crassidis[3]
[1,2,3]Rochester Institute of Technology
1 Lomb Memorial Drive, Rochester, New York, USA 14623
am2313@rit.edu; sos8648@rit.edu; alceme@rit.edu

Abstract - Complex non-square systems require optimal control strategies due to the challenge of limited controllability and observability. This paper proposes a new technique which would be an extension to a Sliding mode control scheme in which trajectories are driven towards a sliding surface, guaranteeing closed loop stability. In this proposed extension, sliding mode control is applied to single input-multiple output nonlinear underactuated system involving a real time estimation of the hyperplane transformation matrix to optimally track specific states. System modeling uncertainty due to unknown parameters is also fixed by implementing a real time estimation for the system parameters. Optimal tracking of specified states was demonstrated in each case, and improvement in control effort was seen when on-line parameter estimation for the system model was introduced.

***Keywords*:** Sliding Mode Control, Non-Square systems - underactuated, Hyperplane Transformation, Parameter estimation

1. Introduction

Non-square systems have unequal number of outputs and inputs and are ubiquitous in engineering. Such systems can be classified into underactuated (greater number of outputs compared to inputs) and over-actuated (greater number of inputs compared to outputs) systems. There is increased difficulty in designing control schemes for such systems because of the lack of controllability and observability of a specific state. One way this problem has been solved is by transformations wherein the number of outputs is made equal to the number of the inputs either by reducing or increasing the number states based on the system type. Singular value decomposition (SVD) is one of the oldest methods used for designing the control schemes for non-square systems [1-2]. While this method has been used for designing different types of control strategies, its highly impractical to implement it in real time due to its computational costs. It's also sensitive to noise and disturbance with an inability to account for actuator constraints making it ineffective.

Cascaded control also known as loop separation is another method wherein the system states to be controlled are split into different loops. This method is beneficial for controlling non-square systems as it helps in the control allocation among actuators efficiently and stabilizes the system when the loop responses are at different rates [3-4]. Even though cascaded control is a great strategy it has difficulties in terms of tuning, disturbance rejection and loop coordination while implementation.

Sliding mode control (SMC) is a well-established control scheme in which state trajectories are driven toward a constructed sliding surface on which trajectories are global asymptotically stable. But controlling non-square systems with SMC is notoriously difficult due to their noninvertible input gain matrix. Current approaches of controlling non-square systems with SMC include dynamic extension, wherein the derivatives of inputs are considered as additional inputs, making the system square. Alternatively, the Moore-Penrose Pseudoinverse [5-6] can be applied to the input gain matrix to make the system square. The drawback to these approaches is that they only allow for perfect tracking of one state, which is determined by the dynamics of the system being controlled.

This issue can be addressed with the application of the hyperplane transformation matrix [7], which instead of increasing the number of input states as in dynamic extension or applying a pseudoinverse matrix, reduces the number of outputs of the system to be equal to the number of inputs. In doing so, the choice of hyperplane transformation matrix can be made to specify which state is weighed more in the control effort, enabling perfect tracking of a specified state.

In this paper, we propose a novel extension of this technique that involves uniting the time varying hyperplane transformation matrix with on-line parameter estimation to allow for real time estimation of the hyperplane transformation to optimally track specified states. This novel control scheme is then applied to an example non-square system with non-unitary input gain. Additionally, the control effort is improved by implementing on-line system parameter estimation.

2. Hyperplane Transformation Matrix Estimator Architecture

The estimation of the hyperplane transformation matrix [T] is handled by a gradient estimator which is used to drive the tracking error of weighted states to zero. This is done by defining different weighting of specific states in the **prediction error**, which drives the estimator. A gradient estimator is chosen for its simplicity and applicability to linear parametrization models whose representation is same as the hyperplane transformation, which is given by,

$$y = Tx \tag{1}$$

The form of the linear parametrization models required for gradient estimation necessitates the estimation of the transpose of the hyperplane transformation matrix $\boldsymbol{T^T}$, which is inconsequential as the output of the estimator can simply be transposed to return the hyperplane transformation matrix. The prediction error chosen to drive the gradient estimator is given by Eq. (2).

$$e = \tilde{x}^T T^T - \tilde{x}^T Q^T T^T \tag{2}$$

Where "Q" is the weighting matrix that controls which, states are estimated and the tracking error vector, $\boldsymbol{\tilde{x}}$, serves as the excitation of the estimator. A steady state hyperplane transformation matrix can only be reached by either minimizing tracking error in the case where a state is heavily weighted by the "Q" matrix *or* minimizing the entry in the hyperplane transformation matrix corresponding to the state when it is not heavily weighted. This suggests that for a specified state to be tracked, the weighting matrix should be a diagonal matrix corresponding to the specified desired state. The hyperplane transformation matrix (transpose) update from the gradient estimator is given by Eq. (3)

$$\frac{d}{dt}(T^T) = -p_0 \frac{\partial(e^2)}{\partial(T^T)} = -p_0 \tilde{x} e \tag{3}$$

With "p_0" being the estimator gain. The lack of explicit reference to any model suggests that this estimator architecture can be applied to any system and eventually to a model free controller [8].

2.1. Implementation of Hyperplane Transformation Matrix in SMC

The system model chosen to implement the hyperplane transformation matrix was based on a nonlinear two mass-spring-damper system with an applied force on one of the masses. This system was selected for its open loop stability and the model's state space representation is given by Eq. (4).

$$\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{k_1}{m_1} & 0 \\ 0 & \frac{k_1}{m_2} \end{pmatrix} \begin{pmatrix} -(x_1 - x_2) - (x_1 - x_2)^3 \\ (x_1 - x_2) + (x_1 - x_2)^3 \end{pmatrix} + \frac{k_2}{m_1} \begin{pmatrix} 0 \\ -x_2 - x_2^3 \end{pmatrix} + \begin{pmatrix} \frac{b_1}{m_1} & 0 \\ 0 & \frac{b_1}{m_2} \end{pmatrix} \begin{pmatrix} -(\dot{x}_1 - \dot{x}_2) \\ (\dot{x}_1 - \dot{x}_2) \end{pmatrix} + \frac{b_2}{m_2} \begin{pmatrix} 0 \\ -\dot{x}_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{m_1} \\ 0 \end{pmatrix} u \tag{4}$$

Where k_1, k_2, m_1, m_2, b_1, b_2 are all system parameters that are unknown, bounded and constant in time while B is the non-unitary input gain which is equal to $\frac{1}{m_1}$. For notational purposes, the model can also be expressed as Eq. (5), with the definitions of its terms given from Eqs. (6)-(10).

$$\ddot{x} = K_1 f_1(x) + K_2 f_2(x) + B_1 g_1(\dot{x}) + B_2 g_2(\dot{x}) + Bu \tag{5}$$

$$K_1 = \begin{pmatrix} \frac{k_1}{m_1} & 0 \\ 0 & \frac{k_1}{m_2} \end{pmatrix};\ f_1(x) = \begin{pmatrix} -(x_1 - x_2) - (x_1 - x_2)^3 \\ (x_1 - x_2) + (x_1 - x_2)^3 \end{pmatrix} \quad (6)$$

$$K_2 = \frac{k_2}{m_1};\ \ f_2(x) = \begin{pmatrix} 0 \\ -x_2 - x_2^3 \end{pmatrix} \quad (7)$$

$$B_1 = \begin{pmatrix} \frac{b_1}{m_1} & 0 \\ 0 & \frac{b_1}{m_2} \end{pmatrix};\ g_1(\dot{x}) = \begin{pmatrix} -(\dot{x}_1 - \dot{x}_2) \\ (\dot{x}_1 - \dot{x}_2) \end{pmatrix} \quad (8)$$

$$B_2 = \frac{b_2}{m_2};\ g_2(\dot{x}) = \begin{pmatrix} 0 \\ -\dot{x}_2 \end{pmatrix} \quad (9)$$

$$B = \begin{pmatrix} \frac{1}{m_1} \\ 0 \end{pmatrix} \quad (10)$$

For this system, the sliding surface is defined by Eq. (11)

$$s = \dot{y} - \dot{y}_d + \lambda(y - y_d) \quad (11)$$

Differentiating the sliding surface, yields Eq. (12).

$$\dot{s} = \ddot{y} - \ddot{y}_d + \lambda(\dot{y} - \dot{y}_d) \quad (12)$$

According to hyperplane transformation,

$$y = Tx \quad (13)$$

The derivatives of y are given by differentiating Eq. (13),

$$\dot{y} = T\dot{x} + \dot{T}x \quad (14)$$

$$\ddot{y} = T\ddot{x} + 2\dot{T}\dot{x} + \ddot{T}x \quad (15)$$

Plugging in Eq. (14) and Eq. (15) to Eq. (12) gives,

$$\dot{s} = T\ddot{x} - T\ddot{x}_d + 2\dot{T}(\dot{x} - \dot{x}_d) + \ddot{T}(x - x_d) + \lambda T(\dot{x} - \dot{x}_d) + \lambda\dot{T}(x - x_d) \quad (16)$$

Further calculations can be simplified using the function definition in Eq. (17).

$$\Gamma = -T\ddot{x}_d + 2\dot{T}(\dot{x} - \dot{x}_d) + \ddot{T}(x - x_d) + \lambda T(\dot{x} - \dot{x}_d) + \lambda\dot{T}(x - x_d) \quad (17)$$

This finally gives Eq. (18) for the derivative of the sliding surface.

$$\dot{s} = T\ddot{x} + \Gamma \quad (18)$$

Plugging in the system model equation (Eq. (5)), to Eq. (18) gives Eq. (19).

$$\dot{s} = T\big(K_1f_1(x) + K_2f_2(x) + B_1g_1(\dot{x}) + B_2g_2(\dot{x})\big) + TBu + \Gamma \quad (19)$$

Trajectories on the sliding surface should be forced to stay on the sliding surface. To force this to be the case, Eq. (19) is set to zero and the control effort is solved for given by Eq. (20).

$$u = -(TB)^{-1}\big(T\big(K_1f_1(x) + K_2f_2(x) + B_1g_1(\dot{x}) + B_2g_2(\dot{x})\big) + \Gamma\big) \quad (20)$$

A switching gain must be added so that surfaces off the sliding surface are driven towards it, satisfying the sliding condition $s\dot{s} \leq -\eta|s|$ and guaranteeing global asymptotic stability of trajectories. Ensuring that the switching gain is placed in the correct location to satisfy the sliding condition and replacing system parameters with best estimates yields Eq. (21).

$$u = -(TB)^{-1}\left(T\left(\hat{K}_1f_1(x) + \hat{K}_2f_2(x) + \hat{B}_1g_1(\dot{x}) + \hat{B}_2g_2(\dot{x})\right) + \Gamma + K\mathrm{sgn}(s)\right) \quad (21)$$

To solve for the switching gain, the sliding condition must be used, with substitution of $\dot{s}$ followed by u. Doing so yields Eq. (22):

$$-\eta|s| \geq s\dot{s} = T\big(K_1 f_1(x) + K_2 f_2(x) + B_1 g_1(\dot{x}) + B_2 g_2(\dot{x})\big)s$$
$$-TB\big(T\hat{B}\big)^{-1}\left(T\left(\hat{K}_1 f_1(x) + \hat{K}_2 f_2(x) + \hat{B}_1 g_1(\dot{x}) + \hat{B}_2 g_2(\dot{x})\right) + \Gamma + K\mathrm{sgn}(s)\right)s + \Gamma s \tag{22}$$

Simplifying Eq. (22) requires the simplification given by Eq. (23).

$$(TB)\big(T\hat{B}\big)^{-1} = \left(\begin{pmatrix}T_1 & T_2\end{pmatrix}\begin{pmatrix}B\\0\end{pmatrix}\right)\left(\begin{pmatrix}T_1 & T_2\end{pmatrix}\begin{pmatrix}\hat{B}\\0\end{pmatrix}\right) = (T_1 B)\big(T_1\hat{B}\big)^{-1} = B\hat{B}^{-1} \tag{23}$$

Substituting Eq. (23) in Eq. (22) gives Eq. (24),

$$-\eta|s| \geq s\dot{s} = T\left(\big(K_1 - B\hat{B}^{-1}\hat{K}_1\big)f_1(x) + \big(K_2 - B\hat{B}^{-1}\hat{K}_2\big)f_2(x) + \big(B_1 - B\hat{B}^{-1}\hat{B}_1\big)g_1(\dot{x}) + \big(B_2 - B\hat{B}^{-1}\hat{B}_2\big)g_2(x)\right)s$$
$$+(1 - B\hat{B}^{-1})\Gamma s - B\hat{B}^{-1}K|s| \tag{24}$$

Since the real values of the system parameters are unknown but bounded, their best estimates can be geometric mean of the upper and lower bounds defined in Eqs. (25)-(30).

$$\hat{k}_1 = \sqrt{k_{1,u}k_{1,l}} \tag{25}$$
$$\hat{k}_2 = \sqrt{k_{2,u}k_{2,l}} \tag{26}$$
$$\hat{b}_1 = \sqrt{b_{1,u}b_{1,l}} \tag{27}$$
$$\hat{b}_2 = \sqrt{b_{2,u}b_{2,l}} \tag{28}$$
$$\hat{m}_1 = \sqrt{m_{1,u}m_{1,l}} \tag{29}$$
$$\hat{m}_2 = \sqrt{m_{2,u}m_{2,l}} \tag{30}$$

A gamma function is defined for the simplification of Eq. (24),

$$B\hat{B}^{-1} = \gamma \tag{31}$$

Using these expressions, Eq. (24) can be solved for the switching gain, leading to Eq. (32).

$$K|s| \geq T\big(K_1\ \gamma - \hat{K}_1\big)f_1(x)s + T\big(K_2\,\gamma - \hat{K}_2\big)f_2(x)s + T\big(B_1\,\gamma - \hat{B}_1\big)g_1(\dot{x})s$$
$$+T\big(B_2 - B\hat{B}^{-1}\hat{B}_2\big)g_2(x)s + (\ \gamma^{-1} - 1)\Gamma s +\ \gamma^{-1}\eta|s| \tag{32}$$

To be the most conservative, the absolute value of each term is taken so that the $|s|$ terms cancel, resulting in Eq. (33).

$$K = \big|T\big(K_1\ \gamma - \hat{K}_1\big)f_1(x)\big| + \big|T\big(K_2\,\gamma - \hat{K}_2\big)f_2(x)\big| + \big|T\big(B_1\,\gamma - \hat{B}_1\big)g_1(\dot{x})\big|$$
$$+\big|T\big(B_2 - B\hat{B}^{-1}\hat{B}_2\big)g_2(x)\big| + \big|(\ \gamma^{-1} - 1)\Gamma\big| +\ \gamma^{-1}\eta \tag{33}$$

Using the methods outlined in [9], the discontinuous signum function can be replaced with a smooth saturation function with a dynamic boundary later. Doing so removes chattering, smoothing the control effort. The updated control effort becomes Eq. (34),

$$u = -(TB)^{-1}\left(T\left(\hat{K}_1 f_1(x) + \hat{K}_2 f_2(x) + \hat{B}_1 g_1(\dot{x}) + \hat{B}_2 g_2(\dot{x})\right) + \Gamma + K^*\mathrm{sat}(s/\phi)\right) \tag{34}$$

With $K^*(\boldsymbol{x})$ given by Eq. (35).

$$K^*(x) = K(x) - \dot{\phi} \tag{35}$$

The sliding condition with the new boundary layer becomes Eq. (36), which forces trajectories outside of the boundary layer to be forced towards the boundary layer. The controller form in Eq. (34) can be shown to reproduce the updated sliding condition, Eq. (36).

$$s\dot{s} \leq \big(\eta - \dot{\phi}\big)|s|, \qquad |s| > \phi \tag{36}$$

Update in the boundary layer is given by Eq. (37), with its initial condition given by Eq. (38).

$$\dot{\phi} + \lambda\phi = K(x_d) \tag{37}$$
$$\phi(0) = \eta/\lambda \tag{38}$$

The control system parameters η and λ will dictate the convergence and control effort. When implementing the hyperplane transformation matrix estimator, a small constant δ was added to the product "(TB)" in Eq. (12) so that the reciprocal does not diverge if the first entry of the hyperplane transformation matrix is near zero.

2.2. Real Time Estimation of system parameters

To avoid the impact of the additional system parameter uncertainty, the system parameters were estimated in real time. This was done with a least-squares estimator with exponential forgetting [10] with prediction error given by Eq. (39)

$$e = W\hat{a} - \ddot{x} \tag{39}$$

With excitation matrix "W" given by Eq. (40) and estimated parameter matrix given by Eq. (41). In Eq. (40), the vectors separated by commas indicate matrix concatenation, so that "W" is a 2×7 matrix.

$$W = \left[f_1(x), f_1(x), f_2(x), g_1(x), g_1(x), g_2(x), \begin{pmatrix}1\\0\end{pmatrix} u\right] \tag{40}$$

$$\hat{a} = \begin{pmatrix}\frac{k_1}{m_1} & \frac{k_1}{m_2} & \frac{k_2}{m_2} & \frac{b_1}{m_1} & \frac{b_1}{m_2} & \frac{b_2}{m_2} & B\end{pmatrix}^T \tag{41}$$

The form of the excitation for the system parameter estimator leverages the linear form of the model given by Eq. (5) and only requires nonzero excitation to achieve convergence, not perfect tracking of states. The update of the system parameter guesses is given by Eq. (42).

$$\dot{\hat{a}} = PW^T e \tag{42}$$

With the estimator gain matrix, "P", calculated with Eq. (43)

$$\dot{P} = \lambda_f P - PW^T WP \tag{43}$$

With "λ_f" being the forgetting factor. Since the system parameters are now being estimated in real time, gain margins can no longer be used to simplify the switching gain expression.

3. Simulation Results for Hyperplane Transformation in SMC

The estimator for the hyperplane transformation matrix was implemented on the nonlinear, second order spring mass damper system defined in Eq. 4. Real time estimation of the system parameters is carried out to improve the control effort of the sliding mode controller. The control system designed was simulated in Simulink and MATLAB using an ode5 solver and was tested for tracking both the states x_1 and x_2. The key simulation parameters used in the testing of this control law are detailed in Table 1. To test the hyperplane transformation matrix estimator, an initial guess of $T_0 = (.1 \quad .1)$ was considered and the weight matrix, "Q", was varied.

Table 1: Key simulation parameters for unitary input gain simulations.

Parameter	Range	Value
k_1	$[1, 2]$	2
k_2	$[1, 3]$	2.5
b_1	$[2,4]$	3.5
b_2	$[3, 5]$	3
m_1	$[5,15]$	10
m_2	$[15,25]$	20
$\hat{k}_1$	N/A	$\sqrt{2}$
$\hat{k}_2$	N/A	$\sqrt{3}$
$\hat{b}_1$	N/A	$2\sqrt{2}$
$\hat{b}_2$	N/A	$\sqrt{15}$
κ_1	N/A	$\sqrt{2}$
κ_2	N/A	$\sqrt{3}$
β_1	N/A	$\sqrt{2}$
β_2	N/A	$\sqrt{5/3}$
η	N/A	0.35
λ	N/A	30
δ	N/A	0.01
p_0	N/A	1
x_{1d}	N/A	$\sin(\pi t/2)$

x_{2d}	N/A	$(1/2)\sin(\pi t/3)$

3.1 Tracking of state x_1

The system was simulated for $t = 25$ using a time step of $dt = 0.0001$. To track state x_1, a weight matrix of $\boldsymbol{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ was chosen. Fig. 1 shows that a close to perfect tracking of the desired state x_1 is achieved by the controller.

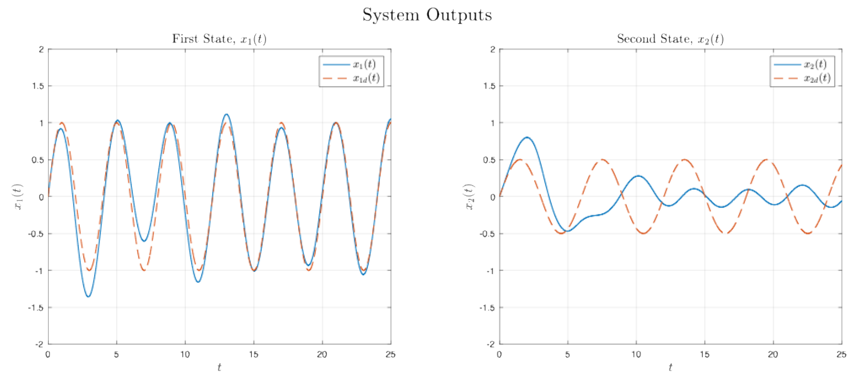

Fig. 1: Tracking of States $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ while Hyperplane transformation matrix is estimated to track state $\boldsymbol{x_1}$.

This tracking is achieved through rapid convergence to an ideal hyperplane transformation matrix, shown in Fig. 2.

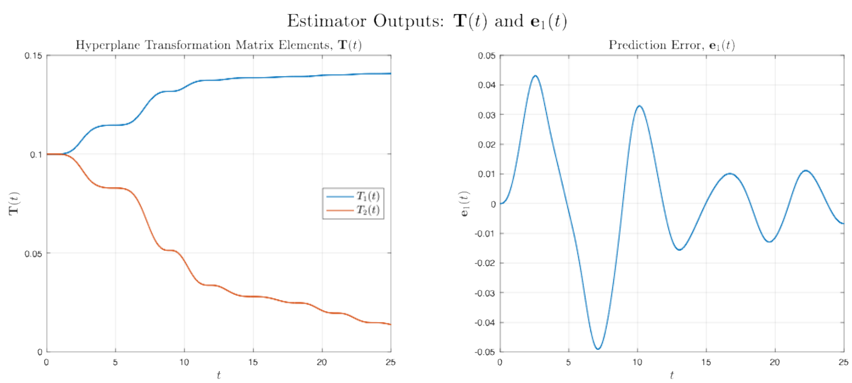

Fig. 2: Estimation of hyperplane transformation matrix and the prediction error driving estimation.

The control effort remains smooth as seen in Fig. 3. The magnitude of the control effort is attributed to the complexity of the system with a nonunitary input gain.

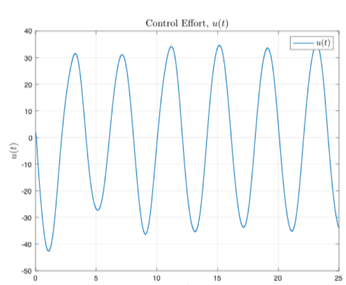

Fig. 3: Control effort with estimation of the hyperplane transformation matrix to track state $\boldsymbol{x_1}$ of nonunitary input gain system with system parameter estimation.

The sliding surface remained within the boundary layer, so the sliding condition was met for all time, shown in Fig. 4. The magnitude of the sliding surface remains well within the boundary layer and near zero, due to the elimination of the error between estimates of system parameters and their true values.

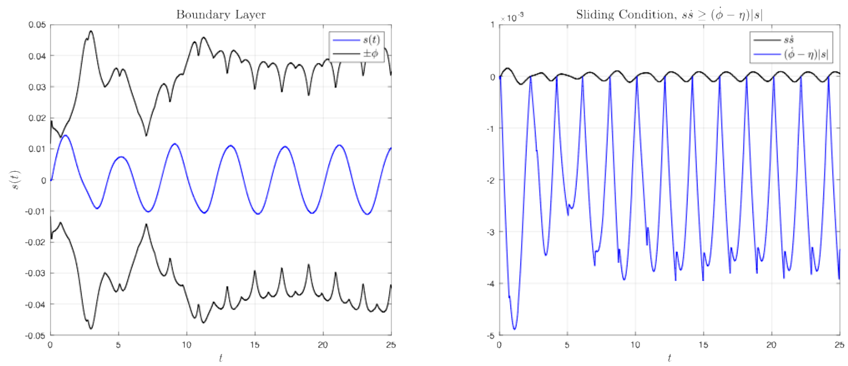

Fig. 4: Sliding surface inside boundary layer and the sliding condition being satisfied ($\boldsymbol{s\dot{s}} \leq (\boldsymbol{\dot{\phi}} - \boldsymbol{\eta})|\boldsymbol{s}|$ for $|\boldsymbol{s}| > \boldsymbol{\phi}$) for the system.

3.2 Tracking of state $\mathbf{x_2}$

To track state x_2 the same system parameters and configuration was considered except for the weight matrix, $\boldsymbol{Q} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. It can be seen from Fig. 5 that a good tracking of the state x_2 was achieved.

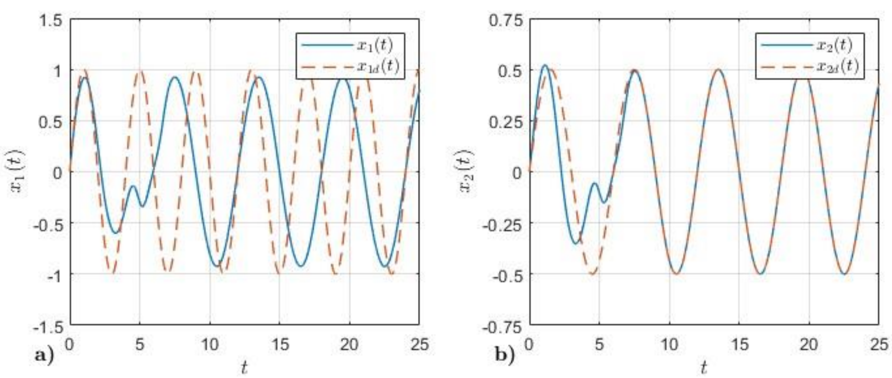

Fig. 5: States a) $\boldsymbol{x_1}$ and b) $\boldsymbol{x_2}$ of the nonunitary input gain system with system parameter estimation. Hyperplane transformation matrix is estimated to track state $\boldsymbol{x_2}$.

The remaining simulation results for tracking the state x_2 are identical to the results obtained for tracking state x_1. The control effort is smooth, and the boundary layer and the sliding condition were satisfied proving the stability of the system.

I. Conclusion

A novel application of SMC to non-square systems involving the real time estimation of the hyperplane transformation matrix using a gradient estimator to allow for optimal tracking of specified states has been proposed. The control law was implemented for a spring mass damper nonlinear system with nonunitary input gain. Simulation of the non-square system shows that perfect tracking of desired states is achieved with a smooth control effort that may be implemented on a real system. The sliding condition remains met for all time, guaranteeing stability of the system. High frequency oscillations in the control effort stemming from an unmodeled disturbance introduced to the control law to prevent divergence of the system were investigated and eliminated with real time estimation of system parameters.

In the future, this technique will be further tested through implementation on non-square systems with more than two outputs states. The hyperplane transformation matrix estimator will also be implemented in a model-free SMC architecture to allow for model free control of non-square systems.

References

[1] W. P. Hunek," New SVD-based matrix H-inverse vs. minimum-energy perfect control design for state-space LTI MIMO systems," in *Proceedings of the IEEE International Conference on System Theory, Control and Computing*, Sinaia, Romania, 2016.

[2] Qi-B. Jin," Design of a multivariable internal model controller based on singular value decomposition," *Canadian J. Chemical Engineering*, vol. 91, pp. 1103-1114, 2013.

[3] J. O. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. and Computer Sci. Eng., Massachusetts Institute of Technology., MA.

[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] Zhang, W., Ou, L., He, X., & Zhang, W," On the pole of non-square transfer function matrix Moore–Penrose pseudo-inverses," *International J. of Systems Science*, vol. 46, pp. 2560–2571, 2013.

[6] R. Q. A. Yousuf and J. Uhlmann," On Use of the Moore-Penrose Pseudoinverse for Evaluating the RGA of Non-Square Systems," *Iraqi J. Computers, Communications, Control & Systems Engineering*, Vol. 3:21, pp. 89-97, 2021.

[7] A. Mandalapu and J. Coleman, "Sliding Mode Control: Boundary Estimation and Hyperplane Transformation for MIMO Systems," *J. Machine Intelligence and Data Science*, vol. 5, pp. 136-143, 2024.

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

[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.