NEURO-SUPER TWISTING SLIDING MODE CONTROL OF A REACTION WHEEL

A Thesis Submitted to the

Department of Aerospace Engineering

INSTITUTE OF SPACE SCIENCE AND ENGINEERING

An Affiliate of

AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfilment of the Requirements for the Degree of

MASTER OF SCIENCE

By

IDOKO SUNDAY ATTAH

(Student ID: 40832)

Institute of Space Science Engineering
www.isse.edu.ng
National Space Research and Development Agency,
Airport Road Abuja, Nigeria

African University of Science and Technology
www.aust.edu.ng
P.M.B. 681, Garki, Abuja
F.C.T, Nigeria.

SEPTEMBER 2021

DEDICATION

This work is dedicated to the giver of life and sustenance, the one that had made the program a success.

CERTIFICATION

This is to certify that the thesis titled, "**NEURO-SUPER TWISTING SLIDING MODE CONTROL OF A REACTION WHEEL**", submitted to the school pf postgraduate studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of the Master's degree, is a record of original research carried out by Idoko Sunday Attah in the Aerospace Department of the Institute of Space Science and Engineering (ISSE), affiliate of AUST.

IDOKO SUNDAY ATTAH		22/11/2021
Name	Signature	Date

NEURO-SUPER TWISTING SLIDING MODE CONTROL OF A REACTION WHEEL

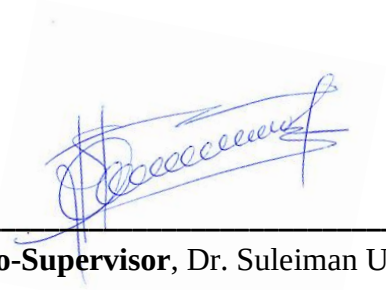
By

IDOKO SUNDAY ATTAH

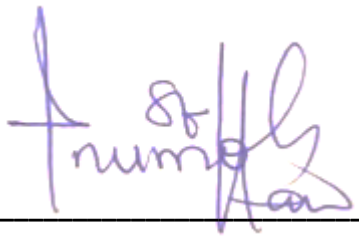
A THESIS APPROVED BY THE DEPARTMENT OF AEROSPACE ENGINEERING

RECOMMENDED:

Supervisor, Prof. Olufemi Agboola — 22/11/2021 Date



Co-Supervisor, Dr. Suleiman U. Hussein — 22/11/2021 Date



Co-Supervisor, Dr. Lawrence O. Inumoh — 22/11/2021 Date

APPROVED:

Head, Department of Aerospace Enginerring — 22/11/2021 Date

ACKNOWLEDGMENTS

With all indebtedness, I will like to sincerely appreciate my HOD and supervisor, Prof. Olufemi Agboola, for his fatherly advice, instructions, and timely supervision. His wealth of experience and knowledge had tremendously contributed to the overall outlook of this work. Sincerely, I am grateful, Sir.

Also, my gratitude goes to Dr. Suleiman U. Hussein for his patience, support, guidance, and direction and for being the chief motivation of this Master's thesis. His encouragements, recommendations, and insightful comments at critical points of this research work are immeasurable. Honestly, I could never have imagined a better supervision and guide for this work. Furthermore, my earnest appreciation goes to my boss and co-supervisor, Dr. Deji Inumoh, for his understanding, practical thinking, and advice, which had contributed immensely to the realization of this thesis.

Aside from my supervisors, I would like to express my gratitude to the lecturers at the Department of Aerospace Engineering, Institute of Space Science and Engineering (ISSE), for their tutelage and for providing the necessary theoretical and practical tools needed for this research work. Their contributions cannot be over-emphasized. Additionally, my sincere gratitude also goes to governing board of both ISSE and the African University of Science and Technology (AUST). This avenue could not be possible without them. A big thank you to them for the support. Moreover, all these will not be possible without the permission and support of the Administration of National Space Research and Development Agency.

Special thanks go to my fellow coursemates and Ph.D. Students at the Institute, for the inspiring and thought-provoking discussions, for the sleepless nights we were working together before deadlines and exams, and for all the fun we have had in the time we spent during our course work.

Lastly, I would like to thank my family; my beautiful wife and daughter, Mrs. Roseline Attah and Miss Ziv Idoko, my parents, and my siblings. They were light in dark tunnels and immense support when I needed it the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Sliding mode controllers are renowned for their robustness and fast dynamic response in speed control of direct current motors. The major drawback to their application is the chattering phenomenon which is harmful to actuators like the reaction wheel and causes performance degradation. To reduce this effect, the super twisting sliding mode control is employed because it tends to resist varying load parameters and reduce chattering. However, the super twisting SMC has two gain pairs that need to be tuned via the trial-and-error method to attain the optimum performance of the controller. This process of trial-and-error tuning makes the implementation of the super twisting SMC hectic and stressful. Therefore, an adaptive method of tuning these gain pairs is proposed in this work that can automatically tune the gain pairs to the required optimum value that matches the change in varying parameters, thereby making it robust to changes. This method is achieved by designing, training, and implementing a shallow neural network called fitnet, which is a multilayer perceptron, to effectively fit the sliding variable to the target curve in finite time. In doing so, the neural network targets the nominal control taking input from the sliding variable and its delayed data, and then predicts the appropriate nominal control input needed to keep the system states state trajectories on the sliding surface in finite subsequent time. More so, this work was simulated using MATLAB/Simulink and a corresponding enhanced performance objective (rise time of 0.7471s, settling time of 1.3493, peak value of 1.001, zero percentage overshoot and undershoot, and steady-state error of 5.951e-05) over the PID, conventional SMC, and super-twisting SMC were recorded respectively. In addition, the proposed neuro-tuned SMC gave an improved control performance when compared with the work of Rakhonde and Kulkarni and Morfin et al.

Keywords: Sliding Mode Control, Super Twisting, DC motor, Neural Network, reaction Wheel, Speed control

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

In present times, the need for electrical machines cannot be overemphasized (Monteiro et al., 2015). Most industrial and domestic processes are carried out with the aid of one or more electrical motors since they have proven to be very effective in both larger and smaller-scale applications. In particular, many research works had been geared towards making these electric drives more efficient, and as such, both software and hardware components of these drives had been studied, leading to their preference owing to their simplicity in structure, lightweight, and small size (Galphade & Sankeshwari, 2015; JyotiRath et al., 2012). As such, the direct current (DC) motors have the following pros that entail, but are not limited to; high efficiency, force power, and speed control had made them more attractive in applications (Hafez et al., 2019), hence the vast use and its indispensability. These qualities had contributed to the use of electric motors (dc motors) in space applications, majorly in; thrust vector control (TVC) actuators, solar array deployment, control moment gyroscopes (CMG), low thermal emission applications, fuel valve control actuators, lightweight and high RPM applications as they apply to spacecraft (Murugesan, 1981; Roving et al., 1991).

Conventionally, in controlling the attitude and stability of spacecraft, the reaction wheels, which are flywheels attached to a dc motor, are used, amongst other active control devices. These reaction wheels are momentum exchange devices that supply reaction torque and store angular momentum when energy is applied or removed from the flywheel (Mehrjardi et al., 2015). By maintaining flywheel rotation, it is possible to stabilize an axis of the spacecraft. A number of reaction/momentum wheels can be employed to give full three-axis attitude control and stability. To achieve this stability and attitude control via reaction wheels, it is necessary to design an effective feedback control system that will enable the reaction wheel to deal with external disturbances or turbulences viewed in a non-linear form.

Historically, feedback controls date back to ancient times, with increasing advances in applications being noted as time progressed; but clear and distinct applications of the feedback control systems were seen or employed in the 17th century with the introduction of systems that can control the temperature in furnaces, rolls in mills, and provide necessary regulations in steam engines (Dewey, 2018). Even with these advantages in applications, the oscillatory

behaviour of these systems still posed a major drawback, as depicted in the relay-based control systems, which is a clear representation of the primordial case of the Variable Structure System (VSS) (Tsypkin, 2003). These variable structure systems are discontinuous nonlinear control systems that operate by the application of high-frequency switching that follows a switching logic as dictated by the control law (Emel'yanov, 2007). More so, the predominant feature of the Variable Structure Control (VSC) is the Sliding Mode Control (SMC) (Mahmoud, 2018; Samantaray & Chakrabarty, 2020), which is one of the commonly applied control techniques in resolving issues of nonlinear uncertain systems (Edwards & Spurgeon, 1998; Emel'yanov, 2007; Sabanovic et al., 2011; Shtessel et al., 2014).

As with all forms of variable structure systems, the sliding mode suffered less attention during these initial periods due to the predominant/prevalent issue resulting from the complexity posed by chattering with sensors, actuators, and switching systems. Nonetheless, in recent times, because of its robustness, finite-time convergence, reduced-order compensated dynamics, and technological advancement, higher-order and enhanced methods have been applied to sliding mode control, thereby increasing its ability to counter and minimise the chattering effects the earlier mode had. However, seeing the effect posed by present systems in correcting the major drawbacks experienced in the application of these techniques, it has become crucial to reduce or even eliminate high-frequency switching of sliding mode controllers, which excites unmodelled dynamics in the feedback control system (Utkin et al., 2017), hence the need for a technique that is capable of chattering reduction, online tuning of the controller, reduction of reaching time, force compensation, fast finite-time convergence, high precision, removal of singularity, and learning algorithm when disturbances arise as the device is in space, or in operation. These functions can be realized through the use of a curve-fitting neural network which is designed and implemented to give a better chattering reduction and overall improved control performance.

1.2 Statement of the Problems

Many research and projects have been carried out, and much more are currently in progress that are geared toward controlling the attitude and stabilization of the spacecraft while in operation. The ability to properly regulate the application of highly controllable torque using low power with discontinuous functions of state disturbances and desired inputs as its control in achieving stability had been the major application of variable structure control or systems.

However, the reaction wheels – which use brushless dc motor systems, have uncertainty and nonlinear characteristics that affect the performance of the controllers coupled with the high-

frequency switching process that generates a negative phenomenon called chattering, which arises from the interaction between parasitic dynamics and finite-frequency switching control. To resolve this, a more robust-hybrid system/technique with good dynamic properties as finite-time convergence and insensitivity to disturbances is desirable to effectively eliminate or reduce the chattering effects on sensors and actuators used.

1.3 Aim and Objectives of the Study

This study aims to develop an adaptive neural-tuned super twisting sliding mode control system that uses a feedback control law to effectively drive the system states to origin by making it asymptotically stable equilibrium.

To attain this aim, the study will seek to achieve the following objectives:

i. Design a super twisting sliding mode control for the brushless dc motor in a reaction wheel model
ii. Develop and implement the neural network architecture for optimal determination of the gains of the super twisting sliding mode control.
iii. Carry out a comparative performance evaluation, analysis, and validation of the system using standard metrics, and also comparing the obtained results with classical Proportional-Integral-Derivative (PID) techniques and the works of (Morfin et al., 2017), (Rakhonde & Kulkarni, 2018), and relating it to (Dursun et al., 2017).

1.4 Significance of the Study

This research work is very useful in a wide range of industrial and traction applications, including applications like robotics, consumer electronics, security surveillance, transportation, and space technology – particularly, reaction wheels. Furthermore, this research is also applicable to a wide range of disciplines; academic researchers, robotic experts, medical machinery, aerospace, and aeronautic engineers, and industrial engineers, since all the advantageous characteristics of the brushless DC motor electric machines and drives are put to use in these fields of endeavour.

Moreover, it will contribute to interdisciplinary discourse because the proper understanding of sliding mode control and its application to the robust control of electric drives reinforces the understanding of other related disciplines explored in this study, such as robotics, aerospace, aeronautic, neural networks, etc. These symbiotic research relationships will bolster the advancement of research in these fields.

1.5 Scope of the Study

The variable structure systems (VSS) are discontinuous nonlinear control systems that function by altering the dynamics of the system through high-frequency switching control, and sliding mode control – as a VSC – covers a very wide area of discourse in decision making and control ranging from autonomous systems, sensor networks, control systems, energy systems, aerospace systems, automation, manufacturing, smart grids, nonlinear systems, power systems, robotics, amongst other. Even brushless dc motors are used in various mechatronics applications. Likewise, there are numerous applications of the different types of sliding mode control.

Therefore, the scope of this research work is limited to the use of super-twisting sliding mode control for the control of brushless direct current motor as used in reaction wheels applied in space technology.

CHAPTER TWO
LITERATURE REVIEW

2.1 General Introduction

In this Chapter, the available literature was reviewed in line with the subject matter of the research project, which is the application of sliding mode control in stabilization and speed control of brushless DC motor. The central themes which are reviewed include the fundamental principles governing the design and selection of sliding mode control with electrical drives – as well as – relevant works that had been employed in making the technique more efficient and applicable to space technology. The general understanding of concepts and principles applied in this work is explained in this section.

2.2 Review of Fundamental Concepts and Theories

The underlying idea in attitude control, as it applies to space technology, is gotten majorly from the application of electrical, mechanical, and control principles. Attitude control can be realized by using the reaction wheels, which are momentum generating devices. These reaction wheels are made up of electromechanical dc motors connected to a flywheel and placed at a predefined location (axis) in the spacecraft to generate torque that will counter disturbances that affect the orientation and pointing accuracy of the spacecraft. As such, the control is included to provide the adequate maneuvering of the flywheels with inputs to the electric motors on board. These concepts are described later in this section.

2.2.1 Electrical Machines

Electrical machines are devices that convert energy in the electrical form to mechanical form and vice versa. These devices generally work on the basic concept of electromagnetic induction (Daware, 2012), which entails the motion or variation between a conductor and a magnetic field leading to the production of voltage called the Electromotive Force (EMF). A simple analogy is shown in Figure 2.1.

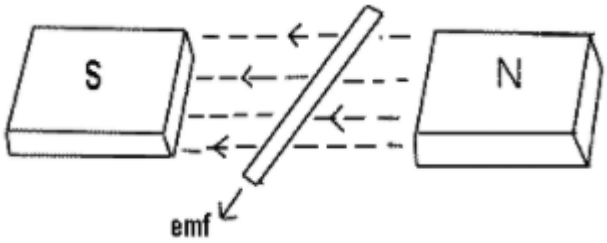


Figure 2.1: Electromagnetic Induction (Daware, 2012)

Besides, some electrical machines do not convert energy from mechanical to electrical or vice versa. They just transform from one level to another, that is, these types of machines convert from one voltage level to a higher or smaller voltage level and are called transformers (Electrical Classroom, 2020). So, in a general sense, electrical machines can be grouped into three main types, which are namely;

- Motors
- Generators
- Transformers

The simple explanation of the difference between the electric motor and the electrical generator principle is illustrated in Figure 2.2.

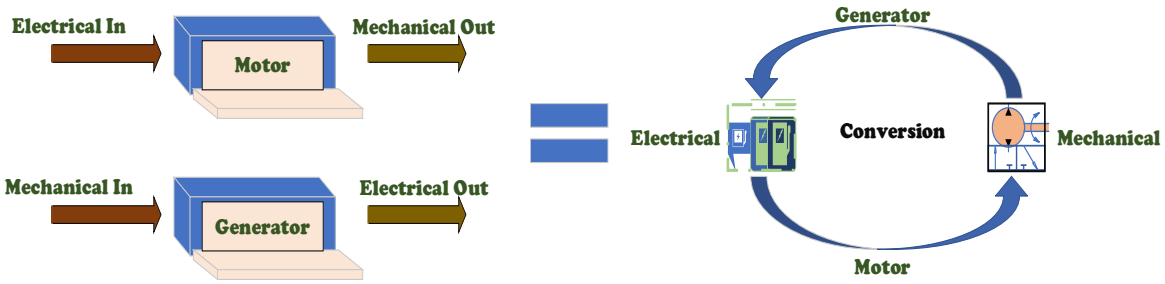


Figure 2.2: Interrelationship between motor and generator (Chaudhari, 2019)

From Figure 2.2, it can be deduced that a motor is an electrical machine that converts or changes electrical energy to mechanical energy. This means it takes as input a form of electricity and outputs motion (mechanical energy) (Chaudhari, 2019). In structure, the motor setup comprises two main parts, which are; the rotor and the stator. These two parts are arranged in a way to enable a current-carrying conductor that is placed in a magnetic field to experience motion, as indicated in Figure 2.1.

Unlike the electric motor, a generator is an electric machine that converts or changes mechanical energy (motion) into electrical energy. This is achieved when a conductor moves in a magnetic field; an emf gets induced within the conductor. This phenomenon is called generator action, and like the electric motor, a generator fundamentally consists of a stator and a rotor (Diesel Service & Supply, 2016). Whereas, in the case of the transformer, it operates purely on the concept of mutual induction occurring between its primary and secondary windings, which implies that they transfer electric power from one circuit to another. When an alternating voltage is applied at its primary windings, magnetizing currents flow through it, which leads to the production of flux that is situated mainly at the magnetic core path of the transformer (Dasgupta, 2002). This flux forms the link between the primary and secondary

windings that are not connected electrically. Hence, the voltage that is self-induced in the primary winding is mutually induced in the secondary winding. Thus, the voltage across the two windings becomes a factor in the number of turns inherent in the two windings. This is termed the "turn ratio" of the transformer, and it can be represented by the equation below and graphically illustrated as in Figure 2.3.

$$Turn\ ratio\ (T.R) = \frac{V_{pri}}{V_{sec}} = \frac{N_p}{N_s} \qquad \ldots (2.1)$$

where V_{pri} and V_{sec} are the primary and secondary voltage, respectively, N_p and N_s correspond to the number of primary and secondary windings.

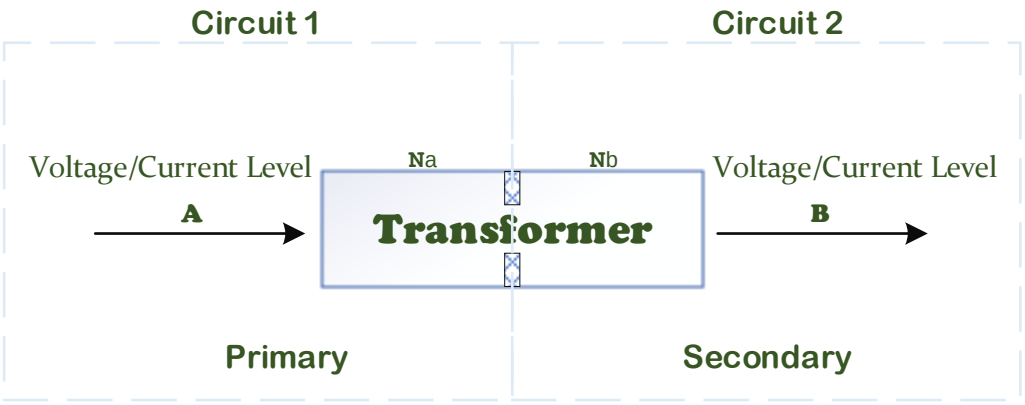


Figure 2.3: Structure of a Transformer (Apogeeweb, 2019)

Fundamentally, an electric machine can be classified under dynamic and static categories. A general classification of the electric machine is shown in Figure 2.4.

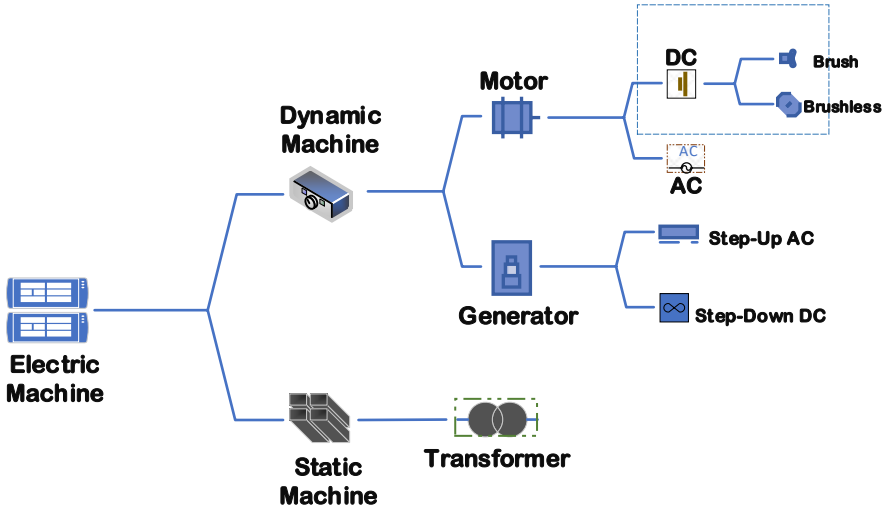


Figure 2.4: Classification of Electric Machine (Electrical Classroom, 2020)

2.2.2 Direct Current Motors

As the name denotes, direct current motor is a type of electrical machine that solely converts direct electrical current to mechanical energy (motion) (Jones, 2018). This direction of motion or mechanical force is given by Fleming's left-hand rule, which obeys the principle of the action that is induced when a current-carrying conductor is placed in a magnetic field. This force can be represented by the equation given below;

$$F = B \times i \times l \quad (newtons) \qquad \ldots (2.2)$$

Where F is the mechanical force, B is the magnetic field, i is the current through the conductor, l is the length of the conductor. As well, when the armature rotates under the influence of the driving torque, a counter electromotive force (EMF) is induced as a result of the armature conductor moving through the magnetic field. This counter emf, also called back emf, acts in the opposite direction to the voltage applied to the system and helps to make the dc motor a self-regulating motor – since it regulates the flow of armature current to meet the load requirements. Mathematically, the back emf can be expressed in relation to the magnetic flux, speed of the motor, number of armature parallel paths, and armature conductors thus;

$$E_b = \Phi Z N \times \left(\frac{P}{A}\right) \qquad \ldots (2.3)$$

where E_b is back emf, Φ = magnetic flux, Z = number of armature conductors, N = speed of the motor in RPM, P = No of poles, A = number of armature parallel paths.

As the motor is in operation and running at the speed of N, torque is being developed by the armature, called the armature torque, T_a. Hence, the power developed can be expressed as:

$$P_d = T_a \times 2\pi N \quad (watts) \qquad \ldots (2.4)$$

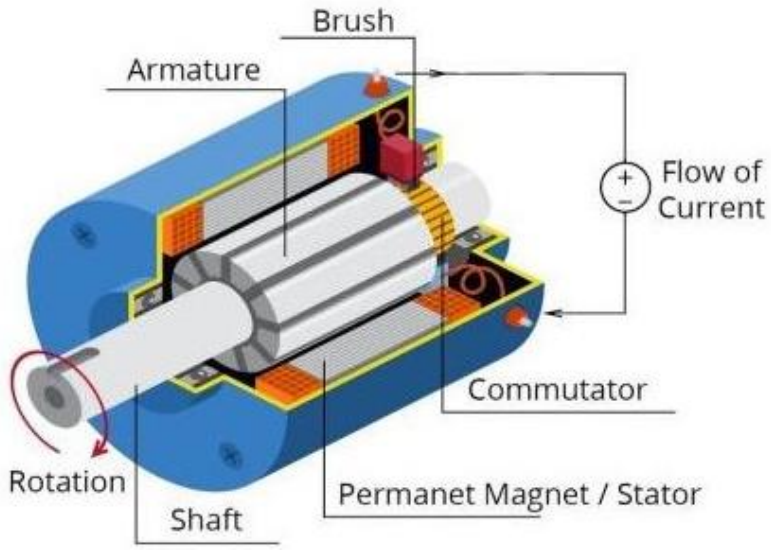


Figure 2.5: DC motor setup (Jones, 2018)

Strikingly, in construction, there is no basic difference between a dc motor and a dc generator, because in most cases, their use can be interchanged (Daware, 2012),(Theraja & Theraja, 2005). The elements that make up the dc motor, as shown in Figure 2.5 above, can be classified into four, which are; the rotating part (armature), the field windings (stator), commutator (acting as a switch that intermittently reverses the current between the external circuit and the rotor), and the field magnets to turn the axle connected to the motor.

2.2.3 Sliding Mode Control

Sliding mode control is a nonlinear control technique that is suitable and robust enough to cater to systems exhibiting uncertainties, changes in the parameters, and disturbances (Ahmed et al., 2019; Derbel et al., 2017; Fridman et al., 2016; Gambhire et al., 2021; Maghfiroh et al., 2020; Perruquetti & Barbot, 2002; Sabanovic et al., 2011; Shtessel et al., 2014; von Ellenrieder, 2021). According to Spurgeon (2014), the sliding mode control is described as a variable structure control that is characterized by some feedback control laws and decision law. So this control technique is based on the variable structure system, which is principally known for its switching logic operating between discontinuous or continuous functions (Hung et al., 1993; Sabanovic et al., 2011). Ideally, as the system is made to move along a line or surface, it is said to be sliding. This sliding happens at the switching surface when the system states are directed towards the surface's attributes. So, if the switching surface function is given as:

$$s(x) = 0 \qquad \ldots (2.5)$$

The system state vectors will tend towards zero (0) or remain in it if already at zero. Thus, the state trajectory x will be towards the positive and negative sides of the sliding surface, resulting in a discontinuity on the differential system(Martínez & Cao, 2019; Wu et al., 2014b). The two reaching points can be expressed as:

$$\lim_{s \to 0^+} \dot{s} < 0 \qquad \ldots (2.6)$$

For the negative side and tending towards zero, and for the positive side and tending towards zero as shown

$$\lim_{s \to 0^-} \dot{s} > 0 \qquad \ldots (2.7)$$

Assuming a dynamical nonlinear system described by the equation

$$\dot{x} = f(x(t), u(t), t) \qquad \ldots (2.8)$$

where $x(t) \in \boldsymbol{R}^p$ = system state vector, $u(t) \in \boldsymbol{R}^q$ = control input to the system and the sliding surface selected as $s(x) = 0$, and $s(x)$ is called the switching function or sliding variable. This is of the form

$$u(t) = \begin{cases} u^+, & s(x) < 0 \\ u^-, & s(x) > 0 \end{cases} \quad where \ u^+ \neq u^- \qquad \ldots (2.9)$$

This is illustrated in Figure 2.6 below;

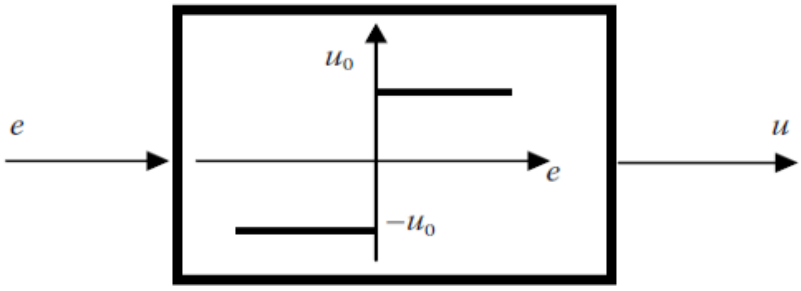


Figure 2.6: The Switching function (Utkin et al., 2017)

The control input aims to force this sliding variable to zero. So, the specifications for sliding mode are achieved if the system achieves the slides along the switching surface. Haven defined the sliding surface; the second step is to design a control law that will keep the system in the trajectories of the sliding hyperplane in the specified finite time. These behaviours can be represented as illustrated in Figure 2.7.

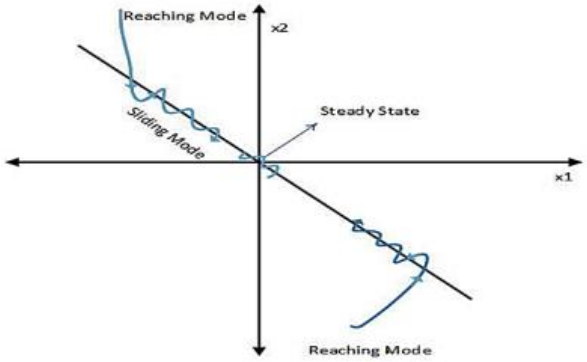


Figure 2.7: Sliding Surface (Samantaray & Chakrabarty, 2020)

Hence, the sliding mode process can be addressed under two conditions (Wu et al., 2014b).

Condition 1: The sliding mode conditions have to be met in finite time and subsequently maintained for all time. This entails designing the sliding mode controller to achieve the reachability of the sliding mode

Condition 2: The dynamics of the sliding mode have to be stable with some definite performance. This involves designing the sliding mode surface that entails the stability of the system dynamics to some certain performances.

Consequently, as equation $s(x) = 0$ holds at finite time, the derivative of the selected switching variable has to follow the equation;

$$s(x)\dot{s}(x) < 0 \qquad \ldots (2.10)$$

Which can be expressed as

$$\dot{s}(x) = \left.\begin{matrix} \dot{s}(x) < 0, & when\ s(x) > 0 \\ \dot{s}(x) > 0, & when\ s(x) < 0 \end{matrix}\right\} \qquad \ldots (2.11)$$

This is the condition that will enable the state to move towards achieving or reaching the sliding surface parameters. Hence, this condition is termed the "reaching condition" (Wu et al., 2014a), and the accompanying system state trajectories are called the "reaching phase" (Hung et al., 1993). There are four (4) classical reaching laws; they are; constant rate, exponential, reaching power rate, and general reaching law (Jinkun Liu, 2017).

Thus, putting these two conditions into perspective, the process of employing sliding mode control can be classified into two steps which are:

Step 1: First thing is to select a sliding surface, s(x) = 0, such that the system dynamics constrained to it will exhibit properties such as disturbance rejection capability, stability, and tracking.

Step 2: Secondly, design a control law that will keep the system state trajectories attracted to the selected sliding surface and maintained on the sliding surface in finite subsequent time.

Some of the basic kinds of sliding surfaces designed are linear, integral, and terminal sliding surfaces (Wu et al., 2014a). They are briefly described in Table 2.1

Table 2.1: Sliding surface designs

Switching Functions	**Equation**	**Reference**
Linear sliding surface	$\mathrm{s(x)} = \mathrm{B^T(x,t)}\left[\frac{\partial \mathrm{V(x)}}{\partial \mathrm{x}}\right]^{\mathrm{T}} = 0,$ Where V(x) is the Lyapunov function for the system, B is the control matrix	(Wu et al., 2014a)
Integral sliding surface	$s(x) = Cx(t) - Cx(t_0) - C\int_{t_0}^{t}(A + BK)x(\tau)d\tau$	(Bessas et al., 2016; Lian et al., 2010)
Terminal sliding surface	$\left.\begin{matrix} s(x_1, x_2) = x_2(t) + \beta {x_1}^{q/p} = 0, \\ \beta > 0, \end{matrix}\right\}$ x_1, x_2 are the system's states p and q are +ve integers that satisfy $p > q$	(Skruch & Długosz, 2019; Vo & Kang, 2019)

Although, according to (Samantaray & Chakrabarty, 2020), the most conventional choice of sliding variable is a linear combination of the following representation:

$$s = \dot{e} + c_0 e$$
$$s = \ddot{e} + c_1 \dot{e} + c_0 e$$
$$s = e^{(k)} + \sum_i^{k-1} c_i \, e^{(i)}$$

Typically, if the sliding surface depends only on a single scalar parameter, c, it becomes

$$s = \left(\frac{d}{dt} + c\right)^{k-1} e \qquad \ldots (2.12)$$

Meanwhile, after selecting the appropriate desired sliding surface, the next step is to design a control law that will steer the system trajectories towards the selected sliding hyperplane in finite time. So, two major algorithms will be discussed. They are first-order sliding mode control (FOSMC) – to represent the traditional SMC, and second-order sliding mode control (SOSMC) – to represent the higher order of sliding mode control.

2.2.3.1 First order sliding mode control

As stated by Jianxing Liu et al., (2020), the FOSMC is the most conventional SMC algorithm in use, and it takes the form

$$u = -c \cdot sign(s) = \begin{cases} -c, & s > 0 \\ c, & s < 0, \end{cases} \qquad \ldots (2.13)$$

Where c is a suitably large positive constant.

Studying Eq. (2.13), it can be deduced that there are two values of c with opposite signs corresponding to the value of s, indicating that the control will continue to switch across the hyperplane $s = 0$. Consequently, this control u will affect the derivative of s such that when $s > 0$, u will be $-c$, and $\dot{s} < 0$. Also, the situation will be reversed when $s < 0$, it implies that $\dot{s} > 0$. Therefore, the relation in Eq. ($\dot{s}s < 0$) will hold until the sliding surface condition is achieved.

This high-frequency switching, which is theoretically infinite in steady-state, will cause a problem with the actuator of switching devices on board. To counter or resolve this system chattering issue, a higher-order SMC is employed which has a modified algorithm that accounts for the high frequency switching phenomena. The commonly used algorithm in the higher-order SMC is the SOSMC.

2.2.3.2 Second-Order Sliding Mode Control

Fundamentally, if s and its derivative $\dot{s}$ are both zero at finite time, that is:

$$s = \dot{s} = 0, \ \forall t \geq T \qquad \ldots (2.14)$$

and meanwhile the second derivative, $\ddot{s}$ is discontinuous, that is, $\ddot{s} \neq 0, \quad \forall t \geq T$, then, it is a second-order sliding mode control (SOSMC).

Considering a nonlinear sliding variable

$$s = s(x_1, x_2) = x_2 + c|x_1|^{1/2} sign(x_1), \quad c > 0 \qquad \ldots (2.15)$$

It can be deduced that the sliding variable is not a straight line anymore due to its nonlinearity. The sliding manifold is $x_2 + c|x_1|^{1/2} sign(x_1) = 0$

The challenge is to design the control u that drives the variable to zero in finite time and maintain it at zero. The controller u that drives the sliding variable to zero in finite time is given as:

$$\left.\begin{array}{l} u = -\lambda sign(s) \\ this\ implies \\ u = -\lambda sign\left(x_2 + c|x_1|^{1/2} sign(x_1)\right) \end{array}\right\} \qquad \ldots (2.16)$$

Where the positive gain λ is set to a sufficiently large figure

As such, if the control u – tasked with forcing the nonlinear system to go in line with the selected sliding manifold – makes the controlled output and its first derivative zero and maintains them there in the presence of bounded disturbance, then the second-order sliding mode is said to be taking place (JKD Power and Energy Solutions, 2021). This control law is called the control with prescribed convergence law (Jianxing Liu et al., 2020).

2.2.4 Super Twisting Sliding Mode Control

One conventional form of second-order sliding mode control (SOSMC) is the super twisting sliding mode control. It is suitable for nonlinear systems, and the objective of the control is to drive the derivative of the sliding variable $\dot{s}$ and the sliding variable s itself to zero in finite time where unknown time-varying external disturbances exist, that is, $s = \dot{s} = 0$ (Lin & Chiang, 2013). This is illustrated in Figure 2.8.

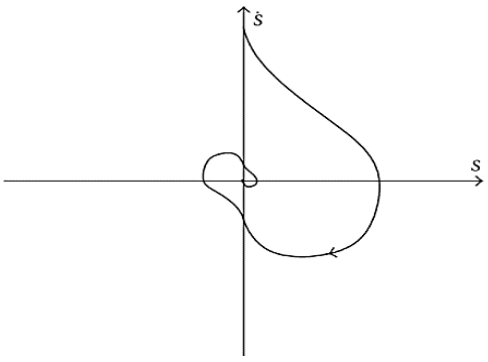


Figure 2.8: Phase plane trajectories of STA (Lin & Chiang, 2013)

Hence, the super-twisting sliding mode algorithm is one of the outstanding robust control algorithms which handles systems that are nonlinear and whose relative degree of u to s is one to avoid chattering (Chi et al., 2009; Lin & Chiang, 2013). The super-twisting algorithm and its closed-loop sliding dynamics can be designed as

$$\dot{s} = -\lambda_1 |s|^{1/2} sign(s) - \lambda_2 \int_0^t sign(s) d\tau + d(t),\ s = s(x,t) \in \mathbb{R} \quad \ldots (2.17)$$

where $d(t)$ represents an unknown time-varying disturbance of the form $d(t) \in \mathbb{R}$, and (λ_1, λ_2) are the gain pairs to be designed to achieve the sliding mode performance in finite time. This can be represented in state-space form as below (Li & Peng, 2021).

Taking $s = s(x,t) \in \mathbb{R}$,

$$\left.\begin{aligned} z_1 &= s(x,t) = s \\ z_2 &= -\lambda_2 \int_0^t sign(s) d\tau + d(t) \end{aligned}\right\} \quad \ldots (2.18)$$

this implies

$$\dot{z}_1 = \dot{s} = -\lambda_1 |s|^{1/2} sign(s) - \lambda_2 \int_0^t sign(s) d\tau + d(t),$$

Since $z_1 = s$, and $z_2 = -\lambda_2 \int_0^t sign(s) d\tau + d(t)$, Eq. (we) becomes

$$\left.\begin{aligned} \dot{z}_1 &= -\lambda_1 |z_1|^{1/2} sign(z_1) + z_2 \\ \dot{z}_2 &= -\lambda_2 sign(z_1) + \dot{d}(t) \end{aligned}\right\} \quad \ldots (2.19)$$

2.2.5 Stability analysis of the control

To achieve the asymptotic convergence of state variables in the presence of disturbance, the sliding variable, s, must be driven to zero in finite time by the control law u. To guarantee that this happens, a stability test/proof has to be proven. This is carried out using the Lyapunov function techniques to the s-dynamics.

Proof:

Let the Lyapunov function candidate be:

$$V = \frac{1}{2} s^2 \quad \ldots (2.20)$$

$$V = s = s(x1, x2) = x2 + cx1,\ \ c > 0$$

For the asymptotic convergence, the following conditions must be satisfied

i. Positive-definite

ii. $\lim_{s \to \infty} V = \infty$

iii. $\dot{V} \leq 0\ for\ asymptotic\ stability$

But for finite-time convergence, the condition (iii) is modified by $s\dot{s} = -\eta\, s\, sign(s)$ sliding condition

$$\dot{V} \leq -\alpha V^{\frac{1}{2}}, \;\; \alpha > 0 \quad \ldots (2.21)$$

$$\frac{dV}{dt} \leq -\alpha V^{\frac{1}{2}}$$

$$\Rightarrow \frac{1}{\alpha} V^{-\frac{1}{2}}\, dV \leq -dt$$

Integrating both sides,

$$\frac{1}{\alpha}\int_{V(0)}^{0} V^{-\frac{1}{2}}\, DV \leq -\int_{0}^{t_f} dt$$

$$t_f \leq \frac{2}{\alpha} V(0)^{\frac{1}{2}} \qquad \alpha > 0 \quad \ldots (2.22)$$

Where t_f is the reaching time, and α is a positive constant.

2.2.6 Concept of Reaction Wheels and DC Motors

By definition (The Economic Times, 2021), the reaction wheel is a type of flywheel employed in the attitude control of spacecraft. It is a kind of active control device also known as a zero-momentum device that can produce small torques necessary for proper pointing and stability of the spacecraft. Among the torque control devices used in a spacecraft, the reaction wheel is the most preferred for very accurate attitude control and also fast maneuverability with less parasitic disturbance torques (Sidi, 2014). The idea behind its ability to be used for very precise attitude control as stated by Bryson (2015) is that unwanted angular momentum of the spacecraft as a result of disturbances and uncertainties can be put into the wheels.

Essentially, they are electrically powered devices that are made up of a dc motor connected to a flywheel and designed to produce a balancing momentum that counters the effects of external disturbances resulting from the routine operations of the spacecraft. Yeh (2010) describes the reaction wheel as a rotating flywheel that is supported by ball bearings and controlled by an internally placed brushless dc motor and connected electronics. In design, reaction wheels are made up of a flywheel whose mass is considered to be concentrated at the rim, and they are arranged in an orthogonal setup of not less than about three reaction wheels – one each to cater for each axis ($x, y, and\ z$) – and a fourth redundant wheel. The diagrammatic representation of the reaction wheel is given in Figure 2.9.

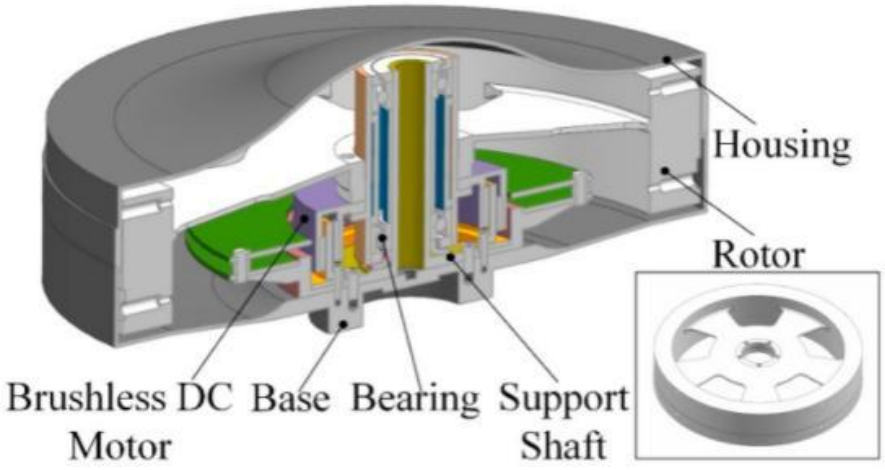


Figure 2.9: Section of a Reaction Wheel (Szeliski, 2010)

More so, to rotate the vehicle or spacecraft in a given direction, the wheel has to spin in the other direction. In importance, failure in one or more reaction wheels on a spacecraft can make it lose its capability to stay in position, thereby leading to the overall failure in its mission. But in recent times, research has gone into underactuated control systems that can still work even if the failure occurs (Bayat & Javaheri, 2020; Derbel et al., 2017; Ismail et al., 2020; Trentin et al., 2020).

2.2.7 System and Attitude dynamics

Considering Euler's moment equations, the spacecraft system dynamics can be generalized thus;

$$T = \dot{h} = \dot{h}_b + \omega * h \qquad \ldots (2.22)$$

Where T is the torque,

Since $h = J * \omega$,

The attitude dynamics exhibited by a spacecraft with a reaction wheel as an actuator can be mathematically represented as shown using the Euler moment equations (Sidi, 2014).

$$J\dot{\omega} + \omega \times h = T_w + T_{ext} \qquad \ldots (2.23)$$

where T_{ext} is the disturbance torque, T_w is the reaction wheel torque, and J is the inertia of the spacecraft, including the reaction wheel.

Thus, the angular rate can be expressed as

$$\dot{\omega} = \frac{1}{J}(-\omega \times h + T_w + T_{ext}) \qquad \ldots (2.24)$$

Likewise, Takehana and Uchiyama (Takehana & Uchiyama, 2017) as well demonstrated how a satellite (spacecraft) model contains the satellite itself, a motor, and a spherical rotor serving as the flywheel. So, the system dynamics of the spacecraft expressed in equation terms can be illustrated thus;

$$J_s \frac{d\omega_s}{dt} + \omega_s \times J_s \omega_s = T_s \qquad \ldots (2.25)$$

Where J_s represented the inertial matrix of the satellite, T_s and ω_s represent the general torque vector and angular velocity vector of the satellite.

Since the resolved angular velocity is dependent on the angular velocity of the satellite, motor, and the flywheel, that is;

$\omega_s \Rightarrow \omega_s + \omega_f$ and $\omega_s \Rightarrow \omega_s + \omega_m$

Hence, the equivalent equations for motor and flywheel dynamics become

$$T_m = J_m \frac{d(\omega_s + \omega_m)}{dt} + (\omega_s + \omega_m) \times J_m(\omega_s + \omega_m) \qquad \ldots (2.26)$$

And

$$T_f = J_f \frac{d(\omega_s + \omega_f)}{dt} + (\omega_s + \omega_f) \times J_f(\omega_s + \omega_f) \qquad \ldots (2.27)$$

Where T_f, T_m, are the torque vectors that are generated by the flywheel and the motor, respectively, ω_f, ω_m are the angular velocity of the flywheel and the motor, and J_f, J_m are the inertia matrix of the flywheel and the motor respectively.

For practical implementation, the slip ratio, according to Takehana and Uchiyama (Takehana & Uchiyama, 2017), experienced between the flywheel and motor is given by the relation

$$Slip\ ratio = \frac{estimated\ motor\ speed - flywheel\ speed}{estimated\ motor\ speed} \qquad \ldots (2.28)$$

Since, $speed = radius \times angular\ speed$, and taking motor and flywheel radius as $r\ and\ R$ respectively

Therefore,

$$Slip\ ratio, M_s = \frac{(r\omega_m - R\omega_f)}{r\omega_m} \qquad \ldots (2.29)$$

To illustrate the relationship between the angular velocities of the motor and the flywheel, the expression below is given

$$M = \frac{\omega_m}{\omega_f} \qquad \ldots (2.30)$$

Consequently, if $M = \begin{bmatrix} M_{fm} & 0 & 0 \\ 0 & M_{fm} & 0 \\ 0 & 0 & M_{fm} \end{bmatrix}$ ... (2.31)

The slip ratio is

$$M_s = 1 - \frac{R}{rM_{fm}} = \frac{rM_{Rm} - R}{rM_{fm}} \qquad \ldots (2.32)$$

When angular momentum is conserved, the Euler moment equation for the spacecraft can be written as

$$(J_f + J_m + J_s)\dot{\omega}_s = (J_m M - J_f)\dot{\omega}_f - \omega_s \times J_s\omega_s \quad \ldots (2.33)$$

2.2.8 Bond graph Language

The bond graph language of system representation is coined around the exchange of power and information (Banerjee, 2005). Literary, the word bond graph is a representation of bonds that are formed linking together ports like the "single port", "double port", and the "multi-port" (Banerjee, 1998; Filippo et al., 1991; Machado & Cunha, 2021). This linkage can be illustrated in the form of a graph, hence the name "bond graph". It is simply a diagram showing a physical dynamic system through elements and connections of the dynamic system not specific to any particular domain, that is, it is a unified way of representing systems either in electrical, mechanical, hydraulic, magnetic, thermal, amongst others not minding their domain (Machado & Cunha, 2021).

If properly done, this technique of system modelling obeys the first principle of conservation of energy in generating the graphical model without the need to formulate the systems equations (Borutzky, 2006). Thus, since energy cannot be created or destroyed but can flow from one form to another, the vertices of the bond graph signify system components or elements while the edges denote the power bonds. The diagram in Figure 2.10 illustrates the bond graph relations.

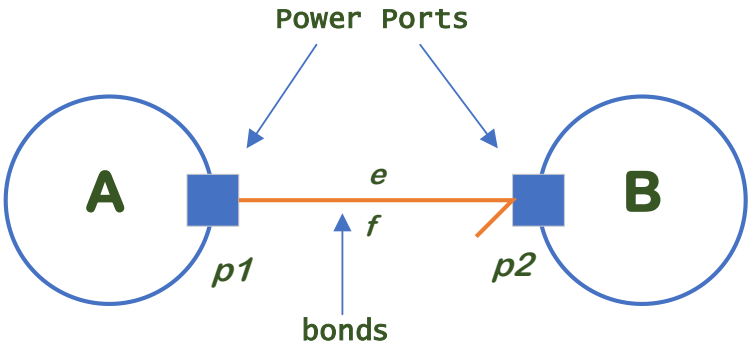


Figure 2.10: Bond and Ports connections (Borutzky, 2006)

In Addition, at each instance of time, the power is related to the effort and flow as;

$$Power = effort\ (e) \times flow\ (f) \qquad \ldots (2.34)$$

Also, the elements can be represented as shown in Table 2.2

Table 2.2: Elements and Junctions in Bond Graph (Banerjee, 2005)

Elements/Junctions	**Corresponding relations**
Inductive – I	$flow = (constant\ or\ a\ function) \times \int_{-\infty}^{present} effort\ dt$
Capacitive – C	$effort = (constant\ or\ a\ function) \times \int_{-\infty}^{present} flow\ dt$
Resistive – R	$effort = (constant\ or\ a\ function) \times flow$ $flow = (constant\ or\ a\ function) \times effort$
Source of Effort – SE	It controls the effort applied to the rest of the system, e.g., voltage source
Source of Flow – SF	It controls the flow into the rest of the system. E.g., current source
1 – Junction (series)	Common flow; summation of efforts
0 – Junction (parallel)	Common effort, summation of flows

Considering the inter-relationship between domains, Table 2.3 highlighted the effort, flow, momentum, and displacement as given by (Borutzky, 2006).

Table 2.3: Equivalent representation of power and energy variable in bond graph language (Borutzky, 2006)

Energy domain	**Effort (e)**	**Flow(f)**	**Generalized momentum (p)**	**Generalized displacement (q)**
Translational mechanics	Force F [N]	Velocity v [m/s]	Momentum p [Ns]	Displacement x [m]
Rotational mechanics	Angular moment M [Nm]	Angular velocity ω [rad/s]	Angular momentum p_ω [Nms]	Angle θ [rad]
Electro-magnetic	Voltage u [V]	Current i [A]	Linkage flux λ [Vs]	Charge q [As]
	Magnetomotive force V [A]	Magnetic flux rate $\dot{\Phi}$ [Wb/s]	–	Magnetic flux Φ [Wb]
Hydraulic	Total pressure, p [N/m^2]	Volume flow rate, Q $\left[\frac{m^3}{s}\right]$	Pressure momentum pp $\left[\frac{N}{m^2}s\right]$	Volume V_C [m^3]
Thermo-dynamic	Temperature T [K]	Entropy flow rate $\dot{S}$ [J/K/s]	–	Entropy S [J/K]
Chemical	Chemical potential μ [J/mole]	Molar flow $\dot{N}$ [mole/s]	–	Molar mass N [mole]

2.2.9 Reaction Wheel Modelling Using Bond Graph

The system is made up of a brushless dc motor and a flywheel interconnected, as illustrated in Figure 3.1. As voltage is inputted into the dc motor, it produces speed that rotates the flywheel,

which in turn generates angular momentum that is required for the stability of the entire system make-up

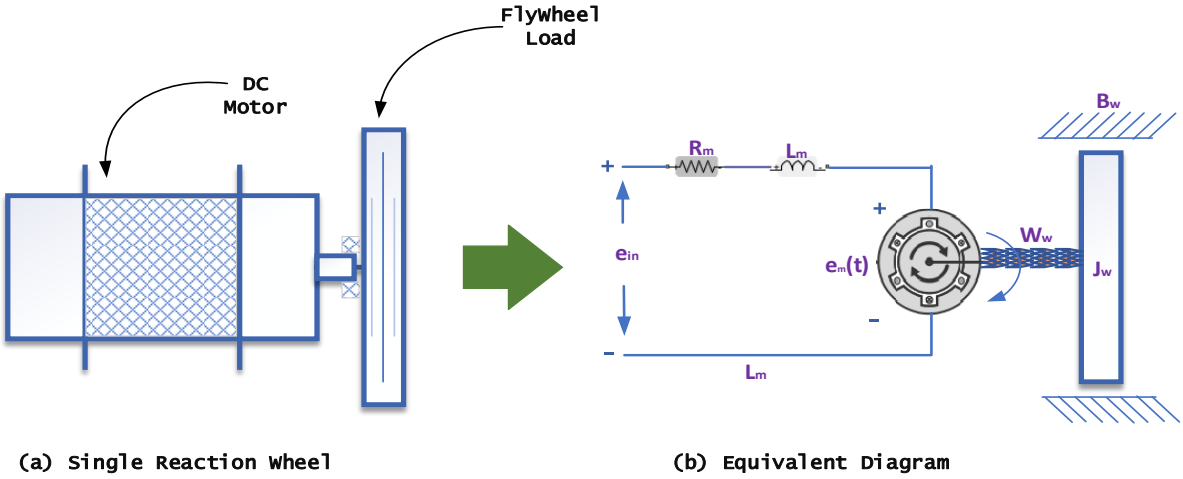


Figure 2.11: Reaction Wheel diagram

The System modelling will be done using Bond graph language as presented in (Kypuros, 2013). Below is the procedure carried out to achieve a good model for the dc motor attached with a flywheel (load) as analogous to the reaction wheel.

Procedures:

i. Diagrammatically represent each component in the engineering setup drawing as simple as possible. See Figure 2.11
ii. Reproduce the equivalent engineering representation of the setup, which will facilitate the identification of the distinct efforts and flow in the system
iii. Identify distinct efforts and establish a 0-junction for each
iv. Add 1-port elements and energy converting 2-ports for gyrators and transformers if available. In this case, a 2-port gyrator is implemented.
v. Allocate power directions. Most preferable, use the power-in-power-out convention as it is convenient in reducing eliminating redundant junctions
vi. Eliminate explicit ground, zero velocities, and all bonds directly attached via a 0-junction
vii. Reduce the bonds in the graph to the optimum by shrinking 2-port 0- and 1-junctions into single bonds and also adjacent junctions that are of the same type.
viii. Cause and effect relations were considered when allocating causal strokes to the bonds. These allocations were done such that the natural behaviour of elements is not affected.

$$flow = (a\ value)\ x \int effort\ dt \qquad \ldots (2.35)$$

Integral causality is most appreciable and applied.

ix. Label efforts and corresponding flows on the energy storing elements according to the causality chosen and recollect that linear:

- $C - elements \rightarrow displacement\ (q), flow\ (\dot{q}), effort\ \left({}^{q}/_{C}\right)$
- $I - elements \rightarrow momentum\ (p), effort\ (\dot{p}), flow\ \left({}^{p}/_{I}\right)$

C is compliance, while I is inertance

x. Apply the primary conditions that occur at each junction and propagate information throughout the graph to generate some differential equations.

xi. Lastly, for the final distribution of information that enhances the derivation of the complete differential equations that models the system, apply the secondary conditions. Figure 2.12 is the 20-sim representation of figure 2.11 in bond graph.

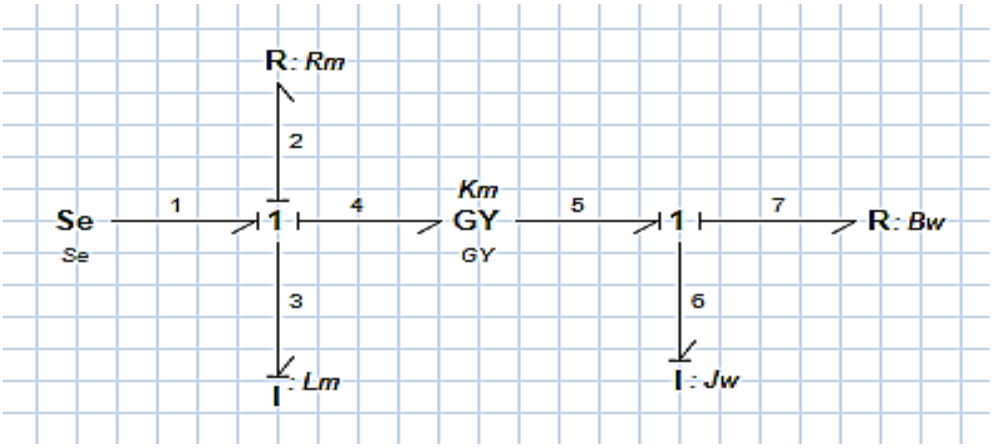


Figure 2.12: Bond graph model of the reaction wheel

<u>Derivations</u>

Taking $e_{in} = E$,

Since we have two (2) causal elements (energy-storing), it implies that the state variables will be two (2), that is:

State variables → $p_3,\ p_6$,

Source → $e_1 = E$

Storage elements → $f_3 = \frac{p_3}{L_m}$

$f_6 = \frac{p_6}{J_w}$

Common flows → $f_1 = f_2 = f_3 = f_4$

$$f_5 = f_6 = f_7$$

Common efforts → $no\ common\ effort, that\ is, no\ zero\ junction$

Other elements → $e_2 = R_m f_2 = R_m f_3 = R_m \cdot \frac{p_3}{L_m}$

$$e_7 = B_w f_7 = B_w f_6 = B_w \cdot \frac{p_6}{J_w}$$

Gyrator: → $e_4 = K_m f_5 = K_m f_6 = K_m \cdot \frac{p_6}{J_w}$

$$e_5 = K_m f_4 = K_m f_3 = K_m \cdot \frac{p_3}{L_m}$$

Using the arrow direction:

- First 1-junction

  $e_1 = e_2 + e_3 + e_4$

  $\Rightarrow\ e_3 = \dot{P}_3 = e_3 = e_1 - e_2 - e_4$

  $Where$

  $e_1 = E,\ e_2 = \frac{R_m}{L_m} p_3,\ e_4 = \frac{K_m}{J_w} p_6$

  Hence,

  $e_3 = \dot{P}_3 = e_3 = e_1 - e_2 - e_4 = E - \frac{R_m}{L_m} p_3 - \frac{K_m}{J_w} p_6$

  $\dot{P}_3 = E - \frac{R_m}{L_m} p_3 - \frac{K_m}{J_w} p_6$ ... (2.36)

- Second 1-junction

  $e_5 = e_6 + e_7$

  $\Rightarrow\ e_6 = e_5 - e_7$

  $where$

  $e_5 = \frac{K_m}{L_m} p_3,\ e_7 = \frac{B_w}{J_w} p_6$

  $e_6 = \dot{P}_6 = e_6 = e_5 - e_7 = e_5 - e_7 = \frac{K_m}{L_m} p_3 - \frac{B_w}{J_w} p_6$

  $\dot{P}_6 = e_6 = e_5 - e_7 = e_5 - e_7 = \frac{K_m}{L_m} p_3 - \frac{B_w}{J_w} p_6$ ... (2.37)

Since there is no 0-junction, and any other junction left, using the Eq. (2.36) and (2.37), the equation of states can be re-written as:

$$\dot{P}_3 = -\frac{R_m}{L_m} p_3 - \frac{K_m}{J_w} p_6 + E \quad \text{... (2.38)}$$

$$\dot{P}_6 = \frac{K_m}{L_m} p_3 - \frac{B_w}{J_w} p_6 \quad \text{... (2.39)}$$

In State Space Form:

$$\left.\begin{bmatrix}\dot{P}_3\\ \dot{P}_6\end{bmatrix} = \begin{bmatrix}-\frac{R_m}{L_m} & -\frac{K_m}{J_w}\\ \frac{K_m}{L_m} & -\frac{B_w}{J_w}\end{bmatrix}\begin{bmatrix}p_3\\ p_6\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}E \right\} \quad \ldots (2.40)$$

2.2.10 Shallow Neural Network

According to (J. Chen & Boyle, 2020), a series of algorithms that operates similarly to the human brain in identifying the fundamental relationships inherent in a set of data is referred to as the neural network. It is an artificial replication of the action of the human organic neuron in an artificial sense or way. A neural network can be classified as shallow and deep based on the number of hidden layers involved. A shallow neural network is made up of only one or two hidden layers (Agrawal, 2019), and it is the basis for understanding the operations of the deep neural network. The deep neural network has more hidden layers than the shallow neural network. An example of the shallow neural network is illustrated in Figure 2.13.

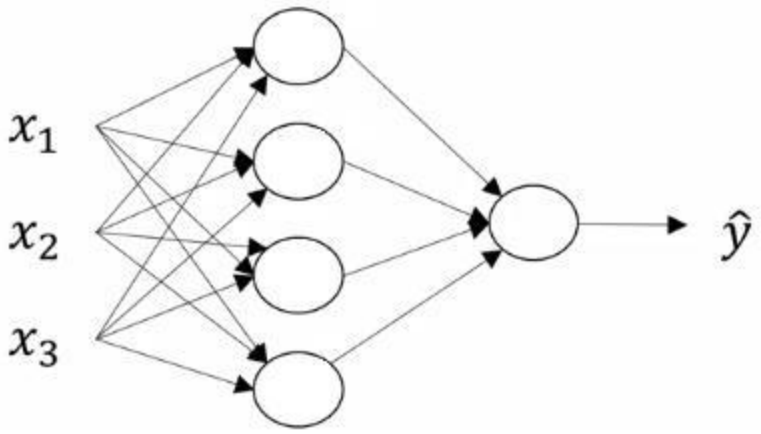


Figure 2.13: Shallow Neural Network (Agrawal, 2019)

The fundamental element or unit of the neural network is the neuron. This neuron serves as a conduit for input sent to it; that is, when input is sent to the neuron, it passes it as an output to the input of the succeeding layer by using weights and biases and performs activation via a selected activation function before passing the output (Mathworks Inc, 2018). These artificial neurons are arranged in such a way to mimic the human brain. A key item is weights and biases. Consequently, layers are arrangements of neurons with the required activation function and working together to achieve the required output. Below is the mathematical representation of

the neuron model or operations of the neuron. These equations are also referred to as the forward-propagation equations.

$$Z_j^{[i]} = w_j^{[i]} x + b_j^{[i]} \quad \ldots (2.41)$$

$$a_j^{[i]} = \sigma\left(Z_j^{[i]}\right) \quad \ldots (2.42)$$

Where $i\ and\ j$ denotes the layer and neuron number respectively, x is the input vector, $w\ and\ b$ are the weight and bias corresponding to the neuron j in the layer $i, Z\ and\ A$ are the intermediate and final output corresponding to the neuron j in the layer i, and finally, σ stand for the activation function which can be represented mathematically as

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \ldots (2.43)$$

This activation function makes the neural network account for nonlinearity as they make the network robust to different scenarios. In neural network applications, many activation functions are being in use. Some of the notable ones are the ReLU, Leaky ReLU, Tanh, and the sigmoid functions. Furthermore, for proper utilization of the neural network, the weights and biases need to be initialized and updated correctly (Agrawal, 2019). Weights are not just initialized by zero; the best method employed in the initialization of weights is Xavier's Initialization which states that a normal distribution with mean equal to zero and the variance a multiplicative inverse of the number of neurons of the previous layer. In the case of bias, it can be set to zero as initialization. At the same time, both the weights and the biases are updated through the use of the process called backpropagation, which computes the derivatives of the inputs present at each step.

According to (Mathworks, 2017), Neural networks are generally employed in financial operations and forecasting, enterprise planning, fraud detection, and risk management, amongst others. These applications are possible due to the basic functions of the neural network. The four basic functions are:

- Function fitting
- Pattern recognition and classification
- Clustering, Feature extraction, and Data dimension reduction
- Time series prediction, forecasting, and dynamic modelling

With each of these functions particularly suitable for each application, the neural network has been implemented in a variety of fields. For the function fitting, it entails training a network on a set of inputs with the intention of producing an associated set of target outputs. Once the network fits the data, it creates a generalization of the input-output relationship, which can be used to generate the required outputs for even the inputs that it was not trained on. A poorly trained network can lead to overfitting or underfitting, as indicated in Figure 2.14.

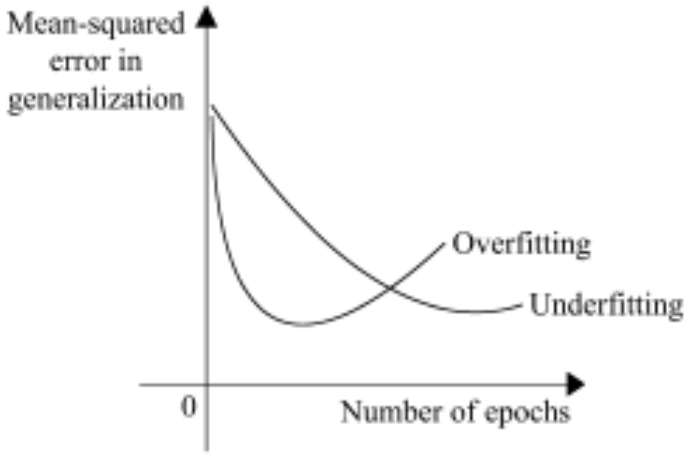


Figure 2.14: Validation Illustration (Mathworks Inc, 2018)

2.3 Review of Related works

Related works of literature are discussed in this section. It is divided into four parts for proper review and presentation. The first part focus on the related works in sliding mode control as a control strategy, which is then followed by works that use this SMC in relation to space technology. After which, it is narrowed down to the application of the SMC to dc motor, and then, works that use neural network in conjunction with the SMC to control dc motor, which is related to the makeup of the reaction wheel.

2.3.1 Sliding mode control

In real-world applications, uncertainty is common, and controlling these uncertainties will require a more robust control that will achieve the stability of the controlled systems. One of the well-known robust control strategies is the sliding mode control strategy. As stated by Jianxing Liu et al. (2020), this strategy is mainly inspired by the switching activities of electromagnetic relays, which are intended to keep or control the system states along a specified and desired path. For the concepts or principles governing the SMC, (Emel'yanov, 2007; Hung et al., 1993) and (Sabanovic et al., 2011) gave a suitable explanation of them. The sliding mode control is one technique that had been well implemented by control engineers, with multiple

research and works, due to its advantages (Li & Peng, 2021). Although Panchade et al. (2007) stated the early point when sliding mode control became known, the works of (Ahmed et al. 2019) and (Sowjanya & Tarakalyani, 2013) categorically explained how sliding mode control originated from the soviet union.

As an improvement to current trends as of then, Yan & Liu (2010) proposed a new strategy that will help in improving the rejection to disturbance capability and also the dynamic response of the system. By considering the reaching term and the equivalent control, they were able to develop an adaptive fuzzy system that is claimed to overcome the impact of the delay experienced in the network control system.

2.3.2 Spacecraft attitude tracking with SMC

Kojima & Trivailo (2019) proposed an underactuated adaptive SMC that is intended to use time delay in stabilizing the attitude of a bias-momentum satellite through two installed reaction wheels. The idea was to use two control torques to achieve stabilization since underactuated systems can be controllable but asymptotically un-stabilized through continuous state feedback since it does not satisfy Brockett's conditions. One major drawback of their work is that it is not valid in the presence of external disturbances and modelling uncertainties, as seen in satellite moments of inertia. Likewise, Yeh (2010) used two non-linear controllers for attitude tracking. The first utilized sliding mode in attitude tracking, while the second used the adaptive sliding mode attitude tracking. Both controllers work simultaneously to effectively estimate the parameters and eliminate disturbances caused by the attitude control system. In a further application of the adaptive SMC control, (Pukdeboon et al., 2010; Tiwari et al., 2018) introduced a higher-order SMC approach to solving the unwinding problem inherent when considering the Quaternion based attitude tracking.

Using a sliding mode friction observer, Peixi Zhang et al. (2017) estimated the friction torque for the feed-forward compensation in reaction wheel in the inner layer, a flexible appendage vibration observer to estimate the vibrations in the middle layer and a H_∞ controller at the outer layer. These three forms a composite scheme aimed at providing an anti-disturbance control or disturbance rejection and robust performance for the spacecraft using the reaction wheel and also enhances its pointing accuracy. In all these, they were faced with the challenge of having to account for the effect of saturation occurring in the reaction wheel. This actuator saturation limitations and the presence of singularities led to the use of a novel and uncommon control configuration which entails the use of two reaction wheels as discussed earlier (Trentin et al., 2020); (Trentin et al., 2020) improved on the issue by proposing the combination of a sliding

mode controller and a nonlinear proportional derivative (PI) controller applied in the control of two reaction wheels pendulum system.

While stabilizing the attitude of a fully actuated zero-momentum satellite, Bayat & Javaheri (2020) focused on orbits that are near-polar and exposed to various uncertainties and unknown disturbances. To achieve global stability and solve the robust attitude tracking problem, they utilized a terminal sliding mode control (TSMC) technique with the controller in a 2-level architecture, that is, finite-time detumbling and tracking of the expected angular velocities. Another notable work on the application of terminal sliding mode attitude control to spacecraft is that of (Qiao et al., 2020). In this case, Qiao et al. (2020) focused on a novel stabilization scheme based on dual disturbances observers and high precision nonsingular terminal sliding mode control to achieve spacecraft attitude stabilization accuracy and convergence rate.

By using a hyperbolic function, Ismail et al. (2020) applied the conventional SMC to the attitude control of satellites with reaction wheels aiming to reduce the chattering problem and also to control the attitude of the overall satellite. The employed control technique is hinged around an underactuated system expected to have two functional reaction wheels, and the applied sliding mode controller is a fractional-order control. A related application of sliding mode control to reaction wheel systems is the work done by (Pengfei Zhang et al., 2020) in stabilizing the attitude instability caused by robotic fish reciprocating movements along the roll axis. They proposed a hybrid controller that combines the sliding mode controller with a neural network compensator that is designed to reject the severe disturbance on the roll angle. One of the challenges of this work is energy consumption which is relatively higher than the passive methods.

2.3.3 Application of SMC to dc motor

Works on the application of sliding mode controls to dc motor are fundamentally classified into position control, speed control, and torque control. Takehana & Uchiyama (2017) designed a controller for a small satellite reaction wheel process which is a spherical rotor using a simple driving principle that applies mini dc motors as actuators with the sole aim of reducing the entire size of the reaction wheel system, which will consequently reduce the size of the entire spacecraft. This system, though effective, encountered three major issues, which are the stick-slip phenomenon, friction wear of the system, and the dynamics of the actuator.

On implementing active control on CubeSats, specifically in the area of attitude maneuvering, Takehana & Uchiyama (2017) also compared the application of reaction wheels for different

sizes of CubeSats to prove their performance as attitude maneuvers. Other reviews on controlling dc motor indicate the output of choice is position, speed, or torque. Working on separately excited dc motor, MaheswararaoCh. et al. (2011) used the sliding mode controller in regulating the speed with the aim of alleviating the peak overshoot, settling time and the steady-state error encountered using conventional PI controllers. Comparing the SMC output with the PI gave an acceptable result. Also, focusing on speed control is the work of (Ambesange et al., 2013; DilKumar & Mija, 2015) which indicated the suitability and impressiveness of the outcomes of the SMC over the conventional PID controllers in application in dc motor control.

From the angle of torque, Mehmood et al. (2020) considered mitigating the effects of environmental conditions and how it affects the overall performance of the brushless dc motor. They suggested that multiple techniques of control had been proposed that lag in the provision of a complete and robust resolution of the imminent issue faced by real-life speed control of dc motor, and as such, their proposed technique will offer such a solution. One major highlight of their proposed work was their view on not needing sensors for current measurement due to the inclusion of the hysteresis current controller. Also, Pillay & Krishnan (1989) used a phase variable model of the DC motor to test the outcome of the speed servo drive with hysteresis and PWM current controllers. In the end, they got similar small and large signal responses that are the same irrespective of the input chosen, either the hysteresis current controller or the PWM current controller.

2.3.4 Neural Network Application to Sliding Mode Control and DC Motor Control

Basic neural networks have been in use with the different variants of the SMC for some time now and even in many commercial and industrial applications (Truong et al., 2019). Works like (H. Chen & Wo, 2013; Hu et al., 2017; R. Liu et al., 2019; Parma et al., 1999; Razmi & Afshinfar, 2019; Truong et al., 2019; Wang et al., 2020) uses the neural network in improving the outcome of the sliding mode control employed. (H. Chen & Wo, 2013) adopted the radial basis function in conjunction with the time-varying SMC to prove that the close loop is uniformly ultimately bounded, as such, providing an overall procedure that accounts for nonlinearities experienced in the control of manipulators. Similarly, Vo & Kang (2018) used the radial basis function, a nonsingular fast terminal sliding function, to significantly reduce the issues caused by disturbances and uncertainties from the dynamic system. In this case, the radial basis function was particularly used to estimate the dynamic model of the system. Likewise, Vo & Kang (2018) presented similar work but focused on attaining a faster convergence and advanced precision control in the piezo positioning stage.

Furthermore, Hu et al. (2017) used a neural net to provide continuous control, which replaces the high switching discontinuous control of the SMC that causes system degradation. Hence, reducing the tracking error and eliminating chattering that can cause the vibration of the folding-boom aerial work platform. R. Liu et al. (2019) proposed a particular neural network SMC meant for backlash-like hysteresis nonlinear systems. They utilized one neural net to estimate the states of the system that are unknown and employed the other neural network SMC for nonlinearities caused by hysteresis. To cope with the effect of uncertainties and improve the characteristics of the robotic manipulator, with the aim of achieving higher accuracy of tracking, Truong et al. (2019) proposed an SMC neural network that can be utilized for both position-tracking control and industrial manipulators with 3-DOF. Using Recurrent Elman Neural Network to tune the gain pairs of the super-twisting sliding mode control, (Suleiman et al., 2018) presented a smoothed super-twisting sliding mode control applicable to the ball and plate system and not the brushless dc motor setup.

2.4 Research Gap and Proposed Work

Based on the works reviewed thus far, it can be deduced that one of the major issues arising from the use of sliding mode controllers is the chattering phenomenon which can cause intolerable energy losses and high wear and tear of the movable parts in the actuator (Suleiman et al., 2018). This chattering effect impacts negatively on the actuators in active use, which might even cause internal disturbances to be generated. More so, present attempts to reduce this chattering – even though encouraging – have their limitations. Some of these notable works in the reduction of chattering and driving the state to asymptotic stability in finite time can be accorded to the use of higher-order sliding mode control, like the super twisting SMC. However, the gain pairs require some level of trial and error tuning to achieve an acceptable performance from the controller. Hence, this work is set to achieve adaptability to changes and robust handling of disturbances through an automatic tuning process, thereby eliminating the need for a trial-and-error method of gain tuning of the super twisting SMC. Figure 3.5 is a representation of the proposed model.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

This chapter elaborates on the methods and procedures implemented in this study. It covers the design of the super-twisting sliding mode control and the neural network employed in tuning the two major gain pairs to attain the asymptotic stability faster. Furthermore, the instruments employed and the step-by-step procedure implemented in line with the research objectives highlighted earlier are presented as enumerated. Also, it includes process modelling and performance evaluation.

3.2 Procedures for Achieving the Objectives

Following the objectives of this study, the procedures engaged in achieving it are further described in the following subsequent sections in the order they were carried out or undertaken.

3.2.1 Study of existing literature

Existing works of literature and texts related to the sliding mode control and its applications were collected and studied to provide a better and more robust method for controlling brushless DC motors. As well, the sliding manifold to be used is identified, with the intended neural network to be employed. These were accomplished with the use of the following steps:

i. Collect relevant materials on the subject matter
ii. Study present application of both first order and other higher-order sliding mode controller design
iii. Identify the specific controller and reaction wheel configuration to be used. Compare the efficacy to other related projects
iv. Write a very comprehensive review of findings

3.2.2 System modelling

From equation (2.40) in section (2.2.10), the state space form is converted to its electrical and rotational mechanical form/equivalent using the following relationship as applied:

Load torque, $\tau_w = \dot{P}_6$ ... (3.1)

$p = J\omega$, $p_6 = J_w \times \omega_w$ ... (3.2)

$\dot{P}_6 = J_w \times \dot{\omega}_w$, $\implies \frac{d(\omega_w)}{dt} = \dot{\omega}_w = \frac{\dot{P}_6}{J_w}$ ... (3.3)

$$\Phi_{(i)} = L_m \times i = p_3 \qquad \ldots (3.4)$$

$$\dot{\Phi}_{(i)} = L_m \times \frac{d(i)}{dt} = \dot{P}_3 \qquad \ldots (3.5)$$

$$\frac{\dot{\Phi}_{(i)}}{L_m} = \frac{d(i)}{dt} = \frac{\dot{P}_3}{L_m} \qquad \ldots (3.6)$$

Thus, Eq. (3.4) and (3.5) become

$$\Longrightarrow \frac{\dot{\Phi}_{(i)}}{L_m} = \frac{d(i)}{dt} = \frac{\dot{P}_3}{L_m} = \frac{1}{L_m}\left(-\frac{R_m}{L_m}(L_m \times i) - \frac{K_m}{J_w}(J_w \times \omega_w) + E\right)$$

$Therefore,$

$$\frac{d(i)}{dt} = \dot{i} = \frac{\dot{P}_3}{L_m} = -\frac{R_m}{L_m} i - \frac{K_m}{L_m}\omega_w + \frac{E}{L_m} \qquad \ldots (3.7)$$

$$\Longrightarrow \quad \frac{d(\omega_w)}{dt} = \dot{\omega}_w = \frac{\dot{P}_6}{J_w} = \frac{1}{J_w}\left(\frac{K_m}{L_m}(L_m \times i) - \frac{B_w}{J_w}(J_w \times \omega_w)\right)$$

$Therefore,$

$$\frac{d(\omega_w)}{dt} = \dot{\omega_w} = \frac{K_m}{J_w} i - \frac{B_w}{J_w}\omega_w \qquad \ldots (3.8)$$

In State Space Form, Eq. (3.7) and (3.8) can be represented thus:

$$\left.\begin{bmatrix} i \\ \dot{\omega}_w \end{bmatrix} = \begin{bmatrix} -\frac{R_m}{L_m} & -\frac{K_m}{L_m} \\ \frac{K_m}{J_w} & -\frac{B_w}{J_w} \end{bmatrix}\begin{bmatrix} i \\ \omega_w \end{bmatrix} + \begin{bmatrix} 1/L_m \\ 0 \end{bmatrix} E \qquad \right\} \qquad \ldots (3.9)$$

$$\dot{\boldsymbol{x}} = \boldsymbol{A}x + \boldsymbol{B}U$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} i \\ \dot{\omega}_w \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} -\frac{R_m}{L_m} & -\frac{K_m}{L_m} \\ \frac{K_t}{w} & -\frac{B_w}{J_w} \end{bmatrix}, x = \begin{bmatrix} i \\ \omega_w \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 1/L_m \\ 0 \end{bmatrix}, \; U \; = \; E$$

Resolving to get the value of the angular speed (ω)

$$sI - A = [s] \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -\frac{R_m}{L_m} & -\frac{K_m}{L_m} \\ \frac{K_t}{J_w} & -\frac{B_w}{J_w} \end{bmatrix} = \begin{bmatrix} s + \frac{R_m}{L_m} & \frac{K_m}{L_m} \\ -\frac{K_t}{J_w} & s + \frac{B_w}{J_w} \end{bmatrix}$$

Determinant of $sI - A$, $\triangle = |sI - A| = \begin{vmatrix} s + \frac{R_m}{L_m} & \frac{K_m}{L_m} \\ -\frac{K_t}{J_w} & s + \frac{B_w}{J_w} \end{vmatrix} = \left(\left(s + \frac{R_m}{L_m}\right)\left(s + \frac{B_w}{J_w}\right) + \frac{K_m}{L_m} \cdot \frac{K_t}{J_w}\right)$

$$\triangle \; = \left(\left(s + \frac{R_m}{L_m}\right)\left(s + \frac{B_w}{J_w}\right) + \frac{K_m K_t}{J_w L_m}\right)$$

From Cramer's Rule (Stapel, 2021), the transfer function $\frac{\omega(s)}{U(s)}$ can be gotten as such;

Replacing the $\omega(s)$ column of the $(SI - A)$matrix with the **B** matrix yields

$$\begin{bmatrix} s + \frac{R_m}{L_m} & \frac{1}{L_m} \\ -\frac{K_t}{J_w} & 0 \end{bmatrix}$$

Hence $\triangle_1 = \begin{vmatrix} s + \frac{R_m}{L_m} & \frac{1}{L_m} \\ -\frac{K_t}{J_w} & 0 \end{vmatrix} = \frac{K_t}{J_m} \cdot \frac{1}{L_m} = \frac{K_t}{J_m L_m}$

Therefore,

$$\frac{\omega(s)}{U(s)} = \frac{\triangle_1}{\triangle} = \frac{\frac{K_t}{J_w L_m}}{\left(s + \frac{R_m}{L_m}\right)\left(s + \frac{B_w}{J_w}\right) + \frac{K_m K_t}{J_w L_m}} \quad \ldots (3.10)$$

Let $a_1 = \frac{K_t}{J_w L_m}$, $a_2 = \frac{R_m}{L_m}, a_3 = \frac{B_w}{J_w}, a_4 = \frac{K_m K_t}{J_w L_m}$

So, the transfer function can be rewritten as,

$$\frac{\omega(s)}{U(s)} = \frac{a_1}{(s + a_2)(s + a_3) + a_4} = \frac{a_1}{s^2 + (a_2 + a_3)s + a_2 a_3 + a_4} \quad \ldots (3.11)$$

Thus,

$$\omega(s)[s^2 + (a_2 + a_3)s + a_2 a_3 + a_4] = U(s)a_1$$

$$s^2\omega(s) + (a_2 + a_3)s\omega(s) + (a_2 a_3 + a_4)\omega(s) = U(s)a_1$$

Transforming from Laplace form to time derivative will give

$$\ddot{\omega} + (a_2 + a_3)\dot{\omega} + (a_2 a_3 + a_4)\omega = u \cdot a_1$$

$$\ddot{\omega} = -(a_2 + a_3)\dot{\omega} - (a_2 a_3 + a_4)\omega + u \cdot a_1 \quad \ldots (3.12)$$

Table 3.1 contains the system parameter used in this work and for simulations, both for the open-loop system and the closed-loop control system design. The parameters are obtained from the work of (Rakhonde & Kulkarni, 2018) to aid the performance evaluation.

Table 3.1: Model Parameters (Rakhonde & Kulkarni, 2018)

Parameter	Value	Description
L_m	0.16273	Motor inductance (H)
R_m	7.72	Motor resistance (Ω)
e_{in}	E	Motor input voltage ($volts$)
J_w	0.0236	Flywheel/load moment of inertia ($Kg \cdot m^2/s$)
B_w	5.1	Flywheel/load-bearing [rotational resistance] - (Nms)
K_t	1.25	Torque constant (Nm/Amp)
T	T	Load/Flywheel torque (Nm)
K_m	0.144	Motor constant/back-emf constant (Vs/rad)
ω	x_1	Angular Velocity (Rad/s)

3.2.3 Sliding Surface Selection

For a speed control situation;

Let $x_1 = \omega(t) = \omega_w$

$x_2 = \dot{x}_1 = \dot{\omega}(t) = \dot{\omega}_w$ ... (3.13)

$\dot{x}_2 = \ddot{x}_1 = \ddot{\omega}(t) = \dot{\omega}_w = -(a_2 + a_3)\dot{\omega}_w - (a_2 a_3 + a_4)\omega_w + u \cdot a_1$ ... (3.14)

For speed control, the required output is angular speed, ω_w, hence, $y = \omega_w = x_1$

Thus, the state variable from Eq. (3.12) and (3.13) can be stated as:

$\dot{x}_1 = x_2$

$\dot{x}_2 = -(a_2 + a_3)x_2 - (a_2 a_3 + a_4)x_1 + u \cdot a_1$

$y = x_1$ (output angular speed of interest)

From the first principle, the general formula for sliding variable according to Eq. (2.12) is applied

$$s = \left(\frac{d}{dt} + c\right)^{k-1} e \quad \text{... (3.15)}$$

Where k is the order of the system, c is an arbitrary constant of the sliding manifold, and e is the error function.

Since the angular speed as seen is a second-order system (11), $k = 2$. Hence, the sliding variable is given as

$$s = \left(\frac{d}{dt} + c\right)^{2-1} e = \frac{de}{dt} + ce \Rightarrow s = \dot{e} + ce \quad \text{... (3.16)}$$

Where $e = \omega_{ref} - \omega_w$ (speed control)

3.2.4 Design of control law

For this system (Reaction wheel), the design of the control law is carried out progressively, in the order given below

- Conventional SMC law
- Second-order SMC law (Super Twisting)
- Adaptive neural tuned super twisting SMC control law

First, the system control is carried out using the first-order control law, then with a similar process, the super twisting law is implemented, after which neural network is used on the nominal part of the control law to adaptively tune the proportional and integral gains (gain pairs) in achieving the reaching law.

3.2.4.1 Conventional Sliding Mode control law design

The sliding surface is thus represented using the nominal control law as:

$$\dot{s} = -\lambda sign(s)$$

Therefore, applying the selected sliding variable as indicated in Eq. (3.13),

$$s = \dot{e} + ce = \frac{d(\omega_{ref} - \omega_w)}{dt} + c(\omega_{ref} - \omega_w) = (\dot{\omega}_{ref} - \dot{\omega}_w) + c(\omega_{ref} - \omega_w) \quad \text{... (3.17)}$$

Taking the time derivative of Eq. (3.14)

$$\left.\begin{aligned} \dot{s} &= (\ddot{\omega}_{ref} - \ddot{\omega}_w) + c(\dot{\omega}_{ref} - \dot{\omega}_w) \\ \dot{s} &= \ddot{\omega}_{ref} + c\dot{\omega}_{ref} - \ddot{\omega}_w - c\dot{\omega}_w = 0 \\ \dot{s} &= -\ddot{\omega}_w - c\dot{\omega}_w = 0 \end{aligned}\right\} \quad \text{... (3.18)}$$

Where $\dot{s} = -\lambda sign(s)$

Consequently,

$\dot{s} = \ddot{\omega}_{ref} + c\dot{\omega}_{ref} - \ddot{\omega}_w - c\dot{\omega}_w = -\lambda sign(s)$

$\ddot{\omega}_{ref} + c\dot{\omega}_{ref} - \ddot{\omega}_w - c\dot{\omega}_w = -\lambda sign(s)$

Substituting the value for $\ddot{\omega}_w = \dot{x}_2 = -(a_2 + a_3)x_2 - (a_2a_3 + a_4)x_1 + u \cdot a_1$ and taking $\ddot{\omega}_{ref}$ as $\dot{x}_{2d}$ and $\dot{\omega}_{ref}$ as x_{2d}

$\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - u \cdot a_1 - cx_2 = -\lambda sign(s)$

$u \cdot a_1 = \dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda sign(s)$

Hence,

$$u_{smc} = \frac{1}{a_1}[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda sign(s)] \quad \text{... (3.19)}$$

Recall, $a_1 = \frac{K_t}{J_w L_m} = 325.5$, $a_2 = \frac{R_m}{L_m} = 47.4, a_3 = \frac{B_w}{J_w} = 0.127, a_4 = \frac{K_m K_t}{J_w L_m} = 46.7$

This implies that the first other sliding mode control input/law is

$$u_{smc} = \frac{1}{325.5}[\dot{x}_{2d} + cx_{2d} + (47.5 - c)x_2 + (46.7)x_1 + \lambda sign(s)] \quad \ldots (3.20)$$

Where the constant of the sliding matrix c given as $C_s \in R^{m\times n}$, $c < 0$

To reduce the chattering effect caused by the signum function, a saturation function can be introduced;

$\lambda sign(s) \approx \lambda sat(s) = \lambda \frac{s}{|s|+\delta}$, Where $0 < \delta < 1$

Accordingly,

$$u_{smc} = \frac{1}{325.5}\left[(47.5 - c)x_2 + (46.7)x_1 + \lambda \frac{s}{|s|+\delta}\right], \quad \ldots (3.21)$$

Although the saturation function reduces chattering, the disadvantage is that its impact on the robustness of the system is much.

The Simulink implementation of the entire system using the conventional sliding mode control is illustrated in Appendix C.

3.2.4.2 Second-order control law (super twisting) design

For the second-order super twisting sliding mode control (Dai et al., 2019; Damiano et al., 2004; Derbel et al., 2017; Li & Peng, 2021; Lin & Chiang, 2013; Perruquetti & Barbot, 2002), the nominal part of the control is expressed as given in Eq. (2.18). This nominal control drives the sliding manifold to zero, hence differentiates it from zero.

$$U_n = U_{n1} + U_{n2} \quad \ldots (3.22)$$

Where, $U_{n1} = -\lambda_1|s|^{1/2}sign(s)$ and $U_{n2} = -\lambda_2 \int_0^t sign(s)d\tau + d(t)$,

Recall that Eq. (2.18) stated that

$$\dot{s} = -\lambda_1|s|^{1/2}sign(s) - \lambda_2 \int_0^t sign(s)d\tau + d(t),\ s = s(x,t) \in \mathbb{R} \quad \ldots (3.23)$$

(λ_1, λ_2) are the gain constant pairs that are to be designed to achieve the sliding mode performance in finite time. One representation of the relationship between the two constant gain pairs is given by (Chacón et al., 2018) to be:

$$\left.\begin{array}{l} \lambda_1 = \sqrt{W} \\ \lambda_2 = 1.1W \end{array}\right\} \quad \ldots (3.24)$$

$\Rightarrow {\lambda_1}^2 = 0.91\lambda_2$

Consequently,

$$u_{smc} = \frac{1}{325.5}\begin{bmatrix} \dot{x}_{2d} + cx_{2d} + (47.5 - c)x_2 + (46.7)x_1 - \lambda_1 |s|^{1/2} sign(s) \\ -\lambda_2 \int_0^t sign(s)d\tau + d(t), \end{bmatrix} \quad \ldots (3.25)$$

where $d(t)$ is the disturbance

So, using the trial-and-error method of tuning, the value of W is taking, which will yield corresponding good performance in the system control. For this project, it W is set as 50,000, since the value yielded a better result, that is, $W = 50{,}000$

The simulation of this system with the super twisting sliding mode control is also illustrated in Appendix C.

3.2.5 Neural Network design and implementation

The proposed adaptive neural network employed in this work for the tuning of the gain pairs is as illustrated in Figure 3.1. It is a two-layer network comprising the hidden layer and the output layer. As well, it has two input nodes that are fed directly to the thirty-two neurons that make up the hidden layer. These neurons are connected to the output layer, which in turn, are linked to the output node (one).

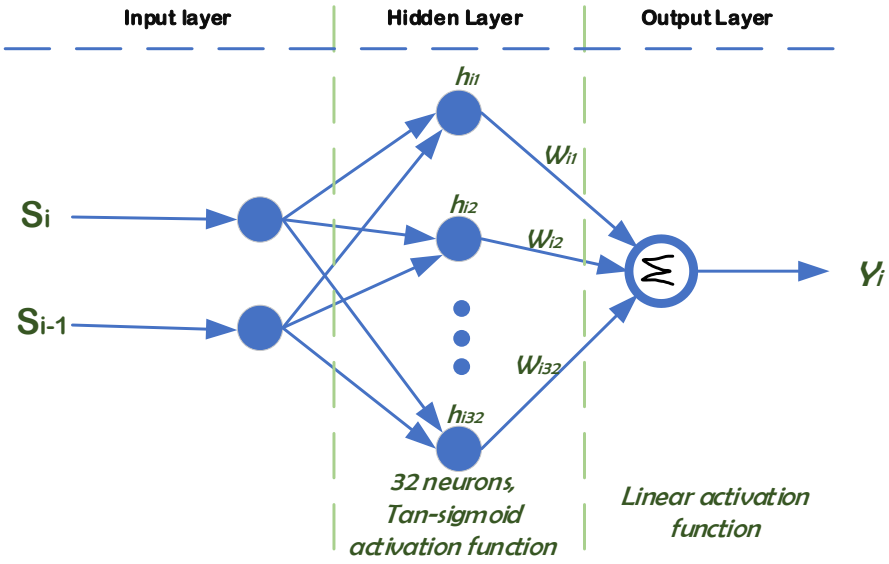


Figure 3.1: Architecture of the Neural Network

The activation function used for the hidden layer is the tan-sigmoid transfer function denoted below

$$f(x) = tansig(x) = \frac{2}{1 + \exp(-2x)} - 1 \quad \ldots (3.26)$$

Whereas the output layer is configured to use a linear activation function, as also indicated below

$$f(x) = x \qquad \ldots (3.27)$$

In designing the neural network that is proposed, the following procedures are followed:

i. The second-order sliding mode control (super twisting) is designed and simulated in Simulink using the appropriate time interval as indicated in appendix C.

ii. Then, the input to the nominal control, which is the sliding variable (s) and the output is tapped out and sent to the workspace, including a unit delay introduced to the input, that is, s_{i-1}. This s_{i-1} will serve as the previous data to be used in the back-propagation algorithm in training the network to predict future possible target values.

iii. From MATLAB workspace, the data corresponding to s and s_{i-1} are grouped to form the input of the neural network, while the data from the output of the nominal control of the super twisting SMC are saved as target data that will define the output of the network.

iv. Load the newly arranged dataset and assign the appropriate variable taking cognizance of the vectoral form while assigning.

v. Create a feedforward network (fitnet) with hidden layer and output layer using the tan-sigmoid and linear transfer functions respectively. Set the number of neurons in the hidden layer to 32. Note: the higher the number of neurons, the better the fitness, but too high will lead to overfitting.

vi. The loaded data is then divided into three (3) parts; training data, validation data, and test data in the ratio 60:20:20 respectively.

vii. The network is then trained using the backpropagation Levenberg-Marquardt algorithm.

viii. The performance of the system was checked, and the training was carried out again if the mean square error is large, the test and validation curve exhibit different characteristics, or there is overfitting of data.

ix. Performance evaluation is carried out by mean square error (MSE) and regression analysis, and the network is sent to Simulink environment where the network block is then copied and used in place of the nominal control in the general Simulink design of the super twisting SMC

The flow chart of the neural network implementation is illustrated in Figure 3.2. The whole process of neural net implementation begins with the data collection and processing, and ends with the performance evaluation of the trained network. More so, the flow chart summarizes the design procedure for the employed shallow neural network.

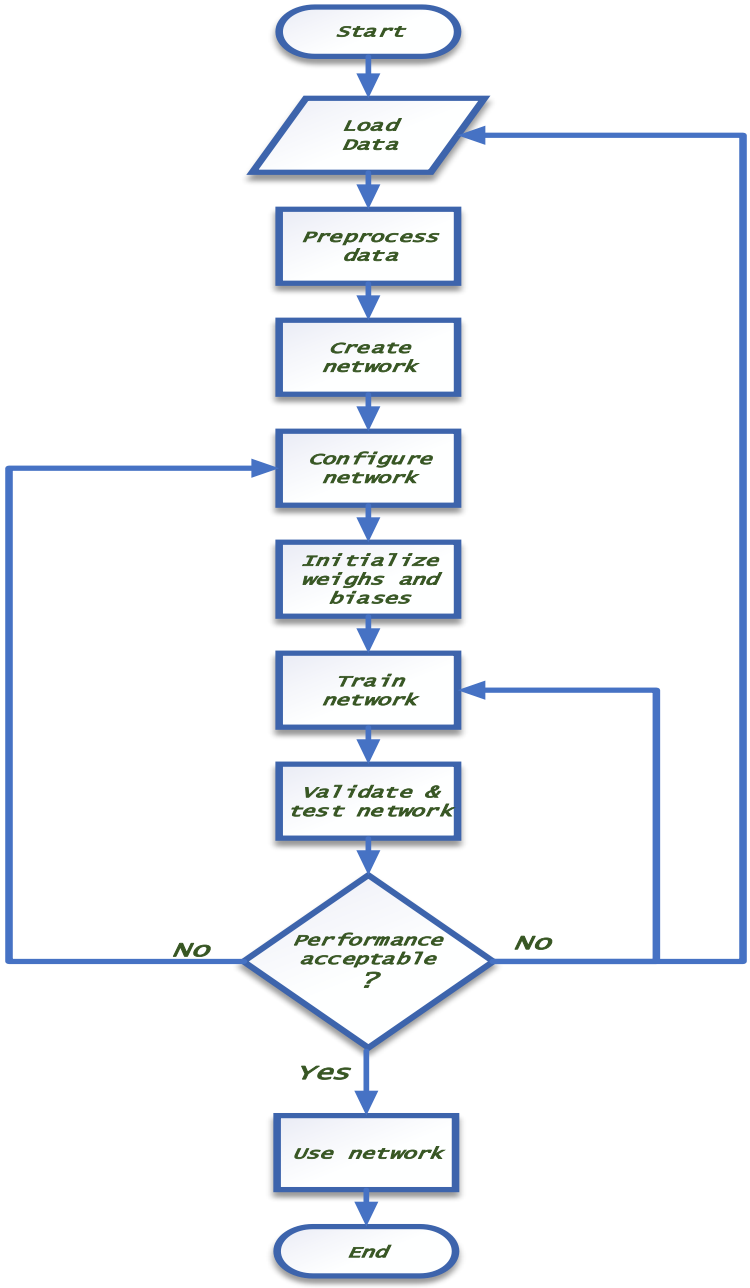


Figure 3.2: Neural network flowchart

At this instant, the data can be viewed by running the word "view (net)" in the command prompt to output the structure of the designed neural network. More so, based on the outcome of the training's evaluation as indicated by the regression analysis and also the validation and testing

output, the network can be retrained or even, the whole process can be restarted using better data samples and also configuring the network to yield better results as indicated in Figure 3.2. The structure of the network that is viewed using the command line "view(net)" is shown in Figure 3.3.

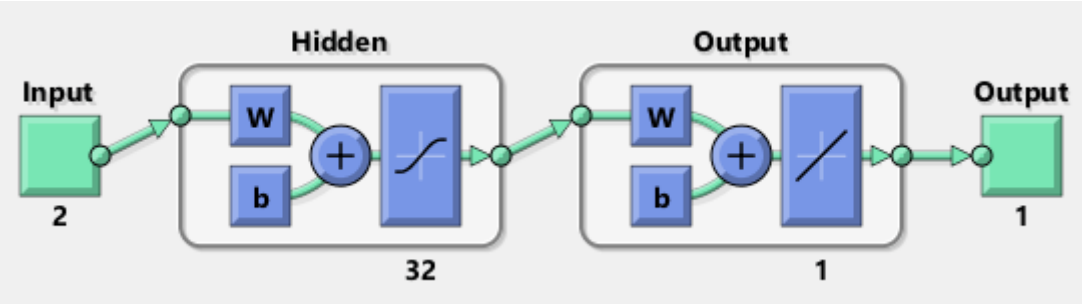


Figure 3.3: The structure of the designed neural network

Finally, looking under the mask of the designed neural network block in Simulink will show the internal structural arrangement as illustrated in Figure 3.4. This arrangement shows the interaction of the input to the output

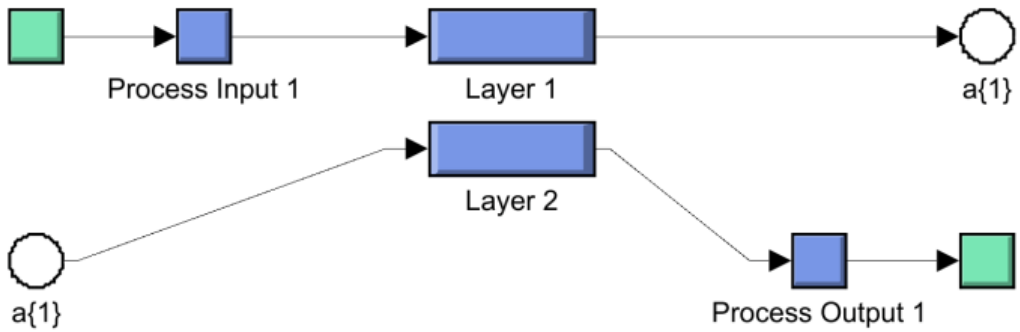


Figure 3.4: Neural Network Mask

Another method for designing this shallow neural network in MATLAB is the use of the preinstalled app called "nftool". Although it is still the same process applied in the procedure above, it offers a graphical user interface (GUI) option for the MATLAB script explained. For the script and generated code, refer to Appendix B.

3.2.6 Neural-tuned Super Twisting Sliding Mode control design

The proposed network was set up in Simulink/MATLAB as indicated in Figure 3.5. It can be noted that the nominal part of the super twisting SMC had been removed and replaced with the designed and trained neural network.

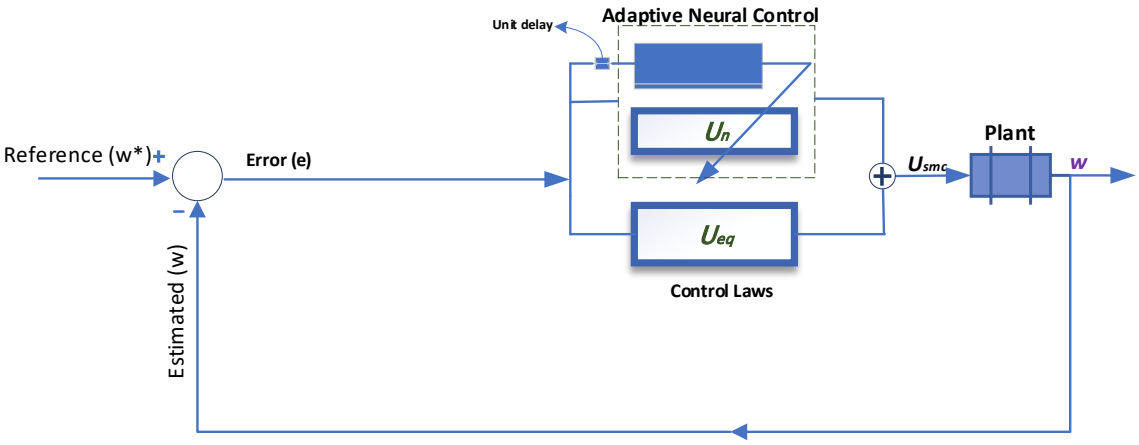


Figure 3.5: Proposed Model Diagram

3.3 Stability check for the selected SMC

Stability proof is carried out to ascertain that the designed control signal for the super-twisting sliding mode control guarantees the zero-convergence of the selected switching variable, that is, the Lyapunov function candidate is positive-definite, so also the time derivative must be negative-definite (Susperregui et al., 2011).

$$V = \frac{1}{2} S_{QP}^T S_{QP} \quad \text{and} \quad \dot{V} = \frac{1}{2}\left(\dot{S}_{QP}^T S_{QP} + S_{QP}^T \dot{S}_{QP}\right) = S_{QP}^T \dot{S}_{QP} < 0 \qquad \ldots (3.28)$$

Using Lyapunov stability function candidate

$$V = \frac{1}{2} s^2 \qquad \ldots (3.29)$$

The under-listed conditions have to be met to ascertain that the designed control signal guarantees a zero-convergence of the system to the choice of switching variable

- The Lyapunov function candidate is positive-definite
- The derivative of the Lyapunov function candidate must be negative-definite

$$\dot{V} \leq 0 \; for \; asymptotic \; stability$$

- As the sliding variable tends to infinity, the Lyapunov function candidate must also tend to infinity

$$\lim_{s \to \infty} V = \infty \qquad \ldots (3.30)$$

From investigation, condition (iii) is already satisfied, that is, as $s \longrightarrow \infty$

$$V = \frac{1}{2} s^2 \Rightarrow \frac{1}{2} \infty^2 = \infty$$

So $V \longrightarrow \infty$.

Since $V = \frac{1}{2}s^2$, the derivative will be

$\frac{dV}{dt} = \frac{dV}{ds} \times \frac{ds}{dt} = \frac{1}{2}(2s) \times \dot{s} = s\dot{s}$

$\dot{V} = s\dot{s} = s(\ddot{e} + c\dot{e}) = s(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - u \cdot a_1 - cx_2)$

$since\ u_{smc} = \frac{1}{a_1}[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda sign(s)]$

Thus,

$$\dot{V} = s\Big(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - \Big(\frac{1}{a_1}[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda sign(s)]\Big) \cdot a_1 - cx_2\Big)$$

$\dot{V} = s\big(-\lambda sign(s)\big) = -\lambda s sign(s) \leq -\lambda|s|$

Hence,

$\dot{V} \leq -\lambda|s|$ ... (3.31)

Where λ is a constant, this implies that the value of $\dot{V}$ is negative-definite.

To check for the robustness of the system, we consider the disturbance $d(t)$; this indicates that the equation for $\dot{x}_2$ becomes

$$\dot{x}_2 = -(a_2 + a_3)x_2 - (a_2a_3 + a_4)x_1 + u \cdot a_1 + d(t)$$

Therefore,

$$\dot{V} = s\dot{s} = s(\ddot{e} + c\dot{e}) = s(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - u \cdot a_1 + d(t) - cx_2)$$

which, when is substituted, will yield

$$\dot{V} = s\Big(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - \Big(\frac{1}{a_1}[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda sign(s)]\Big) \cdot a_1 + d(t) - cx_2\Big) \quad \text{... (3.32)}$$

$\dot{V} = s\big(d(t) - \lambda sign(s)\big) = sd(t) - \lambda s sign(s) \leq |s||d(t)| - \lambda|s| \leq |s|(|d(t)| - \lambda)$

$\dot{V} \leq |s|(|d(t)| - \lambda)$

If $(|d(t)| - \lambda) \leq -\delta$ and also negative definite, then the robustness of the SMC controller design is possible when $\lambda > |d(t)|$ or $\lambda > |d(t)| + \delta$

Accordingly, the reaching time of the sliding mode starting from any arbitrary time is as follows;

$$t_r = \frac{s(0)}{\delta} \qquad \ldots (3.33)$$

This proves that the designed conventional sliding mode control will guarantee zero-convergence of the system to the switching variable selected.

More so, for the Super twisting Sliding Mode Control, all other conditions for stability apply, and eq. (3.32) becomes

$$\dot{V} = s\Bigg(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - \Bigg(\frac{1}{a_1}\Bigg[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda_1|s|^{1/2}sign(s) + \int_0^t \lambda_2 sign(s)d\tau + d(t),\Bigg]\Bigg) \cdot a_1 - cx_2\Bigg)$$

Assuming the disturbance d(t) is zero

$$\dot{V} = s\Bigg(\dot{x}_{2d} + cx_{2d} + (a_2 + a_3)x_2 + (a_2a_3 + a_4)x_1 - \Bigg(\frac{1}{a_1}\Bigg[\dot{x}_{2d} + cx_{2d} + (a_2 + a_3 - c)x_2 + (a_2a_3 + a_4)x_1 + \lambda_1|s|^{1/2}sign(s) + \int_0^t \lambda_2 sign(s)d\tau\Bigg]\Bigg) \cdot a_1 - cx_2\Bigg)$$

$$\Longrightarrow \dot{V} = s(-(\lambda_1|s|^{1/2}sign(s) + \textstyle\int_0^t \lambda_2 sign(s)d\tau))$$

$$= -s\lambda_1|s|^{1/2}sign(s) - s\textstyle\int_0^t \lambda_2 sign(s)d\tau$$

Since the second term is a constant

$$\dot{V} \leq -\lambda_1|s||s|^{1/2}$$

3.5 Instruments/Materials used in the project

The simulations setup comprised of the following software and hardware items

- 20-Sim
- MATLAB 2020a and simulation software (Simulink)
- Microsoft Office tools (MS Word, Visio, PowerPoint, and Excel sheet)

- Core i5 processor laptop

3.6 Testing, Performance Evaluation, and Validation

The proposed control method was assessed on some standard requirements and performance as specified in the work of (Rakhonde & Kulkarni, 2018) and (Dursun et al., 2017; Maghfiroh et al., 2020), and the tracking performance was compared and presented using parameters such as:

- Rising time
- Settling time
- Overshoot
- Steady-state error
- Overshoot in loading (%)
- Time difference for settling in loading

The major evaluation is based on the following:

i. The behaviour of the load Torque with time as expressed in the Torque-time graph
ii. The behaviour of the wheel angular speed with time as expressed in the Speed-time graph
iii. The chattering outcome of the sliding mode controller
iv. The nature of the tracking, as illustrated in the $\omega_r - \omega$ Vs $time$ graph
v. Effect of the controller on the angular speed and current of the reaction wheel

Generally, the entire system as proposed is simulated using MATLAB platform (MATLAB scripts including Simulink, ode45 solver) in conjunction with the Neural Network toolbox and a mini-test of the training, validation, and test performance carried out and recorded.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.1 Introduction

This chapter lays emphasis on the presentation of the results gotten from the design and simulation of the various processes implemented in achieving a robust control of the brushless dc motor. The various corresponding training, test, validation, and evaluation results as recorded are presented accordingly.

4.2 Implementation

This work was implemented on 20-Sim and MATLAB/Simulink R2020a. The system, which is made up of the dc motor and the attached flywheel (load), is modelled using the bond graphs language and then inputted into the 20-Sim software to verify the derived equations of states and also to run the open-loop simulation of the system. The dataset for the neural network training, validation, and testing was locally generated from the just implemented super twisting SMC model running on Simulink in this project. Furthermore, the neural network implemented is the shallow neural network of just two (2) layers; the hidden and output layer. The entire Simulink models for all the controllers designed in this project were individually ran in MATLAB using scripts.

4.3 Outcome of the Neural Network

The overall outcome of the designed neural network, which comprises the data sampling, data distribution, training, validation, and testing are presented in this section with the corresponding results.

4.3.1 Data Acquisition and distribution

The data for this network is a 100001×1 double sample for the s_i and also for the s_{i-1}, also the target output corresponds to the same samples of 100001×1 double data. Since the input is a combination of the s_i and s_{i-1} data, this makes the input data to be a matrix of 100001×2 double set, and the output matrix still 100001×1. These data were collected for 50 seconds while running the simulation for the super twisting SMC with tunable gain pairs and are divided into three parts for training, validation, and testing accordingly in the ratio 60:20:20. A depiction of this distribution is given in table 4.1 and also illustrated in Figure 4.1.

Table 4.1: Data Set Category and Distribution

Categories	Percentages (%)	Samples
Training	60	60001
Validation	20	20000
Testing	20	20000

Graphically, table 4.2 can be represented as shown in Figure 4.1 with the percentage distribution as selected.

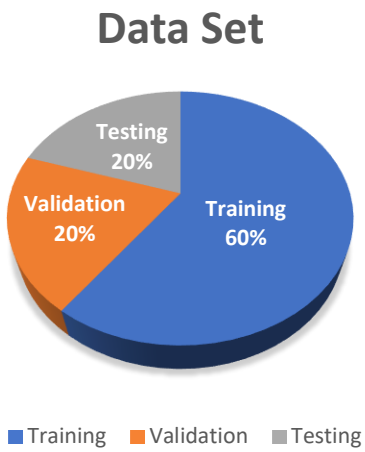


Figure 4.1: Dataset Distribution

This distribution was made so that data used for training are made available to the network when training is in progress, and the weights and biases are adjusted to account for the error. Also, the 20% data attributed to the validation set helps in network generalization, that is, the comparison between the output data and the target correlates, while the test set is used to measure how the training and trained network is performing.

4.3.2 Network training output

The GUI output to the neural network during and after training is represented in Figure 4.2. This figure shows the network configuration employed and the present progress of the training process including the algorithm used.

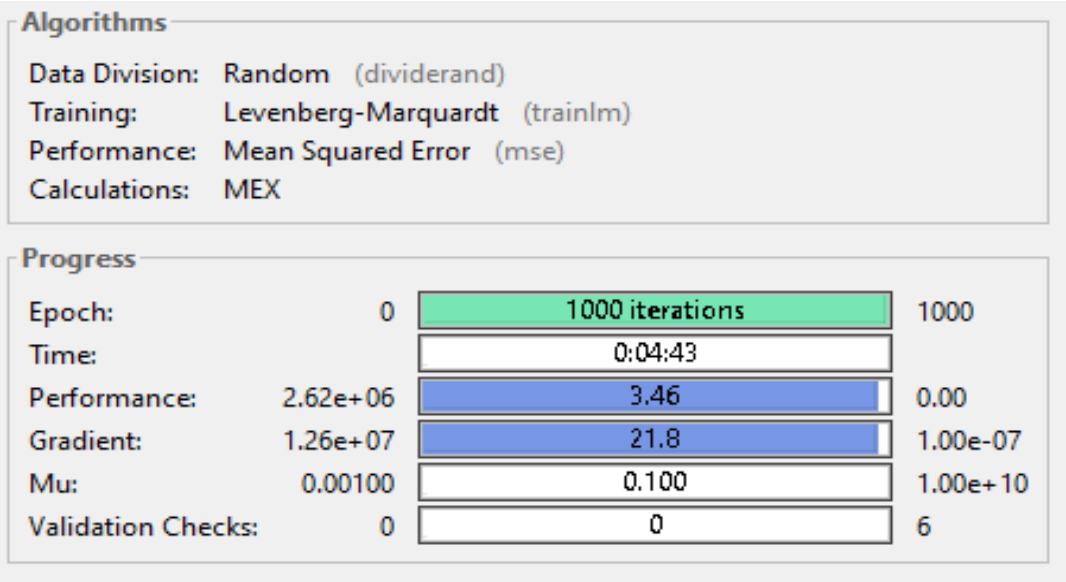


Figure 4.2: Neural Network training progress

At the end of the training, about 1000/1000 iterations took place and the average time it took the network to be trained and the validation checks shown in Figure 4.2. For this case, the generalization is not reduced since the check is not true for the six validations.

The tabular representation of the results outputted after the training is as shown in table 4.2.

Table 4.2: Training Results of Neural Net

Categories	Mean Square Error (MSE)	Regression (R)
Training	544.11056	0.99976
Validation	520.57077	0.99976
Testing	523.75732	0.99975

Figure 4.3 graphically shows the validation performance of the training set using the chosen training algorithm. The best performance in validation occurred at the 112 epoch with a value of 520.5708. Since the training curve tallies with the test and validation curve, it indicates that there is no overfitting and the process performance is good.

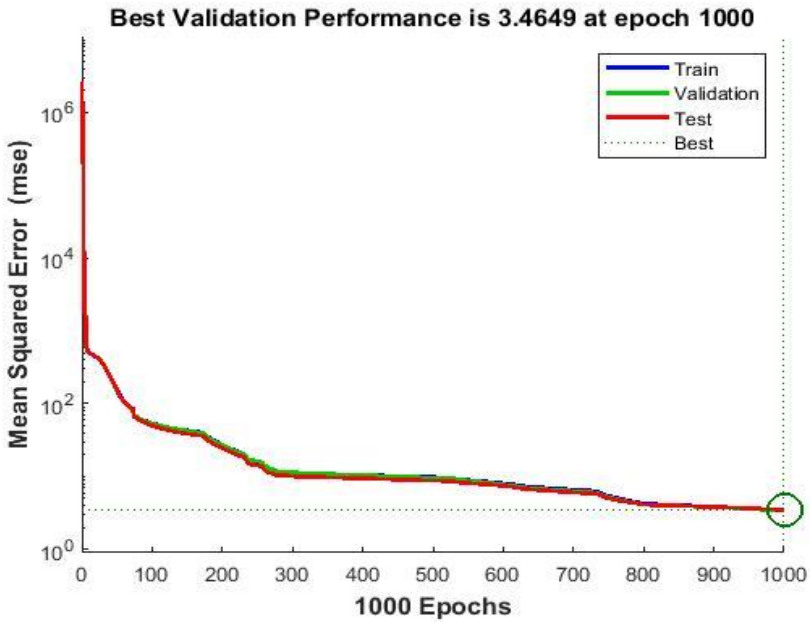


Figure 4.3: Validation Performance

Measuring the correlation between the output gotten and the targets gives the regression relationship. A value closer to one (1) indicates a close relationship between the output and the target, while values closer to zero (0) indicate a farther or random relationship. In Figure 4.4, the regression value is closer to 1, which indicates a very close relationship between the output and the desired outcome or target.

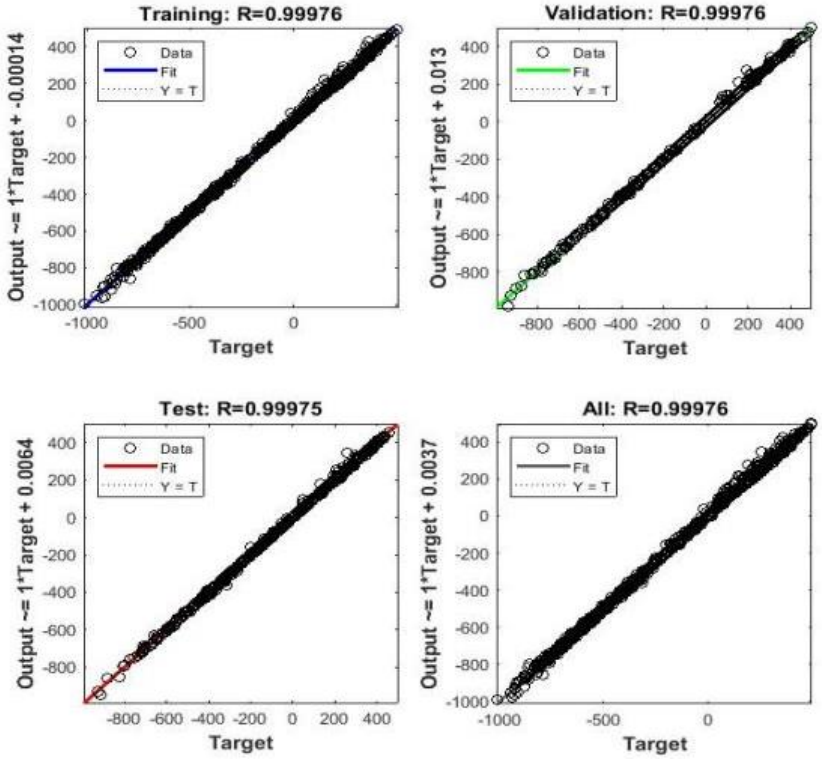


Figure 4.4: Regression Analysis

Since the general R-value is closer to 1, the relationship is better. Figure 4.5 shows the behaviour of the gradient and the training control parameter over the duration of the training.

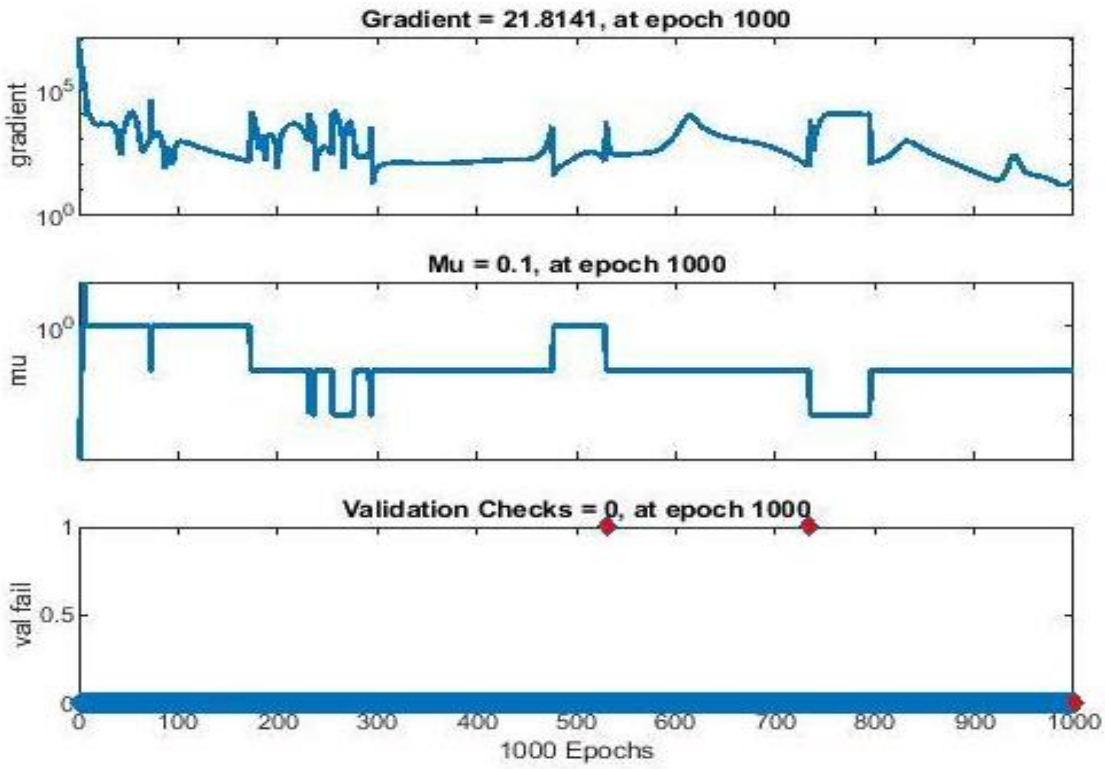


Figure 4.5: Training State

Figure 4.6 shows the error histogram of the network training. From indication, since the data are closer to the zero point and spread around -6.882 and 8.24, it is a better fit.

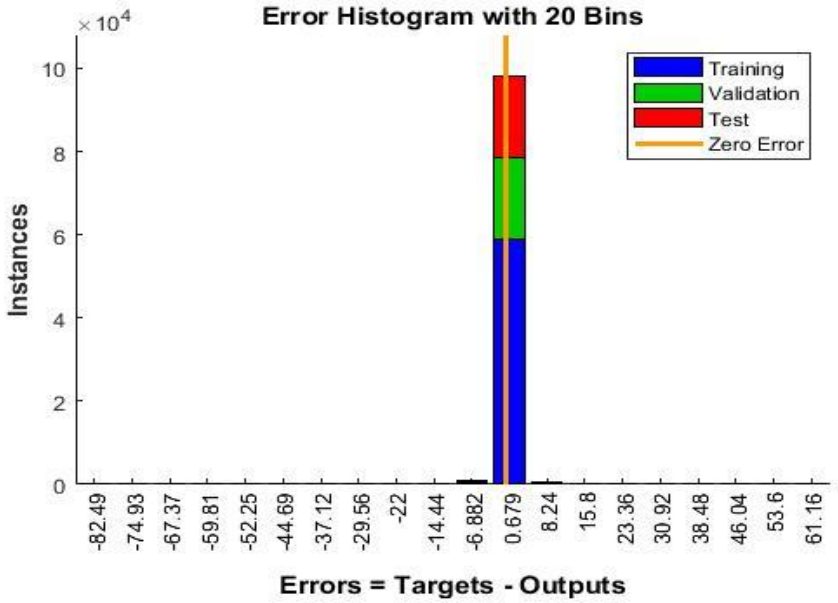


Figure 4.6: Error Histogram

4.4 Controller Design Outcomes

After designing the different controllers, the reference angular rate was set a $(sint - e^{-2t}sint)$ while the initial value of the angular rate was set at one. The corresponding responses of the individual setup were recorded and are presented below.

4.4.1 PID Control Implementation

The first design is the PID control implementation for the plant which is meant to provide the basis and the ease of comparison of the performance of the subsequent SMC controllers to be designed or implemented. Figure 4.7 shows the error output when the reference angular rate is employed. Since the initial value of the w is set at 1, at $t = 0$, the $w_{ref} = 0$. Hence the reason the curve starts at one. Also, not maintaining the error at zero shows that the controller is not controlling the plant in achieving the expected response.

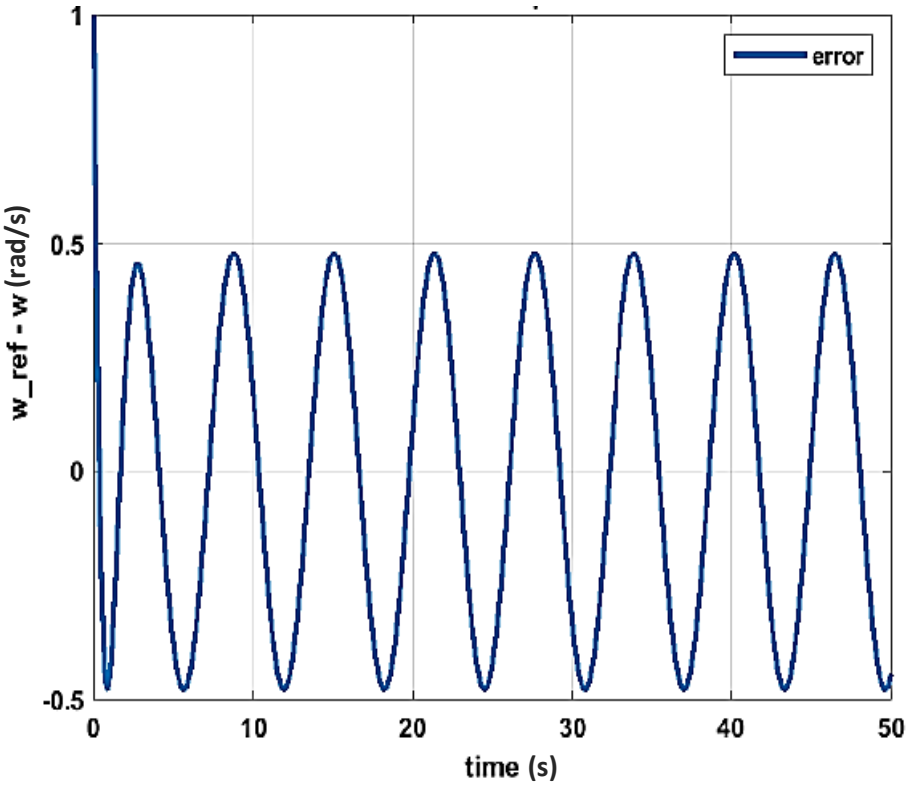


Figure 4.7: PID Error Output

Accordingly, after tuning the PID controller, the behaviour of the controller to the system is indicated in Figure 4.8.

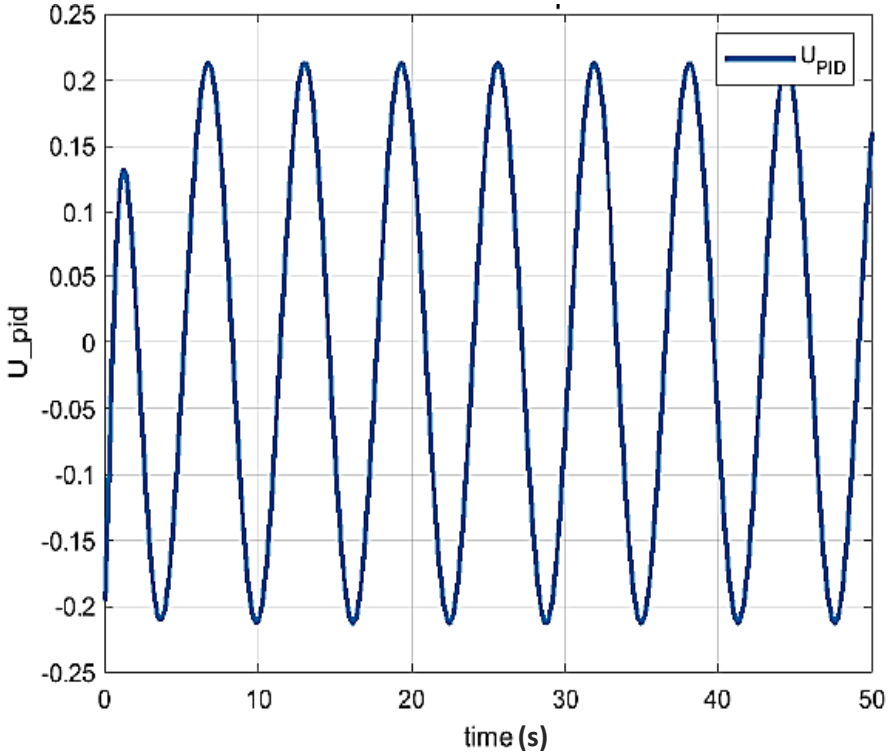


Figure 4.8: PID Control Input

Considering the relationship between the reference rate and the measured output rate, Figure 4.9 shows the tracking when PID is employed. This system is not tracking the reference as expected, hence the poor tracking output. Actual spike behaviour and out-of-phase tracking of the reference by the PID controller are illustrated in the zoomed diagrams of Figure 4.10.

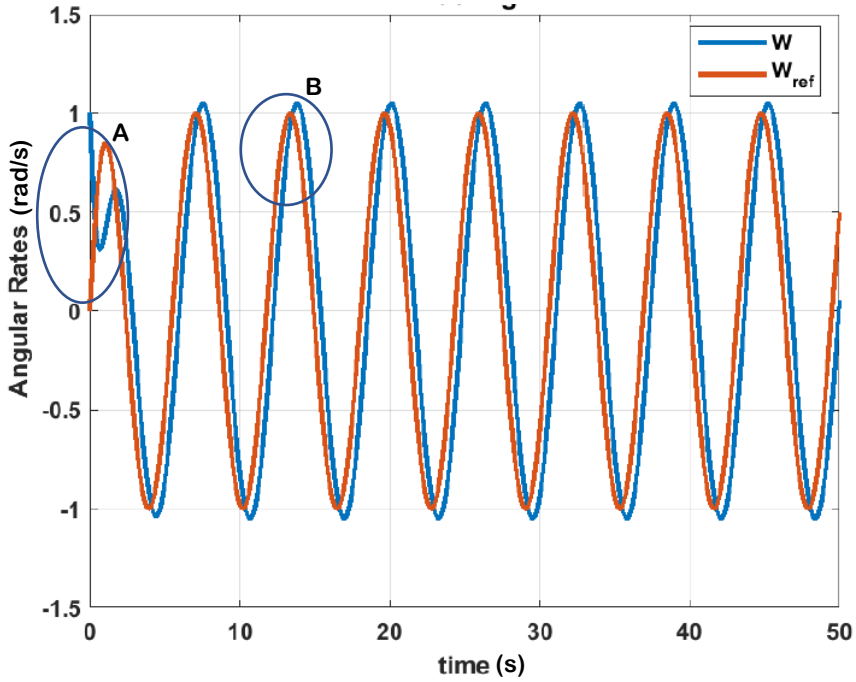


Figure 4.9: PID Tracking Output

Figure4.10A characteristics are as a result of the initial value of the angular rate being set at 1. Figure 4.10B is a zoomed part of the crest of the tracking output.

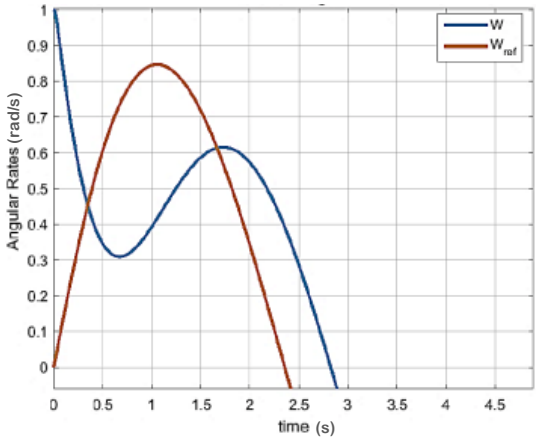


Figure 4.10A: PID Tracking (Zoomed Part A)

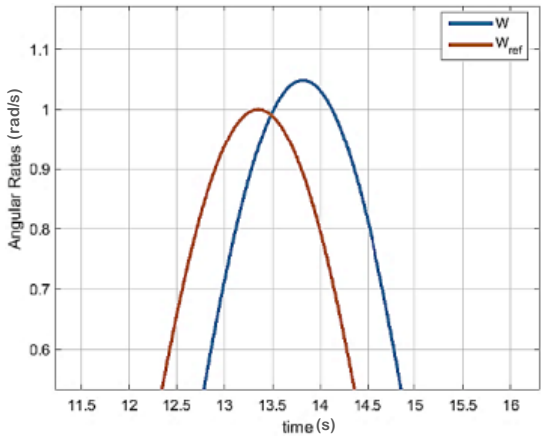


Figure 4.10B: PID Tracking (Zoomed Part B)

So, for more complex systems – like for those of order higher than 1 – the PID controller performance is poor, and the tracking is also less accurate.

4.4.2 Conventional Sliding Mode Control

For better performance, the conventional SMC is employed. Figure 4.11 illustrates the behaviour of the conventional sliding mode controller output. This is responsible for driving the system states to asymptotic stability by keeping the states in the sliding surface. Even with the better results, it can be noted that this control possesses some level of chattering in it as revealed.

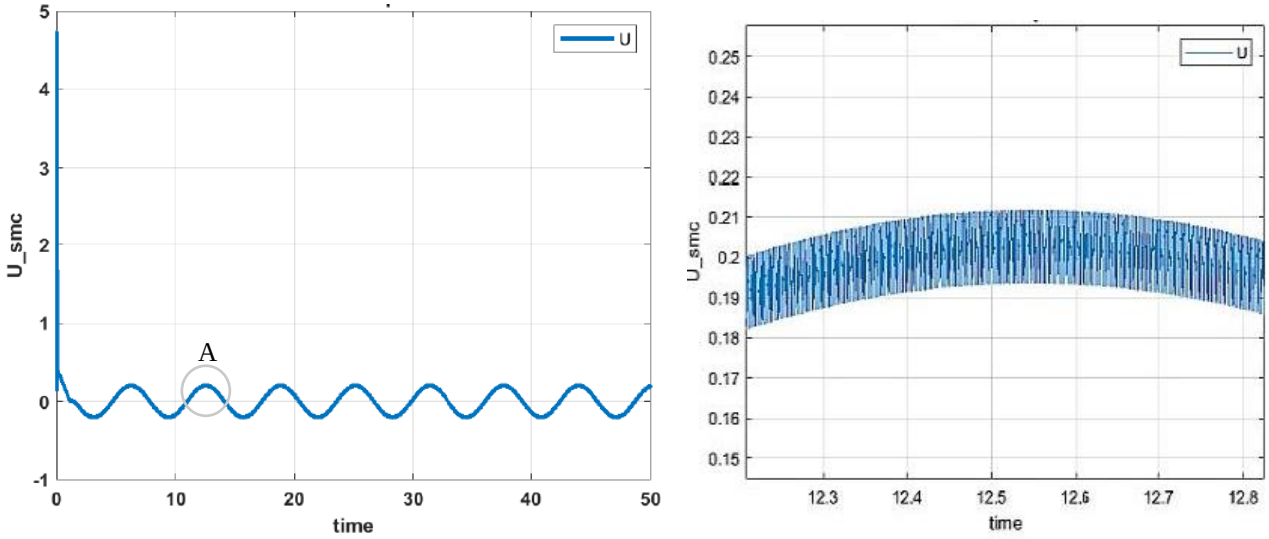


Figure 4.11: Conventional SMC Control Input

The sliding variable selected for the conventional SMC gave the output Figure 4.12.

Though the value of s tends to zero at a fast rate, a closer look as given in Figure 4.12A reveals that the value of *s* variable chatter. This chattering is the major issue of sliding mode controllers.

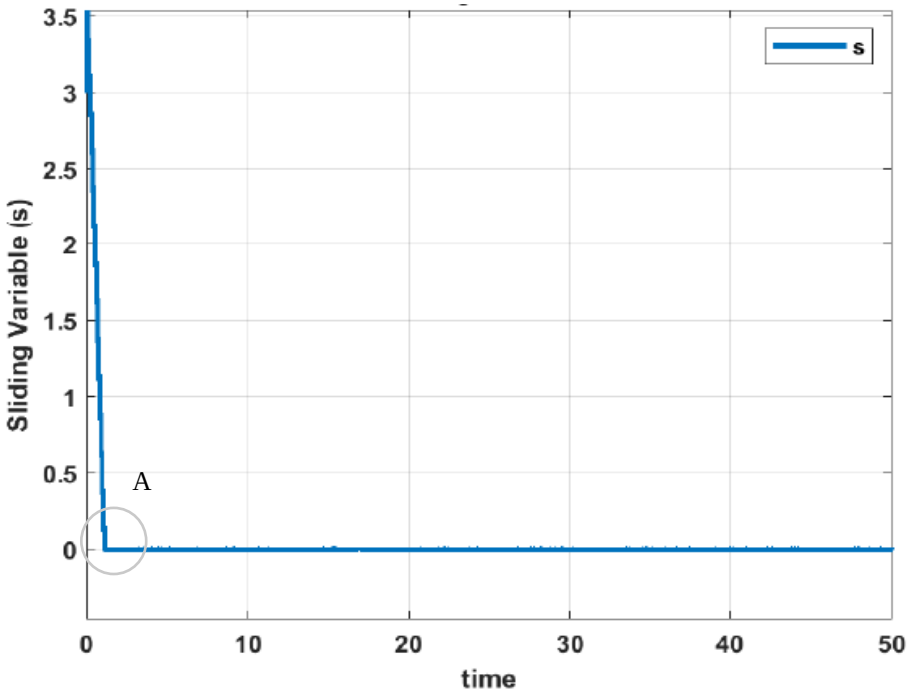


Figure 4.12: Conventional SMC Sliding Variable

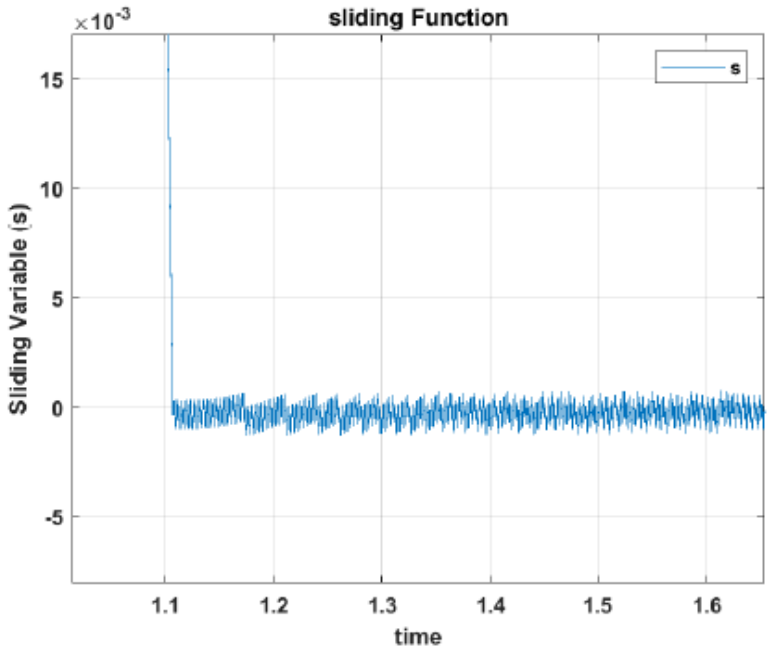


Figure 4.12A: Conventional SMC Sliding Variable (Zoomed part A)

More so, the sliding mode control provides an appreciable tracking function as illustrated in Figure 4.13, with the corresponding error depicted in Figure 4.14

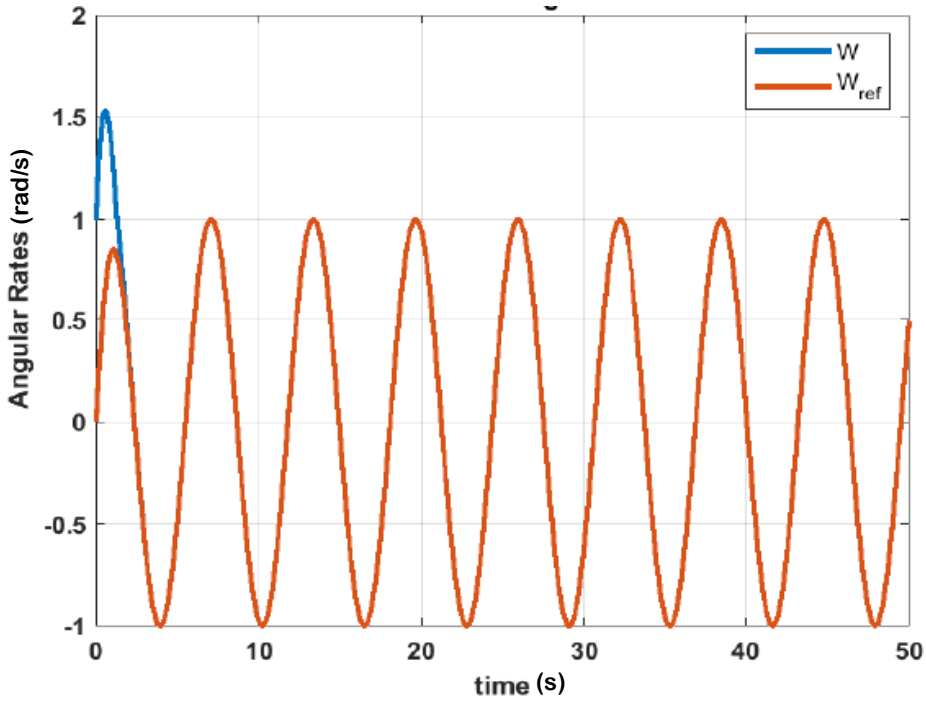


Figure 4.13: Conventional SMC Tracking Output

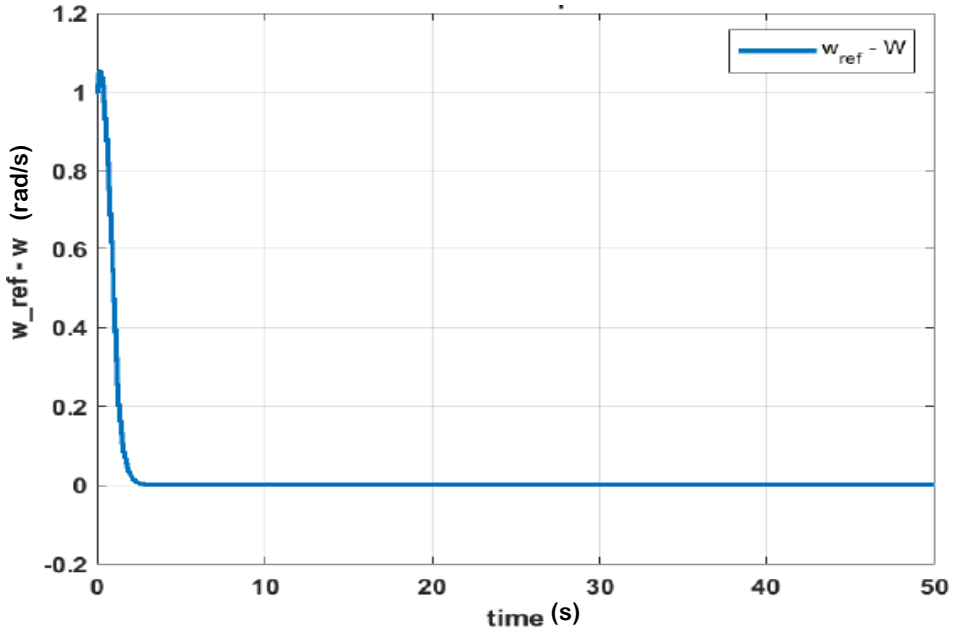


Figure 4.14: Conventional SMC Error Output

4.4.3 Super Twisting sliding mode controller

When the tunable gain pairs relationship is set to 200000 in the super twisting mode controller setup, the error output is as presented in Figure 4.15. It is worth noting that as the gain pairs

are tuned, the behaviour of the system changes. To get the system to operate at maximum performance, the trial-and-error method have to be repeated until a reasonable output is reached

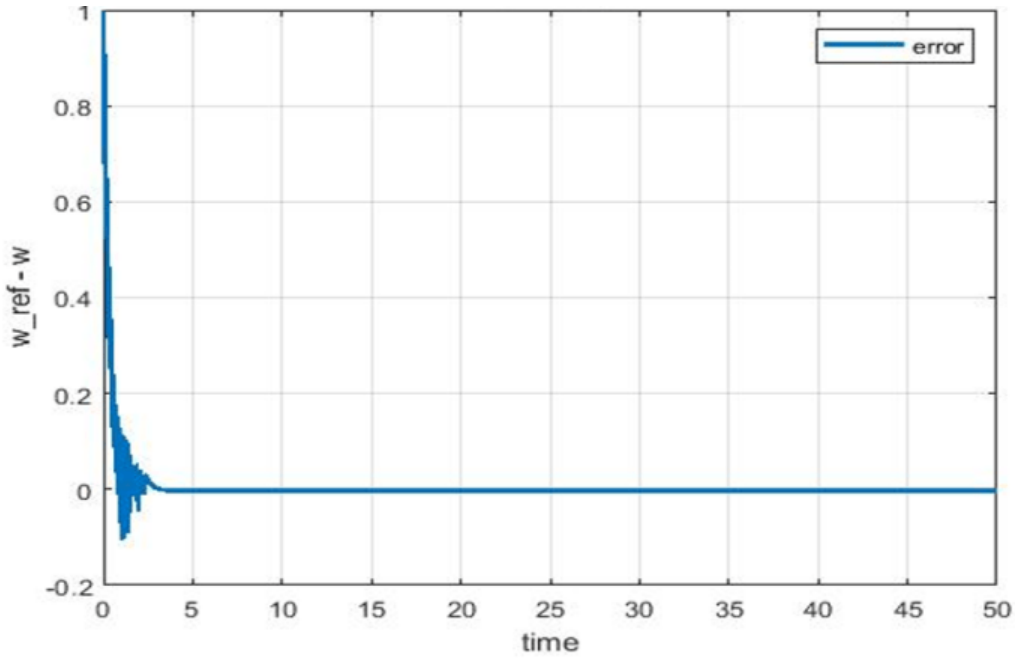


Figure 4.15: Super Twisting Error Output

Figure 4.16 depicts the corresponding behaviour of the sliding variable when the tunable gain pairs is set to 200,000. Before 2.5s, the chattering is very high, as with the value of the error. After the 2.5s margin, the system tends to behave optimally, and the chattering is more stable. This is indicated in Figure 4.17.

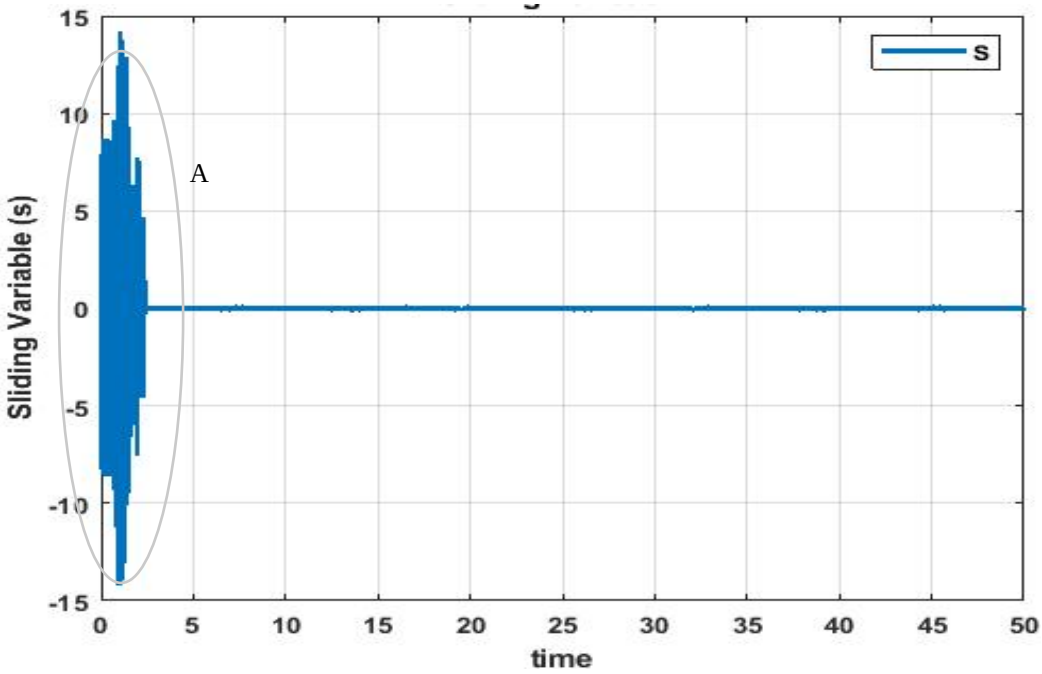
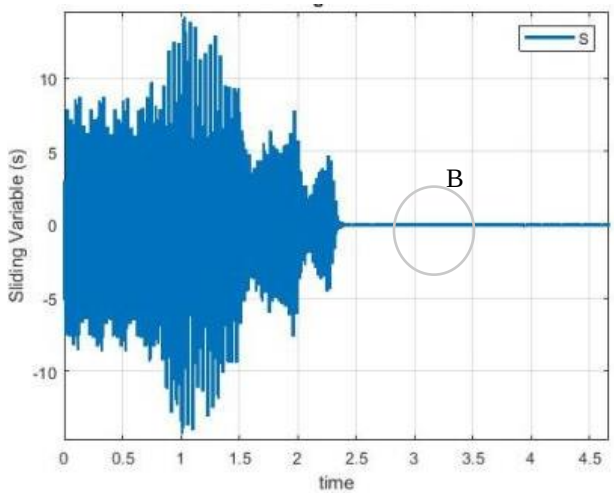


Figure 4.16: Super-Twisting SMC Sliding Variable



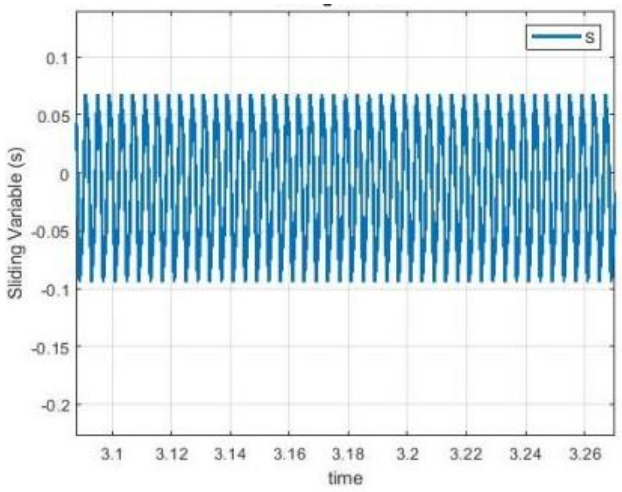


Figure 4.17A: STSMC Variable (Zoomed part A) Figure 4.17B: STSMC Variable (Zoomed part B)

Likewise, Figure 4.18, which represents the super twisting sliding mode control, exhibits an initial erratic behaviour similar to the error output and the sliding variable figures before becoming fairly stable but with the presence of much chattering.

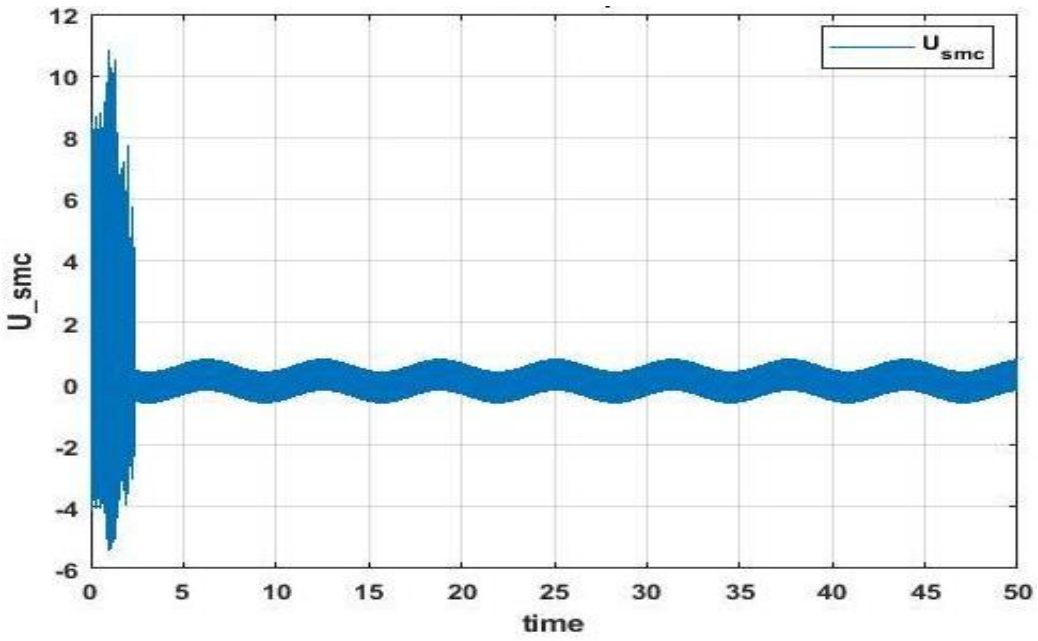


Figure 4.18: Super Twisting SMC Sliding Variable Controller Input

A closer view of the nature of the chattering can be seen in Figure 4.19. Each diagram shows a section of the chattering as at when it occurred.

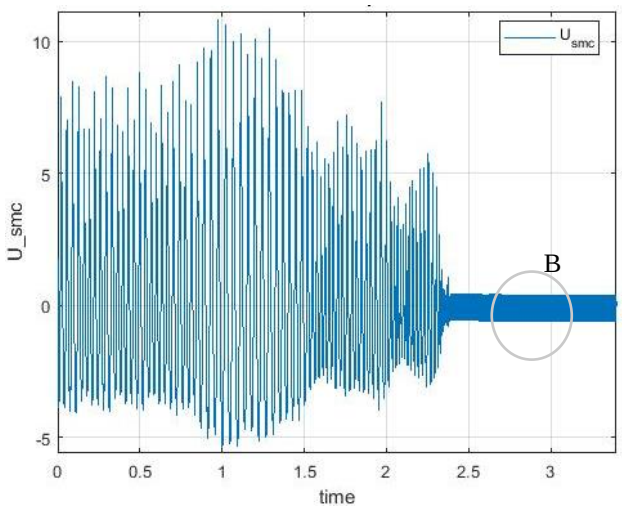


Figure 4.19A: STSMC Control (Zoomed part A)

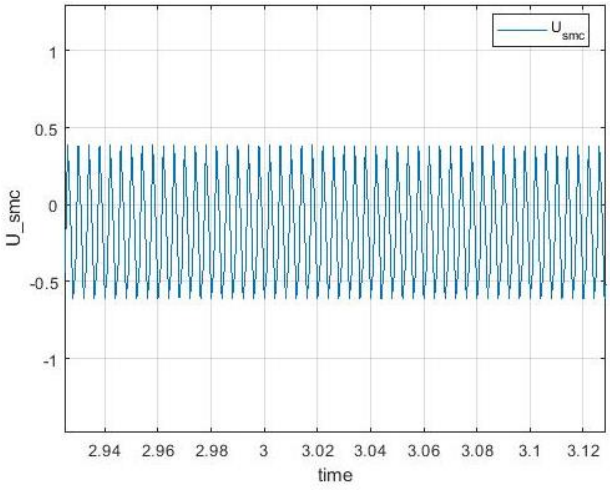


Figure 4.19B: STSMC Control (Zoomed part B)

The tracking output of the super twisting SMC showed acceptable tracking outcomes based on the selected "*W*" value which is 200,000. Although, as with all the outputs recorded thus far, for super twisting system, the initial 2 seconds noted, this characteristic behavior is dependent on the value of the tunable factor of the gain pairs "*W*" selected.

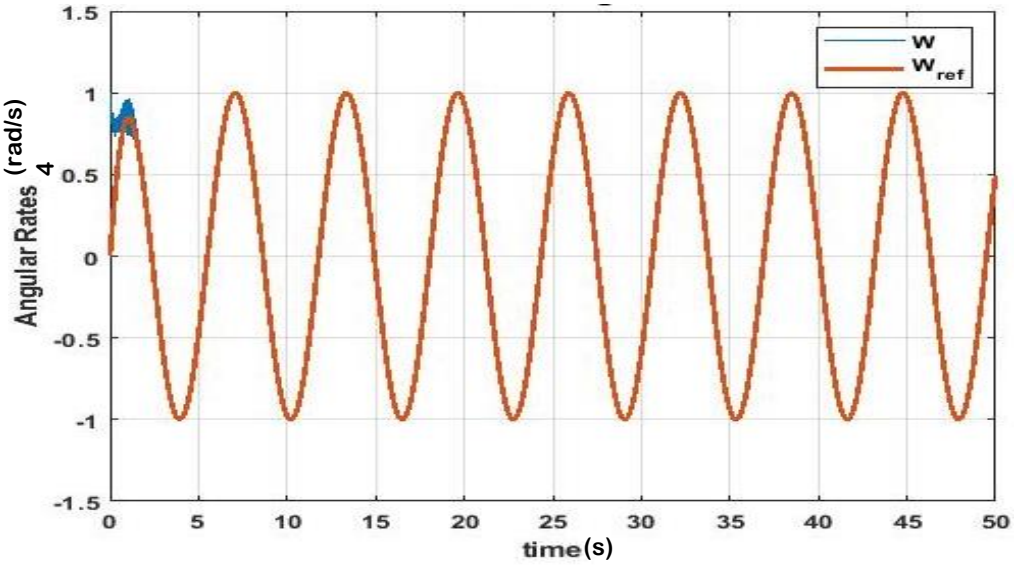


Figure 4.20: Super Twisting SMC Tracking output

All the graphs shown for the super twisting SMC setup are for the tuned gains.

4.4.4 Proposed Neural-Tuned Super Twisting Controller

On application of the proposed neural network tuned gain pairs, the chattering experienced in previous controller setups was greatly reduced to almost zero, as depicted in the graphs illustrated in Figure 4.21. Figure 4.22, Figure 4.23, and Figure 4.24. The control input behaviour shown in Figure 4.21 exhibits a very smooth behaviour, even in the zoomed diagram.

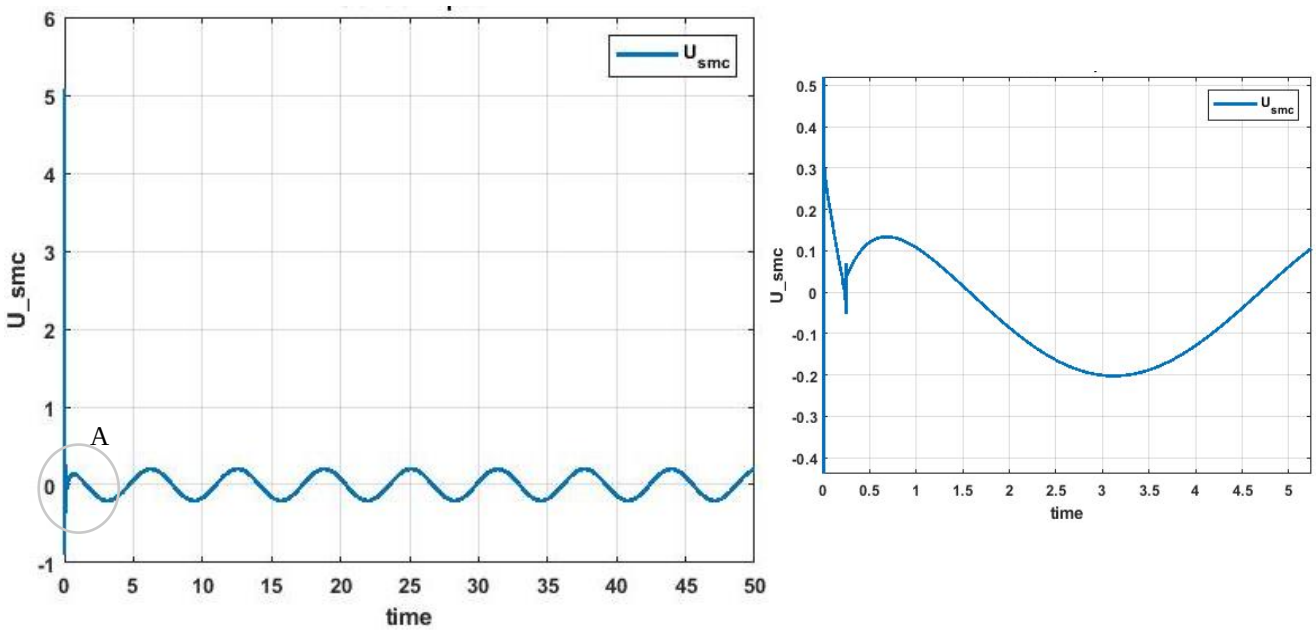


Figure 4.21: Control signal for the proposed neural tuned super twisting system

It is worthy of note that the subsequent chattering in Figure 4.18 and Figure 4.19 had been resolved and the system input behaviour had become refined. Also, the error output greatly improved. Figure 4.22 represents the error output when the super twisting SMC gain pairs were automatically tuned by the neural network. The outcome or curve reaches the equilibrium state in lesser time without serious chattering, as experienced in Figure 4.15.

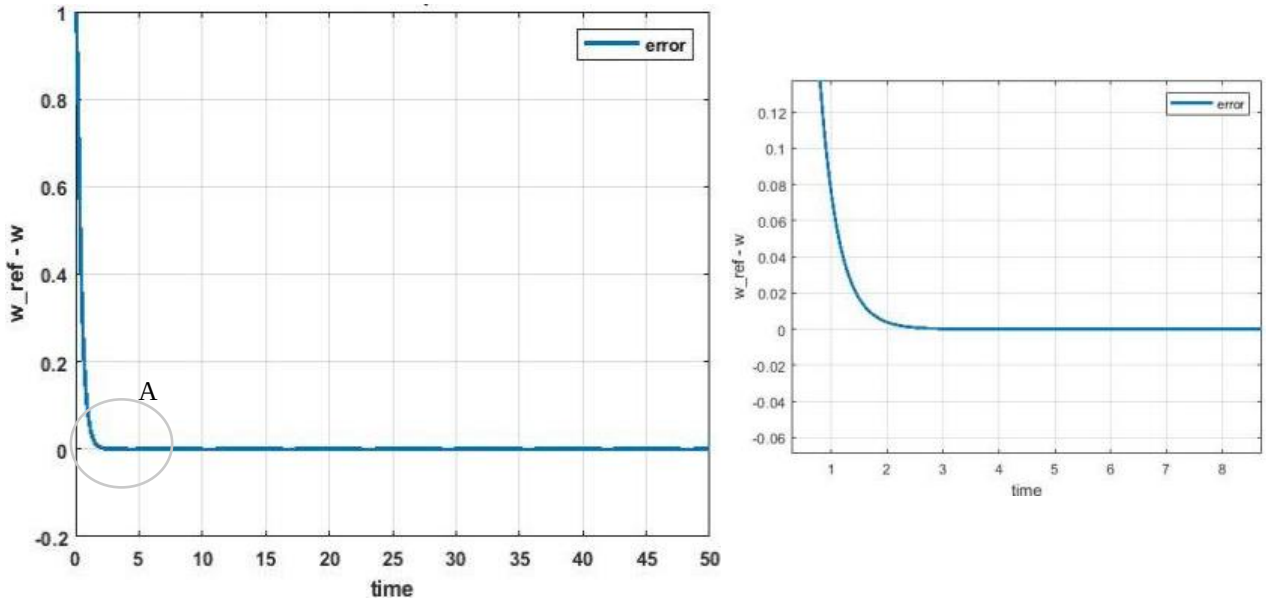


Figure 4.22: Neural-Tuned Super Twisting error output

Impressive sliding variable output, as depicted in Figure 4.23, where it reaches zero in minimal time compared to previous SMC controller setups. More so, the curve is smoother with a significantly reduced chattering effect in its behaviour.

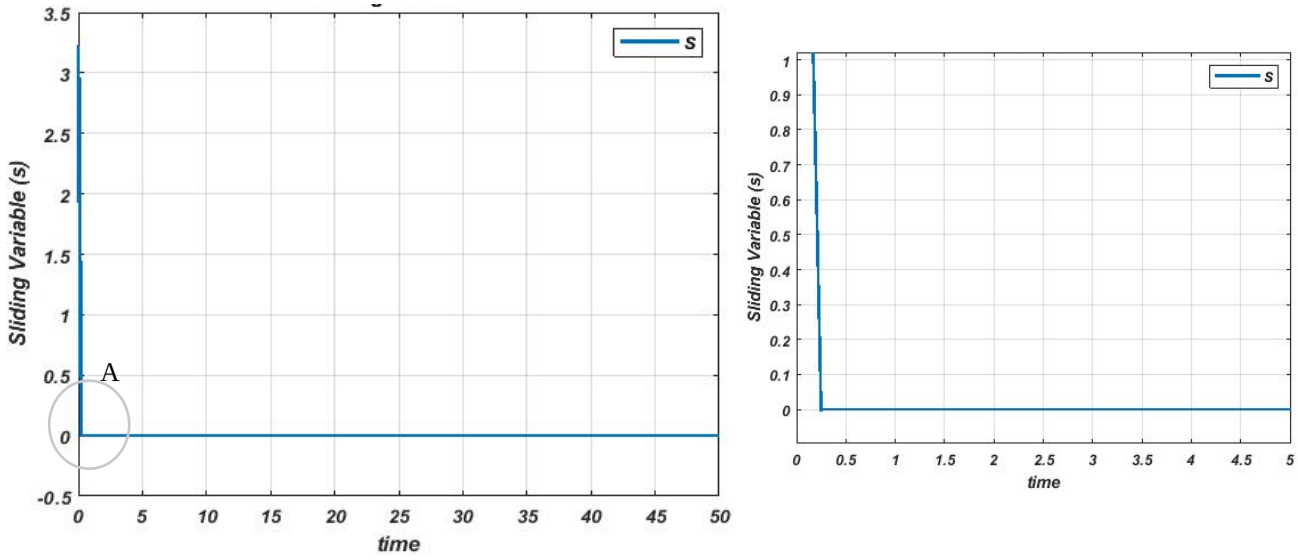


Figure 4.23: Neural-Tuned Super Twisting Sliding Variable

Another very impressive output of the Neural tuned super twisting SMC setup is the tracking performance represented in Figure 4.24. The estimated (outputted) angular rate tracks the reference value excellently. The curve for the output angular rate starts from one (1) since the initial value of *w* was set at 1 to enable us to see how fast the controller can account for it and correct the tracking.

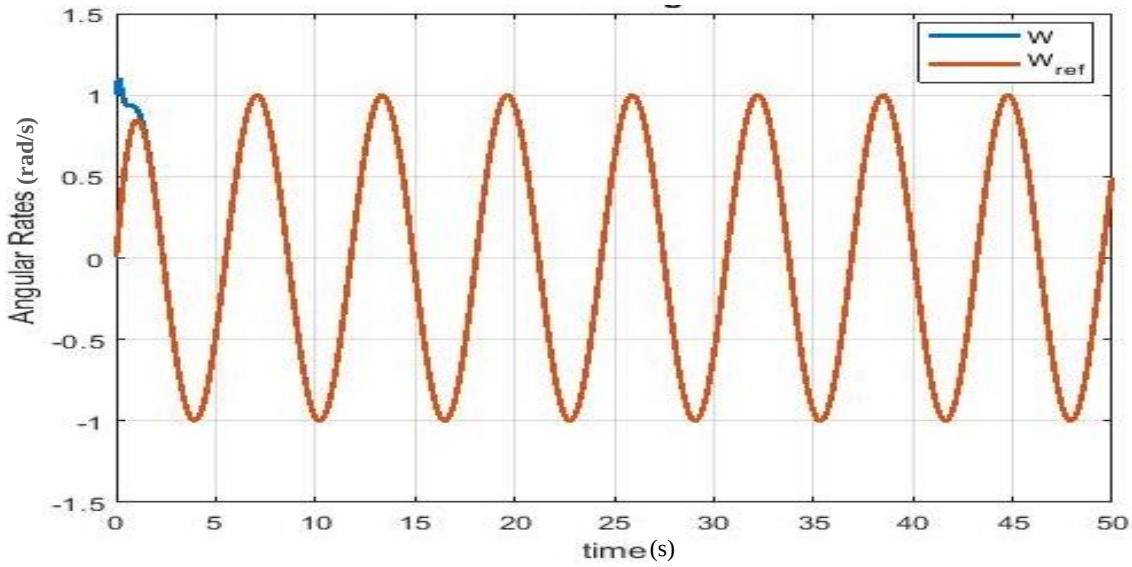


Figure 4.24: Neural-Tuned Super Twisting Tracking

4.5 Comparison of Controller Setups

This section entails the outcome of various comparisons carried out on the controllers. The corresponding effect of tuning the gain pairs relation, the error comparison, and output response tracking are the parameters considered for this comparison.

4.5.1 Tuning of super twisting SMC

For the tuning comparison, the reference signal is a step input and the initial value of x_1 which is the output angular speed is set at 0. This tuning aims to study the behaviour of the system when the gain pairs relations, W, is set at 50,000, 200000, 250000, and 500000. The corresponding outcomes are shown in Figure 4.25, Figure 4.26, Figure 4.27, and Figure 4.28.

***Case I*: $W = 50,000$**

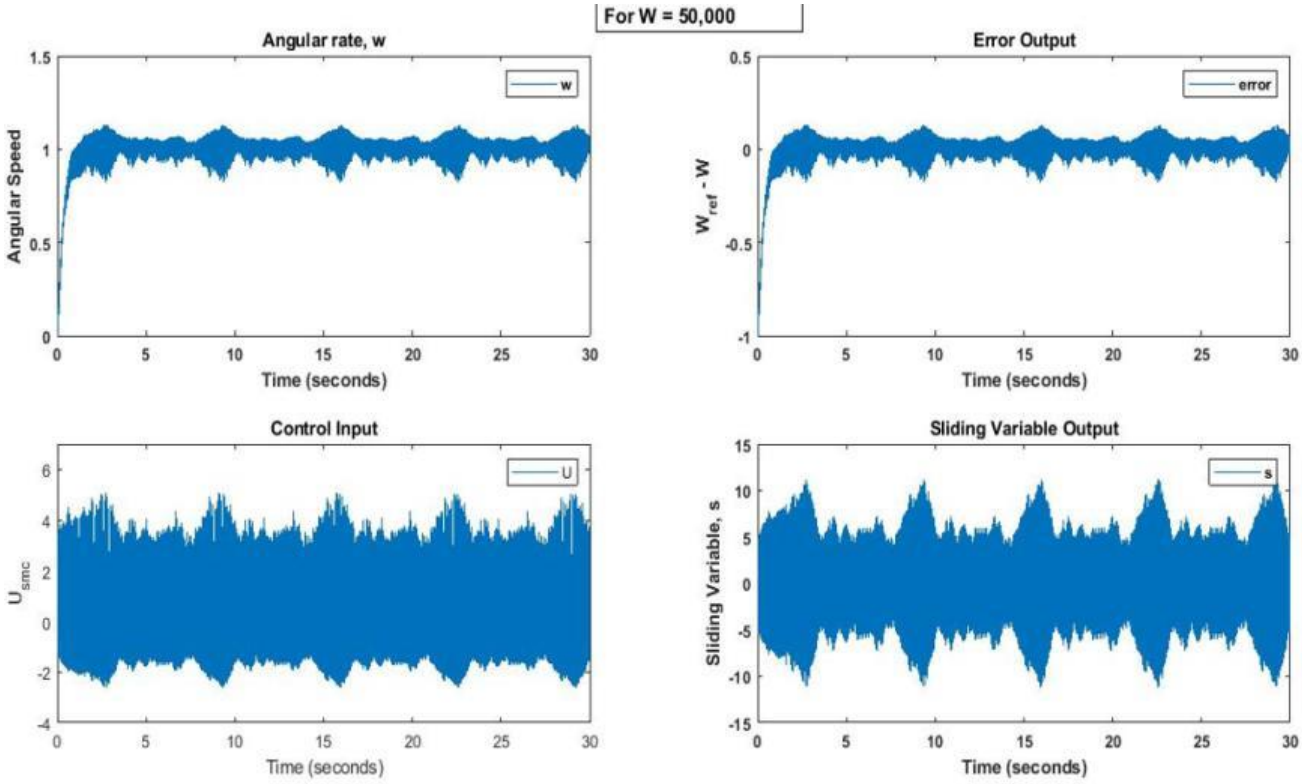


Figure 4.25: Outcomes of Super Twisting SMC at W = 50,000

Figure 4.25 illustrates the output angular speed, error gotten from the difference between the output signal and the reference signal, the super twisting control input to the system, and the sliding variable output.

***Case II*: $W = 200,000$**

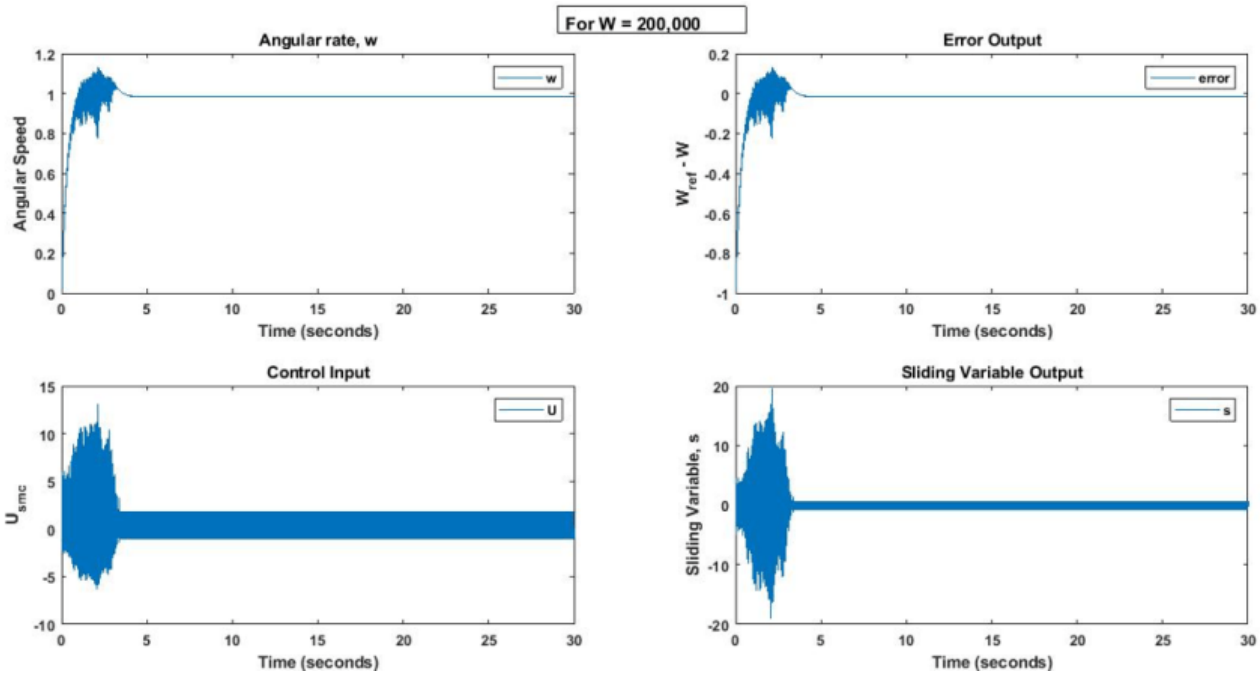


Figure 4.26: Outcomes of Super Twisting SMC at W = 200,000

In Figure 4.26, the outputs are better than the ones in Figure 4.25. although look at the initial value of the control and sliding variable, it indicated higher overshoots in Figure 4.26 than in Figure 4.25.

***Case III*: $W = 250,000$**

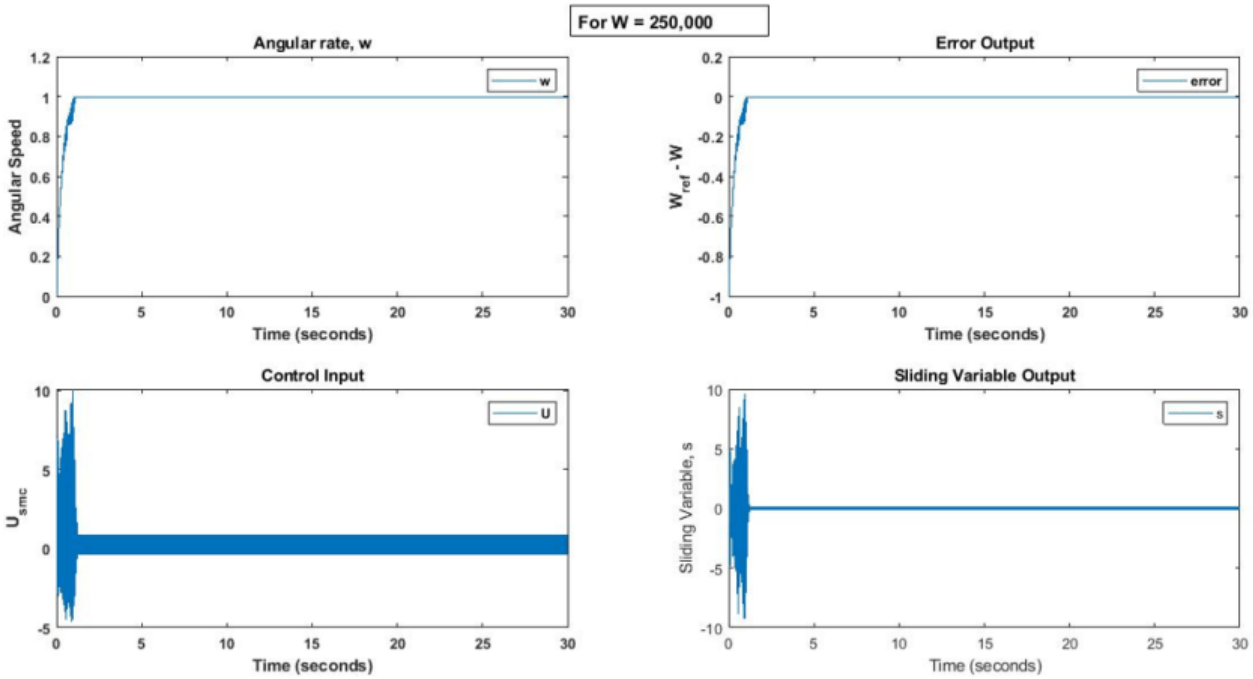


Figure 4.27: Outcomes of Super Twisting SMC at W = 250,000

Figure 4.27 is a finer representation of the system optimum performance when compared to the other two previous cases as depicted in Figure 4.25 and Figure 4.26.

***Case III*: $W = 250,000$**

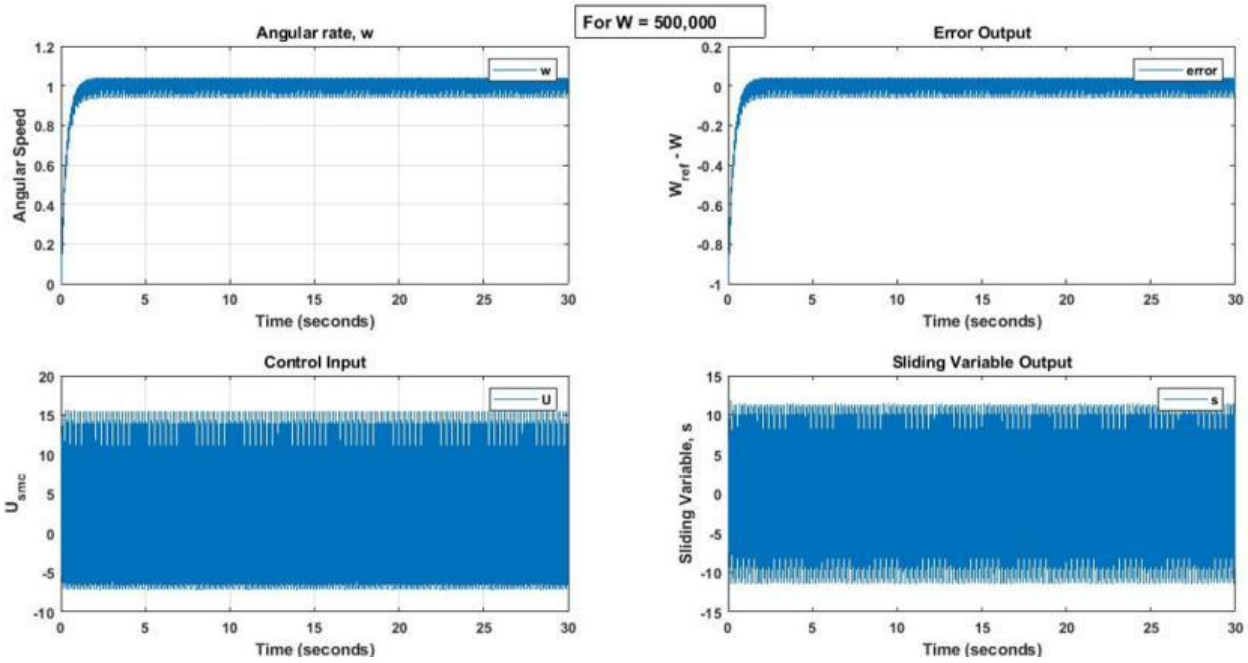


Figure 4.28: Outcomes of Super Twisting SMC at W = 500,000

In Figure 4.28, the behaviour of the system changed, indicating that the change in the value of the gain pairs will give a corresponding change in the outputs of the system for a super twisting SMC. Hence the necessity in applying an optimization algorithm that will help in fine-tuning these gain pairs without the trial-and-error method.

4.5.2 Error Comparison of all the Controllers setup

Using the sinusoidal reference input and the step reference input, the following results are obtained from all the controllers as indicated. Concatenating the four controllers, Figure 4.29 shows the error outcome when a sinusoidal signal is used as a reference. It can be noted that the PID error outcome is large since it is not performing good tracking. For a controller to be able to track effectively, the error output must tend to zero in the shortest possible time since the initial value of the output (angular speed, x_1) is set to 1. The corresponding value of super twisting SMC, conventional SMC, and the Neural-tuned super twisting SMC (proposed) are illustrated on the graph as well, where the best performance with lesser error is given by the proposed setup.

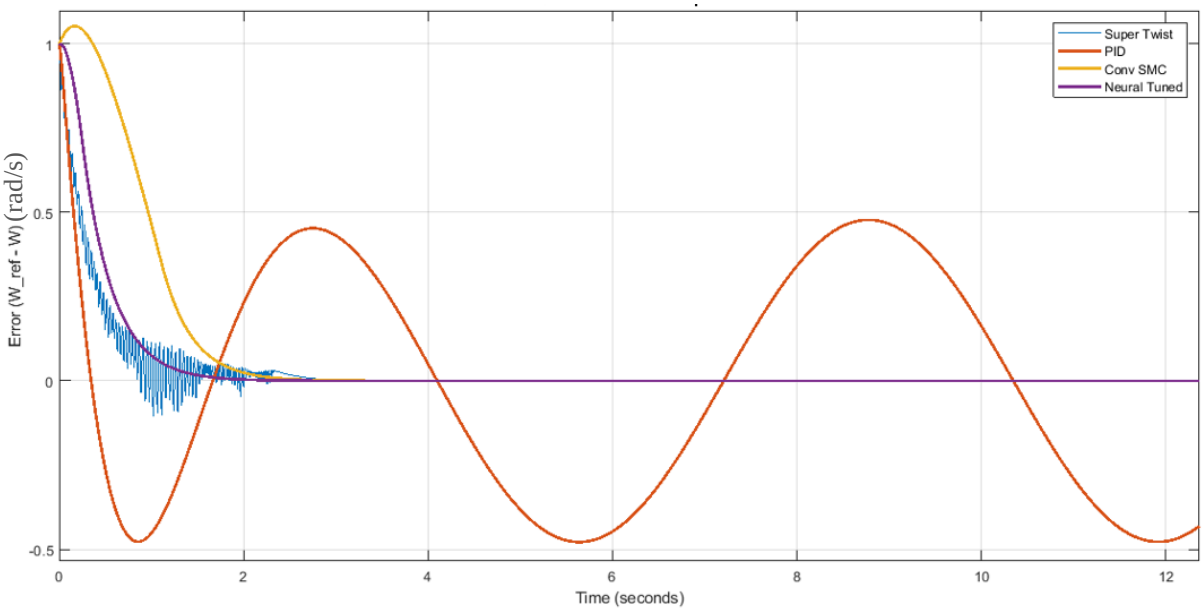


Figure 4.29: Error Outcomes for Sinusoidal reference input

Also, Figure 4.30 shows the error outcomes when a unit step reference signal is used. The best performance also goes to the proposed neural-tuned super twisting SMC since it converges fastest to zero and settles before the other controllers.

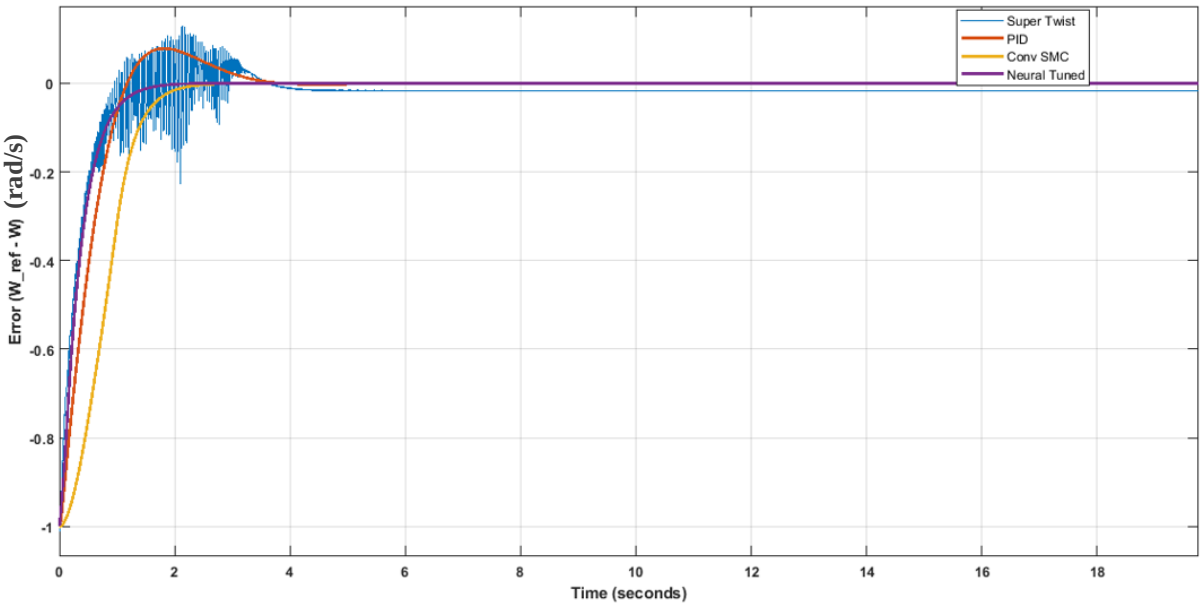


Figure 4.30: Error Outcomes for Unit Step Reference Input

The value of the selected gain pairs for the super twisting SMC affected its outcome as represented in Figure 4.30. Further tuning will reduce the initial chattering level and might also introduce steady-state error.

4.5.3 Tracking output comparison for all controller setups

Setting the initial value of the output (angular speed, x_1) to 1 and using a sinusoidal reference the tracking is represented in Figure 4.31. This figure shows that the PID controller tracking is poor as its output signal is out of phase with the reference signal. With little overshoot, the proposed Neural-tuned super twisting SMC had the best tracking performance.

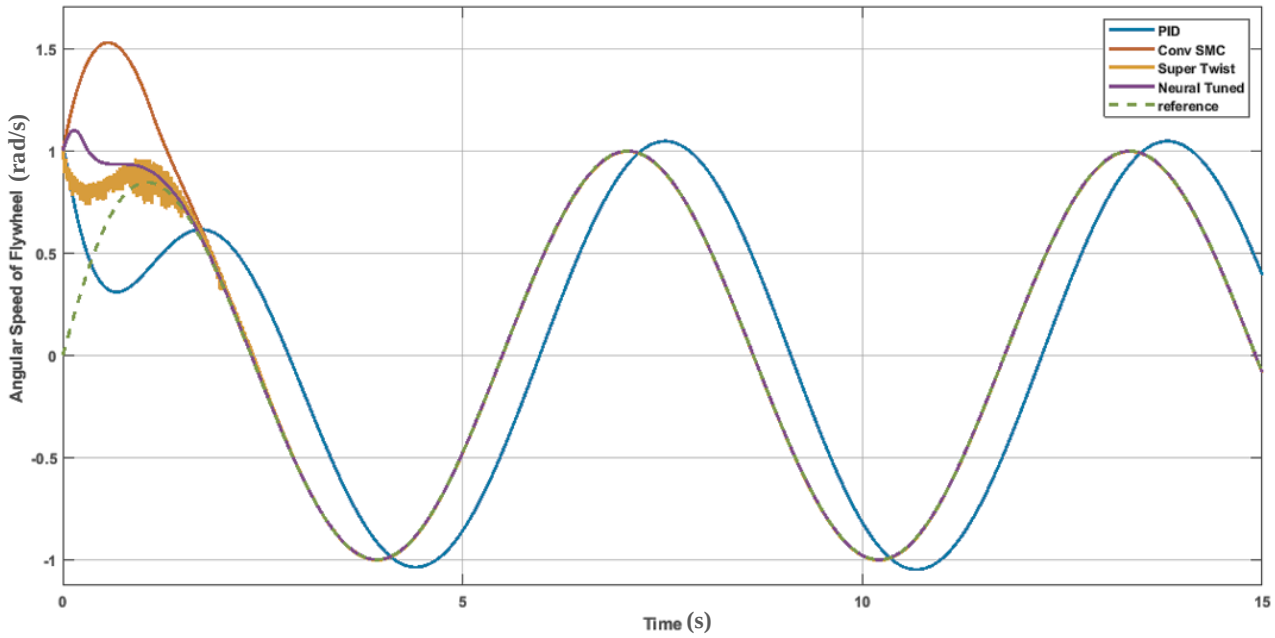


Figure 4.31: Sinusoidal Tracking outcomes for all the controllers

Subsequently, using a unit step reference signal and setting the initial value of the output (angular speed, x_1) to 0, the tracking outcome recorded in Figure 4.32.

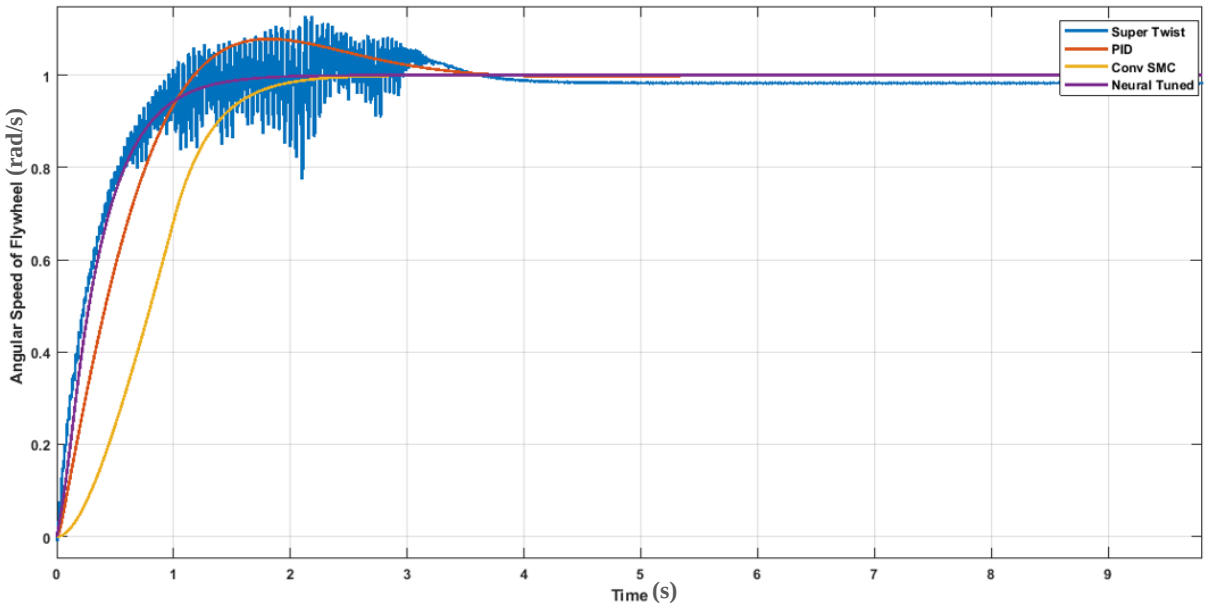


Figure 4.32: Unit Step Tracking outcomes for all the controllers

The proposed neural-tuned super twisting SMC also showed the best tracking output when subjected to unit step reference signal as shown in the diagram depicted by Figure 4.32.

4.6 Chattering Analysis

Chattering is a common phenomenon expected in the output of a sliding mode control. Taking a sample portion of the control input signal for the conventional and the proposed neural-tuned controller yielded the curve as illustrated in Figure 4.33.

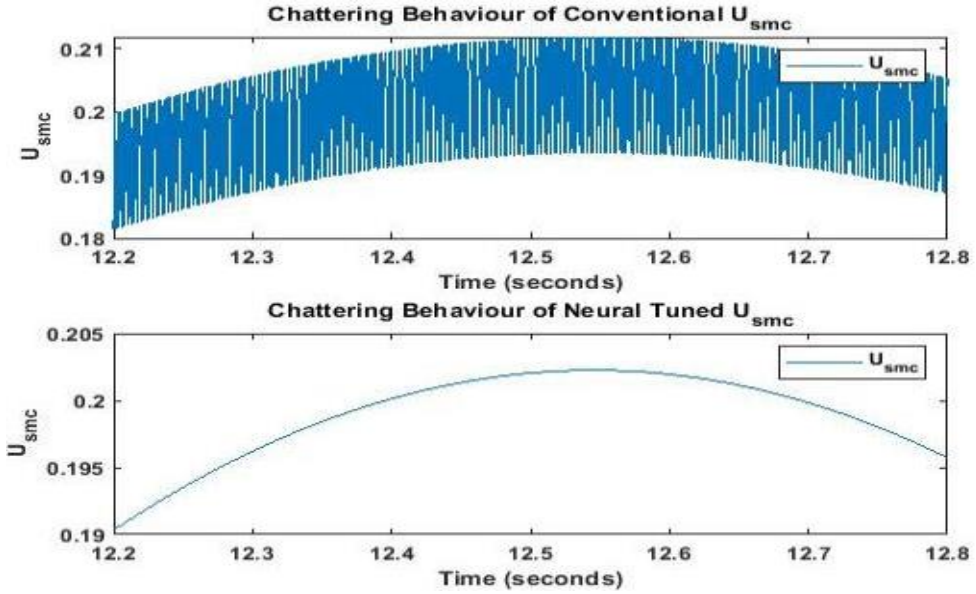


Figure 4.33: Control Input signal Chattering Analysis

Taking the time frame from 12.2 seconds to 12.8 seconds as illustrated in Figure 4.33, it can be noted that the conventional SMC control had much chattering as indicated when compared to the neural tuned control output.

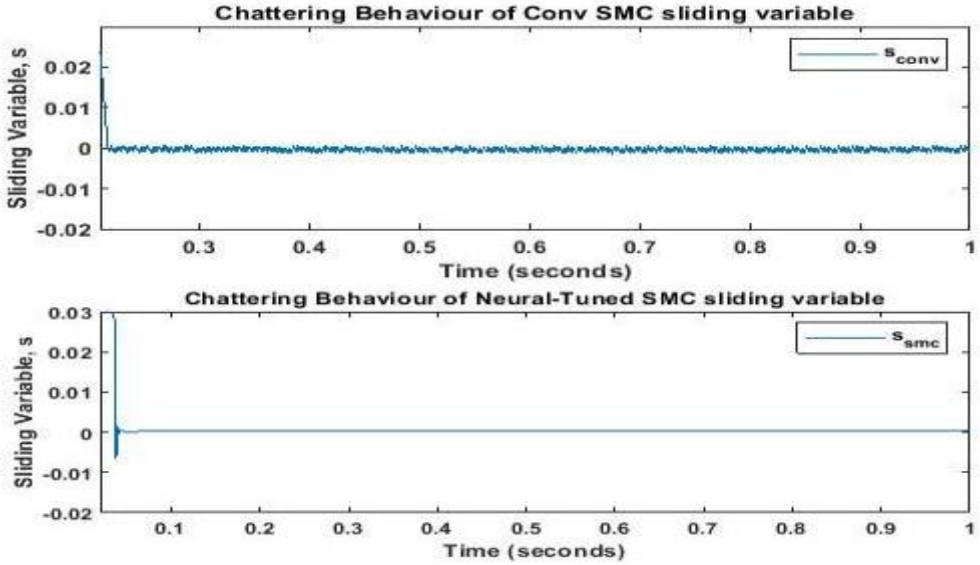


Figure 4.34: Sliding Variable Analysis

Also, considering the time frame between 0 to 1 seconds, the sliding variable exhibited high frequency switching for the conventional SMC unlike the neural-tuned SMC with almost zero chattering output for the sliding variable.

4.7 Performance Evaluation

To carry out the performance evaluation and analysis of this proposed Neural-tuned super twisting SMC with other controllers, as stated in the objectives of this work, after simulations in MATLAB, the system output response is gotten using "stepinfo" and the corresponding values are recorded in Table 4.3 with the inclusion of the steady state error.

Table 4.3: Controller performance (Unit Step Function)

Parameters	PID	Conv. SMC	Super-Twist SMC	Neural-tuned SMC
Rise Time (s)	0.5565	1,0884	0.0854	0.7471
Settling Time (s)	5.98	1.9232	49.9932	1.3493
Overshoot (%)	144.898	0.0412	95.4141	0
Undershoot (%)	0	0	18.5950	0
Steady State Error	0	2.012e-05	1.692e-02	5.951e-05
Peak	120	1.0002	1.3457	1.0001
Peak time (s)	5	38.7350	38.4900	10.3570

Using the PID performance as the standard for comparison, this implies that the PID controller performed better with its rise time except for the super-twisting SMC that is about 84% better. For the settling time, the conventional SMC performed 67.8% better while the super-twisting SMC in this case, performed poorly, and the neural-tuned SMC performed 77.4% better than the PID controller. With a percentage overshoot of 144.898 by the PID controller, the conventional SMC, super-twisting SMC, and the proposed neural-tuned SMC correspondingly performed about 34.2%, 99.97%, and 100% better. So, based on these performances, the proposed neural-tuned SMC yielded better response results when compared with the PID, conventional SMC, and the super-twisting SMC.

Considering the work of (Morfin et al., 2017), (Rakhonde & Kulkarni, 2018), and relating it to (Dursun et al., 2017), the performance is compared with the proposed setup as illustrated in Table 4.4.

Eq. (19) in (Rakhonde & Kulkarni, 2018), represents a plant model as:

$$\frac{\omega(s)}{U(s)} = \frac{325.54}{s^2 + 47.77s + 413.52}$$

Using the Eq.(3.11) : $\frac{\omega(s)}{U(s)} = \frac{a_1}{s^2+(a_2+a_3)s+a_2a_3+a_4}$

It implies that $a_1 = 325.54,\ a_2 + a_3 = 47.77,\ a_2a_3 + a_4 = 413.52$

Inserting these values into the proposed neural-tuned model yields the following results as shown in Table. 4.4 for the proposed Neural-Tuned Super-Twisting SMC.

Table 4.4: Comparison with Published works

Control Method	**Rise Time (Seconds)**	**Overshoot (%)**	**Undershoot (%)**
SMC (Rakhonde & Kulkarni, 2018)	0.086	0.71	1.53
STSMC (Morfin et al., 2017)	0.085	6.2	0
Proposed Neural-Tuned SMC	0.751	0	0

Generally, the proposed neural-tuned SMC yielded a much better performance when considering overshoot and undershoot results of the response as indicated in Table 4.5, hence it is preferable and recommended.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The results gotten from the proposed neural network tuned gain pairs showed promising output behaviour with significantly improved chattering reduction across all the system responses in addition to reduce steady-state error. Obviously, sliding mode control produces remarkable performance even under varying complex load parameters hence the robust nature of its operation. In contrast to the super twisting SMC outputs, the proposed setup provided excellent outcomes with faster rise and settling time as compared to others as shown in the aforementioned results. As such, using both step function and also the sinusoidal function had little to no impact on its performance. Furthermore, the tracking performance of the proposed setup exhibits promising and excellent characteristics.

Therefore, this neural tuned setup is very robust to matched disturbances introduced into the system and is recommendable in situations and applications that require robust controller settings like in the space technology applications.

Generally, this work has succeeded in providing a robust and adaptive control system that can improve the dynamic response speed, and the disturbance rejection capability of the brushless direct current motor using a second-order feedback control law to drive its states to origin and attaining a stable equilibrium asymptotically.

5.2 Limitations and Recommendations

Based on the work carried out in the design and simulation of this project, the following recommendations are proposed for future work:

- This work focuses on effectively controlling the angular speed hence the angular momentum of the load attached to the dc motor as it relates to the reaction wheel, so application of the designed controller to the actual collaborative attitude control of the spacecraft should be considered for future works. In addition to this, advanced work on using the real-life model of the reaction wheel, including all the viscosity, bearing, and friction
- Physical implementation and validation of the project.

REFERENCE

Agrawal, R. (2019). *Shallow Neural Networks - Towards Data Science*. TowardsDataScience. https://towardsdatascience.com/shallow-neural-networks-23594aa97a5

Ahmed, S. F., Raza, Y., Mahdi, H. F., Muhamad, W. M. W., Joyo, M. K., & Asadullah, S. (2019). *Review on Sliding Mode Controller and Its Modified Types for Rehabilitation Robots*.

Ambesange, S. V., Kamble, S. Y., & More, D. S. (2013). Application of Sliding Mode Control for the speed control of DC motor drives. *Proceedings of the IEEE International Conference on Control Applications*, 832–836. https://doi.org/10.1109/CCA.2013.6662853

Apogeeweb. (2019). *Structure, Function and Parameter of Transformer*. Apogeeweb Semiconductor Electronic. https://www.apogeeweb.net/article/206.html#ii-the-role-of-the-transformer

Banerjee, S. (1998). Dynamics for engineers. In *European Journal of Mechanics - A/Solids* (Vol. 17, Issue 2). https://doi.org/10.1016/s0997-7538(98)80092-2

Banerjee, S. (2005). *Dynamics for Engineers*. John Wiley & Sons, Ltd. https://nbn-resolving.org/urn:nbn:de:101:1-201411104485

Bayat, F., & Javaheri, M. (2020). Two-Layer Terminal Sliding Mode Attitude Control of Satellites Equipped with Reaction Wheels. *Asian Journal of Control*, *22*(1), 388–397. https://doi.org/10.1002/asjc.1878

Bessas, A., Benalia, A., & Boudjema, F. (2016). Integral Sliding Mode Control for Trajectory Tracking of Wheeled Mobile Robot in Presence of Uncertainties. *Journal of Control Science and Engineering*, *2016*, 7915375. https://doi.org/10.1155/2016/7915375

Borutzky, W. (2006). Bond graph modelling and simulation of mechatronic systems an introduction into the methodology. *20th European Conference on Modelling and Simulation: Modelling Methodologies and Simulation Key Technologies in Academia and Industry, ECMS 2006*, 17–28. https://doi.org/10.7148/2006-0017

Bryson, A. E. (2015). *Control of Spacecraft and Aircraft*. Princeton University Press. https://doi.org/doi:10.1515/9781400880034

Chacón, J., Vargas, H., Dormido, S., & Sánchez, J. (2018). Experimental Study of Nonlinear PID Controllers in an Air Levitation System. *IFAC-PapersOnLine*, *51*(4), 304–309. https://doi.org/10.1016/j.ifacol.2018.06.082

Chaudhari, D. (2019). *12 Difference between Motor and Generator in Tabular Form*.

DipsLab.Com. https://dipslab.com/difference-motor-generator/

Chen, H., & Wo, S. (2013). RBF neural network of sliding mode control for time-varying 2-DOF parallel manipulator system. *Mathematical Problems in Engineering*, *2013*. https://doi.org/10.1155/2013/201712

Chen, J., & Boyle, M. J. (2020). *Neural Network*. Investopedia. https://www.investopedia.com/terms/n/neuralnetwork.asp

Chi, S., Zhang, Z., & Xu, L. (2009). Sliding-Mode Sensorless Control of Direct-Drive PM Synchronous Motors for Washing Machine Applications. *IEEE Transactions on Industry Applications*, *45*(2), 582–590. https://doi.org/10.1109/TIA.2009.2013545

Dai, M., Qi, R., & Cheng, X. (2019). Super-twisting sliding mode control design for electric dynamic load simulator. *Chinese Control Conference, CCC*, *2019-July*(1), 3078–3083. https://doi.org/10.23919/ChiCC.2019.8865725

Damiano, A., Gatto, G. L., Marongiu, I., & Pisano, A. (2004). *Second-Order Sliding-Mode Control of DC Drives*. *51*(2), 364–373.

Dasgupta, I. (2002). *Working Principle of a Transformer* (First edit). McGraw-Hill Education. https://www.accessengineeringlibrary.com/content/book/9780070436404/chapter/chapter1

Daware, K. (2012). *What is Electrical Machine?* Electricaleasy.Com. https://www.electricaleasy.com/2012/12/what-is-electrical-machine.html

Derbel, N., Ghommam, J., & Zhu, Q. (2017). *Studies in Systems, Decision and Control 79 Applications of Sliding Mode Control*. http://www.springer.com/series/13304

Dewey, R. A. (2018). *General Systems Toolkit: Negative Feedback*. Psychweb. https://www.psywww.com/gst/negative-feedback.html

Diesel Service & Supply. (2016). *How Does A Generator Create Electricity? Article on How Generators Work*. Generator Source. http://www.dieselserviceandsupply.com/How_Generators_Work.aspx

DilKumar, T. R., & Mija, S. J. (2015). Design and performance evaluation of robust SMC schemes for speed control of DC motor. *Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014*, 88–92. https://doi.org/10.1109/ICACCCT.2014.7019235

Dursun, E. H., Levent, M. L., Durdu, A., & Aydoğdu, Ö. (2017). Speed Control of a Variable Loaded DC Motor by Using Sliding Mode and Iterative Learning Control. *International Journal of Electrical Energy*, *5*(1), 22–28. https://doi.org/10.18178/ijoee.5.1.22-28

Edwards, C., & Spurgeon, S. K. (1998). *Sliding mode control : theory and applications*. Taylor

& Francis.

Electrical Classroom. (2020). *An overview of Electric Machines*. Electricalclassroom.Com. https://www.electricalclassroom.com/electric-machines/

Emel'yanov, S. V. (2007). Theory of variable-structure control systems: Inception and initial development. *Computational Mathematics and Modeling*, *18*(4), 321–331. https://doi.org/10.1007/s10598-007-0028-6

Ferrara, A. (n.d.). *Sliding Mode Control Handout Advanced Automation and Control Course*.

Filippo, J., Delgado, M., Brie, C., & Paynter, H. (1991). A survey of bond graphs : Theory, applications and programs. *Journal of The Franklin Institute-Engineering and Applied Mathematics - J FRANKLIN INST-ENG APPL MATH*, *328*, 565–606. https://doi.org/10.1016/0016-0032(91)90044-4

Fridman, L., Barbot, J. P., & Plestan, F. (2016). Recent trends in sliding mode control. *Recent Trends in Sliding Mode Control*, 1–419. https://doi.org/10.1049/PBCE102E

Galphade, N. P., & Sankeshwari, S. S. (2015). Simulation of Bldc Motor Control Using Sliding Mode Control Technique. *International Journal of Advances in Engineering & Technology*, *7*(6), 1857–1865.

Gambhire, S. J., Kishore, D. R., & Pawar, P. S. L. S. N. (2021). Review of sliding mode based control techniques for control system applications. *International Journal of Dynamics and Control*, *9*(1), 363–378. https://doi.org/10.1007/s40435-020-00638-7

Hafez, A. T., Sarhan, A. A., & Givigi, S. (2019). Brushless DC motor speed control based on advanced sliding mode control (SMC) techniques. *SysCon 2019 - 13th Annual IEEE International Systems Conference, Proceedings*, 1–6. https://doi.org/10.1109/SYSCON.2019.8836754

Hu, H., Cai, N., Cui, L., Ren, Y., & Yu, W. (2017). A neural network–based sliding mode controller of folding-boom aerial work platform. *Advances in Mechanical Engineering*, *9*(10), 1–9. https://doi.org/10.1177/1687814017720876

Hung, J. Y., Gao, W., & Hung, J. C. (1993). Variable Structure Control: A Survey. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, *40*(1), 2–22.

Ismail, Z., Varatharajoo, R., & Chak, Y. C. (2020). A fractional-order sliding mode control for nominal and underactuated satellite attitude controls. *Advances in Space Research*, *66*(2), 321–334. https://doi.org/10.1016/j.asr.2020.02.022

JKD Power and Energy Solutions. (2021). *Second-order Sliding mode control*. MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/75115-second-order-sliding-mode-control

Jones, E. (2018). *DC Motors Explained: Brushed vs Brushless*. GALCO. https://blog.galco.com/dc-motors-explained-brushed-vs-brushless/

JyotiRath, J., Saha, S., & Ikkurti Prasad, H. (2012). Effective Speed Control in 3-Phase BLDC Motor by Reaching Law based Sliding Mode Technique. *International Journal of Computer Applications*, *43*(16), 25–32. https://doi.org/10.5120/6189-8675

Kojima, H., & Trivailo, P. M. (2019). Adaptive time-delay estimated sliding mode control for a bias momentum satellite with two reaction wheels. *Transactions of the Japan Society for Aeronautical and Space Sciences*, *62*(4), 236–245. https://doi.org/10.2322/tjsass.62.236

Kypuros, J. (2013). System Dynamics and Control with Bond Graph Modeling. In *System Dynamics and Control with Bond Graph Modeling*. CRC Press, Taylor & Francis Group. https://doi.org/10.1201/b14676

Li, Y.-R., & Peng, C. (2021). Super-Twisting Sliding Mode Control Law Design for Attitude Tracking Task of a Spacecraft via Reaction Wheels. *Mathematical Problems in Engineering*, *2021*, 6644033. https://doi.org/10.1155/2021/6644033

Lian, J., Zhao, J., & Dimirovski, G. M. (2010). Integral Sliding Mode Control for a Class of Uncertain Switched Nonlinear Systems. *European Journal of Control*, *16*(1), 16–22. https://doi.org/https://doi.org/10.3166/ejc.16.16-22

Lin, W.-B., & Chiang, H.-K. (2013). Super-Twisting Algorithm Second-Order Sliding Mode Control for a Synchronous Reluctance Motor Speed Drive. *Mathematical Problems in Engineering*, *2013*, 632061. https://doi.org/10.1155/2013/632061

Liu, Jianxing, Luo, W., Gao, Y., Yin, Y., & Sun, G. (2020). Sliding mode control of grid-connected power converters for microgrid applications. In *Distributed Control Methods and Cyber Security Issues in Microgrids* (pp. 3–27). Elsevier Inc. https://doi.org/10.1016/b978-0-12-816946-9.00001-3

Liu, Jinkun. (2017). Sliding mode control based on LMI. In Jinkun Liu (Ed.), *Sliding Mode Control Using MATLAB*. Academic Press. https://doi.org/10.1016/b978-0-12-802575-8.00005-9

Liu, R., Gao, X., & Zeng, J. (2019). Neural network identification and sliding mode control for hysteresis nonlinear system with backlash-like model. *Complexity*, *2019*. https://doi.org/10.1155/2019/4949265

Machado, J. A. T., & Cunha, V. M. R. (2021, May). Bond Graph Modeling and Simulation Using 20-sim. *An Introduction to Bond Graph Modeling with Applications*, *35*, 171–194. https://doi.org/10.1201/9781003057741-8

Maghfiroh, H., Sujono, A., Ahmad, M., & Apribowo, C. H. B. (2020). Basic Tutorial on Sliding

Mode Control in Speed Control of DC-motor. *Journal of Electrical, Electronic, Information, and Communication Technology*, *2*(1). https://doi.org/10.20961/jeeict.2.1.41354

MaheswararaoCh., U., Kishore Babu, Y. S., & Amaresh, K. (2011). Sliding mode speed control of a DC Motor. *Proceedings - 2011 International Conference on Communication Systems and Network Technologies, CSNT 2011*, *1*, 387–391. https://doi.org/10.1109/CSNT.2011.86

Mahmoud, M. S. (2018). Chapter 5 - Advanced Control Design. In M. S. Mahmoud (Ed.), *Advanced Control Design with Application to Electromechanical Systems* (pp. 177–253). Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/B978-0-12-814543-2.00005-9

Martínez, C. M., & Cao, D. (2019). Integrated energy management for electrified vehicles. In C. M. Martínez & D. Cao (Eds.), *Ihorizon-Enabled Energy Management for Electrified Vehicles* (pp. 15–75). Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/B978-0-12-815010-8.00002-8

Mathworks. (2017). *Shallow Networks for Pattern Recognition, Clustering and Time Series - MATLAB & Simulink - MathWorks France*. Mathworks Inc. https://fr.mathworks.com/help/deeplearning/gs/shallow-networks-for-pattern-recognition-clustering-and-time-series.html

Mathworks Inc. (2018). *Multilayer Neural Networks and Backpropagation Training*. Mathworks Inc. https://www.mathworks.com/help/deeplearning/ug/multilayer-neural-networks-and-backpropagation-training.html

Mehmood, J., Abid, M., Khan, M. S., & Khan, A. Q. (2020). Design of Sliding Mode Control for a Brushless DC Motor. *Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020*, 1–5. https://doi.org/10.1109/INMIC50486.2020.9318215

Mehrjardi, M. F., Sanusi, H., & Ali, M. A. M. (2015). Developing a proposed satellite reaction wheel model with current mode control. *International Conference on Space Science and Communication, IconSpace*, *2015-Septe*, 410–413. https://doi.org/10.1109/IconSpace.2015.7283839

Monteiro, J. R. B. A., Oliveira, C. M. R., & Aguiar, M. L. (2015). Sliding mode control of brushless DC motor speed with chattering reduction. *IEEE International Symposium on Industrial Electronics*, *2015-Septe*, 542–547. https://doi.org/10.1109/ISIE.2015.7281525

Morfin, O. A., Castañeda, C. E., Valderrabano-Gonzalez, A., Hernandez-Gonzalez, M., & Valenzuela, F. A. (2017). A real-time SOSM super-twisting technique for a compound DC

motor velocity controller. *Energies*, *10*(9), 1–18. https://doi.org/10.3390/en10091286

Murugesan, S. (1981). An Overview of Electric Motors for Space Applications. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, *IECI-28*(4), 260–265. https://doi.org/10.1109/TIECI.1981.351050

Panchade, V. M., Waghmare, L. M., Patre, B. M., & Bhogle, P. P. (2007). Sliding mode control of DC drives. *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, 1576–1580. https://doi.org/10.1109/ICMA.2007.4303784

Parma, G. G., de Menezes, B. R., & Braga, A. P. (1999). Neural networks learning with sliding mode control: the sliding mode backpropagation algorithm. *International Journal of Neural Systems*, *9*(3), 187–193. https://doi.org/10.1142/S0129065799000174

Perruquetti, W., & Barbot, J.-P. (2002). Sliding Mode Control In Engineering. In *Sliding Mode Control In Engineering*. https://doi.org/10.1201/9780203910856

Pillay, P., & Krishnan, R. (1989). Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part II: The Brushless DC Motor Drive. *IEEE Transactions on Industry Applications*, *25*(2), 274–279. https://doi.org/10.1109/28.25542

Pukdeboon, C., Zinober, A. S. I., & Thein, M. W. L. (2010). Quasi-continuous higher order sliding-mode controllers for spacecraft-attitude-tracking maneuvers. *IEEE Transactions on Industrial Electronics*, *57*(4), 1436–1444. https://doi.org/10.1109/TIE.2009.2030215

Qiao, J., Li, Z., Xu, J., & Yu, X. (2020). Composite Nonsingular Terminal Sliding Mode Attitude Controller for Spacecraft with Actuator Dynamics under Matched and Mismatched Disturbances. *IEEE Transactions on Industrial Informatics*, *16*(2), 1153–1162. https://doi.org/10.1109/TII.2019.2936172

Rakhonde, S., & Kulkarni, V. (2018). Sliding mode controller (SMC) governed speed control of DC motor. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings*, *Ci*, 1657–1662. https://doi.org/10.1109/RTEICT42901.2018.9012572

Razmi, H., & Afshinfar, S. (2019). Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerospace Science and Technology*, *91*, 12–27. https://doi.org/10.1016/j.ast.2019.04.055

Roving, L., Facility, A. X. A., & NASA. (1991). *Selection of Electric Motors for Aerospace Applications*. Current. http://llis.nasa.gov/lesson/893

Sabanovic, A., Fridman, L., & Spurgeon, S. (2011). Variable structure systems with sliding modes in motion control - A survey. *IEEE Trans. Ind. Inf. IEEE Transactions on Industrial Informatics*, *7*(2), 212–223.

Samantaray, J., & Chakrabarty, S. (2020). Discrete Time Sliding Mode Control. In C. Volosencu, A. Saghafinia, X. Du, & S. Chakrabarty (Eds.), *Control Theory in Engineering [Working Title]*. IntechOpen. https://doi.org/10.5772/intechopen.91245

Shtessel, Y., Edwards, C., Fridman, L., & Levant, A. (2014). *Sliding Mode Control and Observation*. Springer New York : Imprint : Birkhauser. http://link.springer.com/openurl?genre=book&isbn=978-0-8176-4892-3

Sidi, M. J. (2014). Spacecraft dynamics and control: A practical engineering approach. In *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815652

Skruch, P., & Długosz, M. (2019). Design of terminal sliding mode controllers for disturbed non-linear systems described by matrix differential equations of the second and first orders. *Applied Sciences (Switzerland)*, *9*(11). https://doi.org/10.3390/app9112325

Sowjanya, P., & Tarakalyani, S. (2013). *PI AND SLIDING MODE CONTROL FOR PERMANENT MAGNET BRUSHLESS DC MOTOR*. *1*, 497–502.

Spurgeon, S. (2014). Sliding mode control : a tutorial. *2014 European Control Conference, ECC 2014*, *1*, 2272–2277. https://doi.org/10.1109/ECC.2014.6862622

Stapel, E. (2021). *Cramer's Rule*. Purplemath. https://www.purplemath.com/modules/cramers.htm

Suleiman, H. U., Murazu, M. B., Zarma, T. A., Salawudeen, A. T., Thomas, S., & Galadima, A. A. (2018). Methods of chattering reduction in sliding mode control: A case study of ball and plate system. *IEEE International Conference on Adaptive Science and Technology, ICAST*, *2018-Augus*. https://doi.org/10.1109/ICASTECH.2018.8506783

Susperregui, A., Tapia, G., & ItsasoMartinez, M. (2011). Sensorless First- and Second-Order Sliding-Mode Control of a Wind Turbine-Driven Doubly-Fed Induction Generator. In A. Bartoszewicz (Ed.), *Sliding Mode Control* (pp. 109–132). https://doi.org/10.5772/632

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications* (2010 Draft). Springer.

Takehana, R., & Uchiyama, K. (2017). Attitude Controller Design for a Small Satellite Using Spherical Reaction Wheel System. *2017 11th Asian Control Conference (ASCC) Gold Coast Convention Centre, Australia*.

The Economic Times. (2021, August). Space-Technology: Definition of "Reaction Wheels." *The Economic Times E-Paper*. https://economictimes.indiatimes.com/definition/reaction-wheels

Theraja, B. L., & Theraja, A. K. (2005). *A Textbook of Electrical Technology* (6th ed.). S. Chand & Company Ltd.

Tiwari, P. M., Janardhanan, S., & un-Nabi, M. (2018). Spacecraft Anti-Unwinding Attitude Control Using Second-Order Sliding Mode. *Asian Journal of Control*, *20*(1), 455–468. https://doi.org/https://doi.org/10.1002/asjc.1601

Trentin, J. F. S., Da Silva, S., De Souza Ribeiro, J. M., & Schaub, H. (2020). Inverted Pendulum Nonlinear Controllers Using Two Reaction Wheels: Design and Implementation. *IEEE Access*, *8*, 74922–74932. https://doi.org/10.1109/ACCESS.2020.2988800

Truong, H. V. A., Tran, D. T., & Ahn, K. K. (2019). A neural network based sliding mode control for tracking performance with parameters variation of a 3-DOF manipulator. *Applied Sciences (Switzerland)*, *9*(10). https://doi.org/10.3390/app9102023

Tsypkin, Y. Z. (2003). *Relay Control Systems*. Cambridge University Press. https://doi.org/10.1142/9789812564436_0003

Utkin, V., Guldner, J., & Shi, J. (2017). Sliding mode control in electro-mechanical systems, second edition. In *Sliding Mode Control in Electro-Mechanical Systems, Second Edition* (Second Edi). Taylor & Francis Group LLC. https://doi.org/10.1201/9781420065619

Vo, A. T., & Kang, H.-J. (2018). An Adaptive Neural Non-Singular Fast-Terminal Sliding-Mode Control for Industrial Robotic Manipulators. *Applied Sciences*, *8*(12). https://doi.org/10.3390/app8122562

Vo, A. T., & Kang, H. J. (2019). An Adaptive Terminal Sliding Mode Control for Robot Manipulators with Non-Singular Terminal Sliding Surface Variables. *IEEE Access*, *7*, 8701–8712. https://doi.org/10.1109/ACCESS.2018.2886222

von Ellenrieder, K. D. (2021). *Sliding Mode Control* (pp. 489–526). Youtube. https://doi.org/10.1007/978-3-030-75021-3_12

Wang, S., Cao, Y., Huang, T., Chen, Y., Li, P., & Wen, S. (2020). Sliding mode control of neural networks via continuous or periodic sampling event-triggering algorithm. *Neural Networks*, *121*, 140–147. https://doi.org/10.1016/j.neunet.2019.09.001

Wu, L., Shi, P., & Su, X. (2014a). *Sliding Mode Control of Uncertain Parameter-Switching Hybrid Systems* (First). Wiley & Sons, Ltd.

Wu, L., Shi, P., & Su, X. (2014b). Introduction. In *Sliding Mode Control of Uncertain Parameter-Switching Hybrid Systems* (pp. 1–31). John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781118862612.ch1

Yan, X., & Liu, J. (2010). A novel sliding mode control for BLDC motor network control system. *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, *2*, 0–4. https://doi.org/10.1109/ICACTE.2010.5579156

Yeh, F. K. (2010). Sliding-mode adaptive attitude controller design for spacecrafts with thrusters. *IET Control Theory and Applications*, *4*(7), 1254–1264. https://doi.org/10.1049/iet-cta.2009.0026

Zhang, Peixi, Qiao, J., Guo, L., & Li, W. (2017). Sliding mode friction observer based control for flexible spacecraft with reaction wheel. *IET Control Theory and Applications*, *11*(8), 1274–1281. https://doi.org/10.1049/iet-cta.2016.0802

Zhang, Pengfei, Wu, Z., Dong, H., Tan, M., & Yu, J. (2020). Reaction-Wheel-Based Roll Stabilization for a Robotic Fish Using Neural Network Sliding Mode Control. *IEEE/ASME Transactions on Mechatronics*, *25*(4), 1904–1911. https://doi.org/10.1109/TMECH.2020.2992038

GLOSSARY

CMG – Control Moment Gyroscope

Conv – Conventional

DC – Direct Current

DoF – Degree of Freedom

EMF – Electromotive Force

FOSMC – First Order Sliding Mode Control

GUI – Graphical User Interface

MSE – Mean Square Error

NN – Neural Network

PID – Proportional Integral Derivative

PWM – Pulse Width Modulated

RPM – Revolution Per Minute

S – Sliding Variable

SMC – Sliding Mode Control

SOSMC – Second Order Sliding Mode Control

STA – Super Twisting Algorithim

STSMC – Super Twisting Sliding Mode Control

TVC – Thrust Vector Control

VSC – Variable Structure Control

VSS – Variable Structure Systems

APPENDIX

Appendix A: Bond Graph Output (20-Sim Model Equations)

Bond Graph Output (20-sim Model Equations)

Continuous-time equations

==========

model: C:\Users\Idoko\Documents\Aerospace Project\20-Sim\bond1.emx

date: July 19, 2021

time: 8:27:42 PM

static equations:

Se\p.e = Se\effort;

dynamic equations:

I\p.f = I\state / I\i;

I2\p.f = I2\state / I2\i;

GY\p1.e = GY\r * I2\p.f;

GY\p2.e = GY\r * I\p.f;

R\p.e = R\r * I\p.f;

R2\p.e = R2\r * I2\p.f;

I\p.e = Se\p.e - (GY\p1.e + R\p.e);

I2\p.e = GY\p2.e - R2\p.e;

system equations:

I\state = int (I\p.e, I\state_initial);

I2\state = int (I2\p.e, I2\state_initial);

removed equations:

OneJunction1\p2.f = I\p.f;

OneJunction1\p4.e = Se\p.e;

OneJunction2\p2.f = I2\p.f;

OneJunction1\p1.f = I\p.f;

OneJunction1\p3.f = I\p.f;

OneJunction1\p4.f = I\p.f;

OneJunction2\p1.f = I2\p.f;

OneJunction2\p3.f = I2\p.f;

OneJunction1\flow = I\p.f;

OneJunction2\flow = I2\p.f;

GY\p2.f = I2\p.f;

GY\p1.f = I\p.f;

R\p.f = I\p.f;

Se\p.f = I\p.f;

R2\p.f = I2\p.f;

Se\flow = I\p.f;

OneJunction2\p1.e = GY\p2.e;

OneJunction1\p1.e = GY\p1.e;

OneJunction1\p3.e = R\p.e;

OneJunction2\p3.e = R2\p.e;

OneJunction1\p2.e = I\p.e;

OneJunction2\p2.e = I2\p.e;

Output:

The model bond1.emx contains:

6 submodels

9 equations

7 variables

2 independent states

The model has 0 errors and 0 warnings.

Model processing succeeded

Appendix B: Neural Network MATLAB Script

```
function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 13-Aug-2021 00:24:15.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = Qx2 matrix, input #1
% and returns:
%   y = Qx1 matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [-4.24607313658014;-4.24607313658014];
x1_step1.gain = [0.230472061515611;0.230472061515611];
x1_step1.ymin = -1;

% Layer 1
b1 = [2.8450645441309809769;7.5865400847409727447;-1.490220377791934947;
        -2.1106109810949855721;-5.1314057546887692496;5.2829968240878342201;
        -1.5629721366917914693;-2.9094697482702587266;1.3639256838911519853;-
        2.1393711099708538903;3.0507741746306704655;-
        0.86229253199002353636;0.53082538093999187545;1.910330946863463053;-
        5.7459966341362482822;-
        1.5192660214949655906;2.0001123041575752382;1.209506696353107813;-
        4.3606018243287598679;1.8873158834182104382;-1.1800618228838888335;-
        1.6627803488562373779;-1.3789795957849004804;0.76195055559493762853;-
        1.0718754511915329353;1.1897743028120859599;-0.75689996132557169073;-
        1.7960296390280339196;-6.346045376296460816;-
        0.21225040753768459334;7.3001302482210501665;7.5862871576324497624];
IW1_1 = [-6.7906958213335375518 -1.372503952786809478;
        -3.6787910904519875999 6.911863575500001744;-1.8938884461802747428
        -0.003144812783958081479 2;6.8490811898612387054 -
        5.439938046051679521;7.5356390578619896203 -1.7832145972847299742;-
        6.1956659788471171169 4.9553217911120119865;11.686769769072254732
        2.1835839041314590858;6.5315206237798300037 -5.9659051290635689568;-
        4.8161896416652627551 -7.398166472675446137;6.9143056118977312963
        1.4703882732322166582;-6.0279458103602490127 -
        0.75065110179553728997;3.6913633975713646684
        0.86549093283525047049;0.17368236690127988875
        30.360289558024163625;2.9904370691377177849 -
        8.2912820717668935799;6.660415730556291436 7.9195317754692462486;-
        22.695158945786960913 -56.918516953845774253;-8.9594633667445577885 -
        2.0434760654439720895;14.575121213733423176 1.9326765948890225477;-
        4.7322893887303658644 6.1997360657623623581;22.454109363506191244
```

```
        56.650136528602715202;17.177591025132162628 4.2784733079738446193;-
        22.124891909750761698 -55.489534590637362044;-0.28306621736359477426 -
        56.712304017185772409;0.20874180520363488922 36.80282166685854861;-
        0.35665212868237833321 -46.633909311073800552;8.8502121937779296701
        1.3285344328437111994;-7.6013395369057130679 -1.1765595135645512403;-
        2.5586552128193882361 7.5693609234335417213;-5.5600794610567971787
        5.6727862814670277913;-1.2478770837137687089
        0.98792210210805131432;5.2835237821030816008 -
        5.9747261782966889498;3.9080682328952494053 -7.12146354693334071581;

% Layer 2
b2 = 1.157758923744858226;
LW2_1 = [-1.3578571853246337309 1.1636792338663570767
        0.83222443458640649183 0.60780422503142128576 -0.27796900287886777692
        1.4618609612304152012 1.3148167120289540044 -1.7472092911649175218
        -0.17476344622148548558 3.5277384562065878804 -0.82786999345193146294 -
        13.085367627196092144 2.7822661483887571876 1.3217130923968285483 -
        0.005239972731996709748 6.5803569079304962131 -2.039777195930157383 -
        0.65204277961383283468 5.15575964839635148 -6.1123900662709642617
        0.37293088281643205217 -12.692714863802141423 2.8956479590003145752 -
        6.8985508240592592344 -7.1329512730493540573 -2.9490986236618677019 -
        5.8682938566612543596 1.6456120988569851171 0.26727084027188918736 -
        6.065403913898764543 0.064171627550855120381 -0.342267415298198540711;

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 0.00132521944743512;
y1_step1.xoffset = -1013.64906097319;

% ===== SIMULATION ========

% Dimensions
Q = size(x1,1); % samples

% Input 1
x1 = x1';
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
y1 = y1';
end

% ===== MODULE FUNCTIONS ========
```

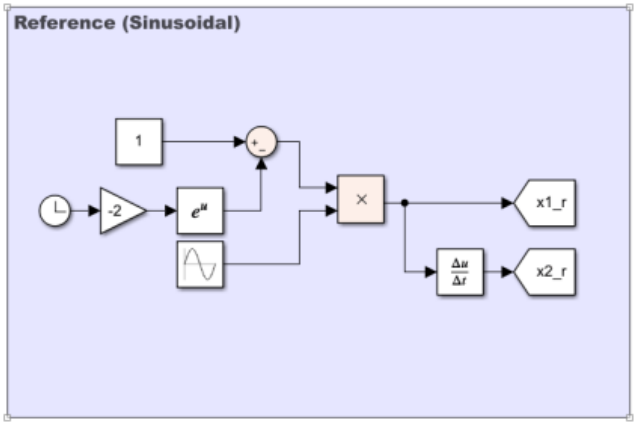
```
 % Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

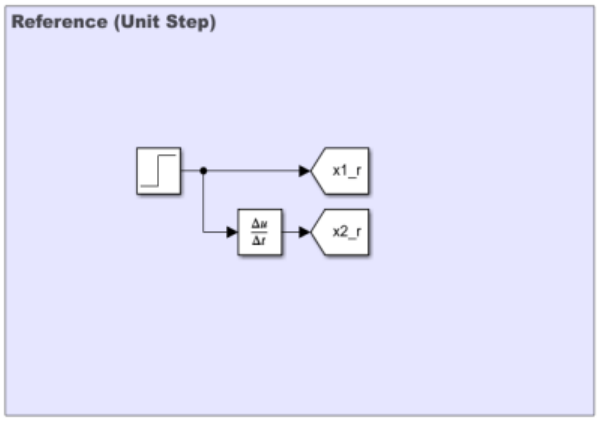
```
% This code shows the underlined script for the Neural Fitting App
%%%%%
% Part of this code was auto-generated
% This script assumes these variables are defined:
%
%   s1 - input data.
%   so - target data.

x = s1';
t = so';

% Select the training function to be employed as
% Levenberg-Marquardt backpropagation.
trainFcn = 'trainlm';

% Create a Fitting Network and choose a neuron size of 32
hiddenLayerSize = 32;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose actual Input and Output Pre/Post-Processing Functions
%
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% process data into three group for
% Training, Validation, and Testing
net.divideFcn = 'dividerand';  % Divide data in a random form
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 60/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
```

```
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y); performance = perform(net,t,y) ;

% Recalculate Training, Validation and Test Performance
trainTargets = t.* tr.trainMask{1};
valTargets = t.* tr.valMask{1};
testTargets = t.* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the outcome of the Network
view(net)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

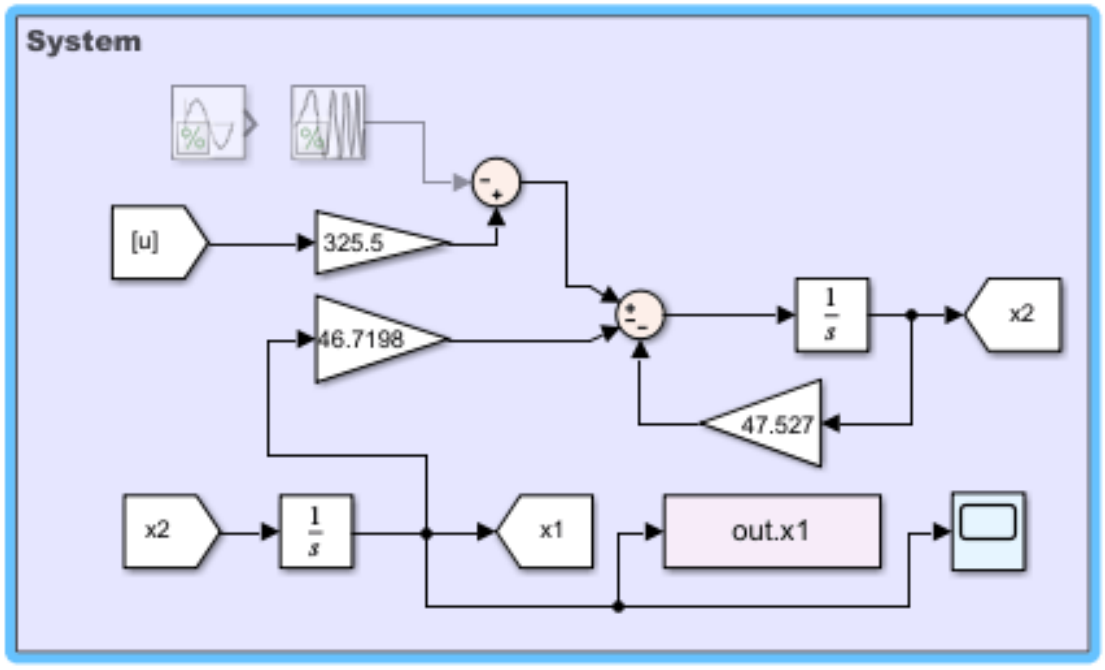
Appendix C: Controller Simulink Setup

C1: Reference

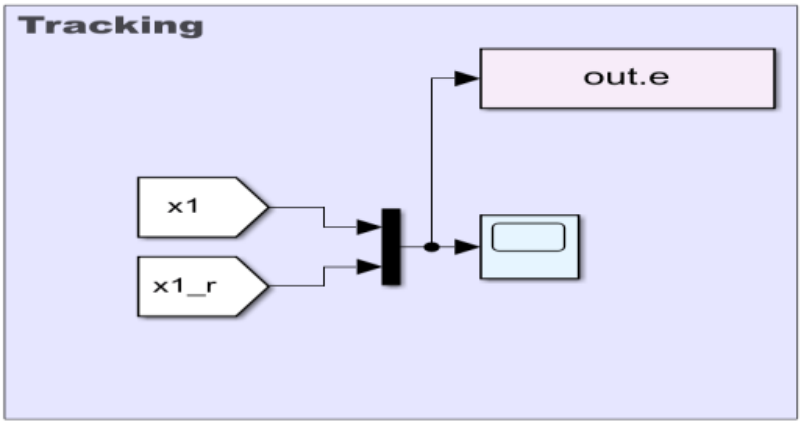




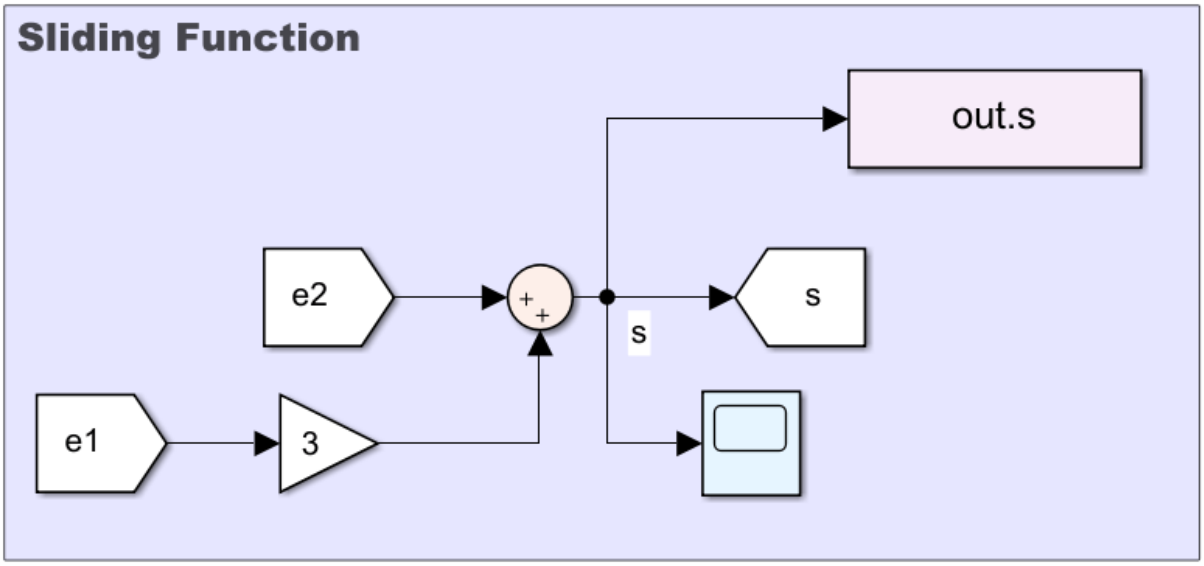
C2: System



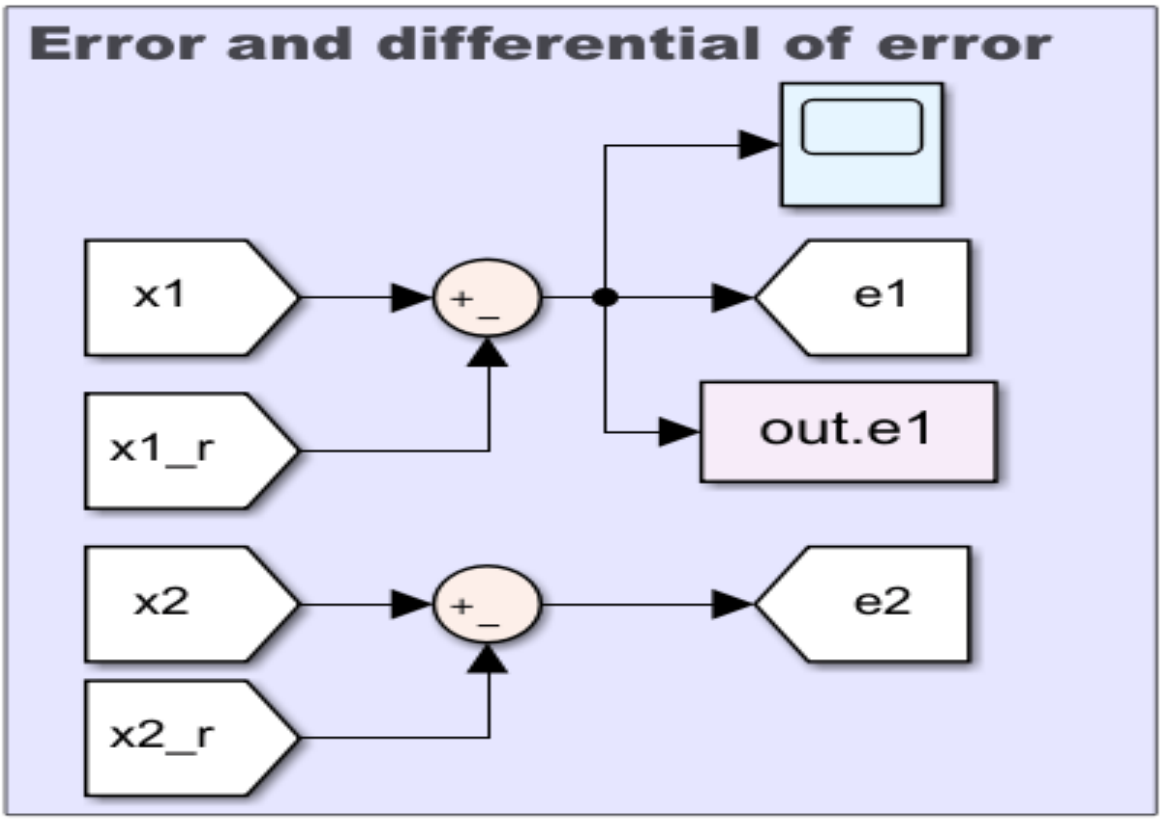
C3: Tracking Setup



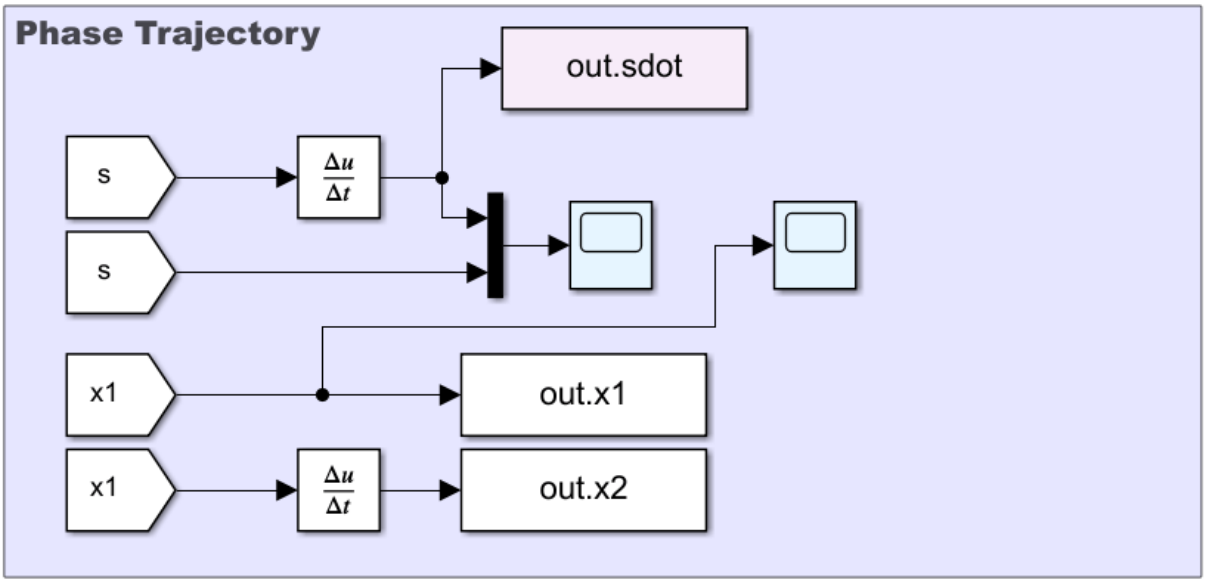
C4: Sliding Function



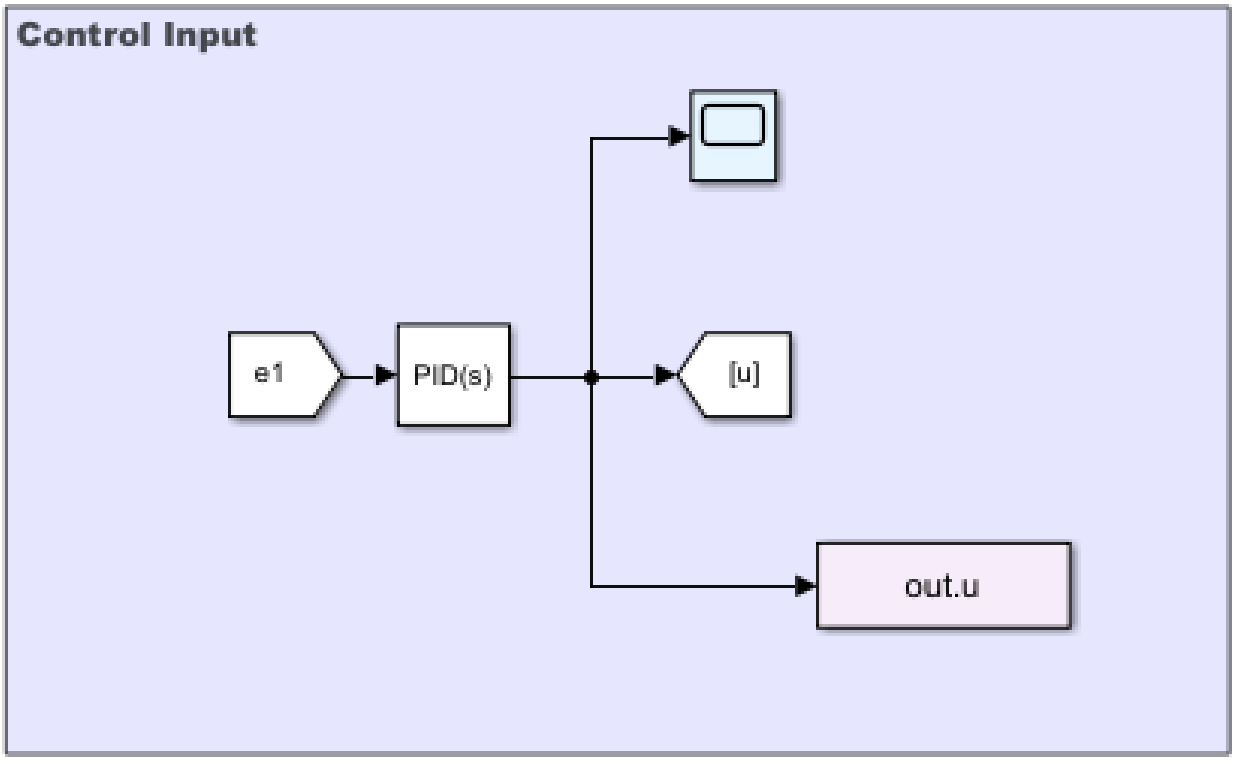
C5: Error and differential of error



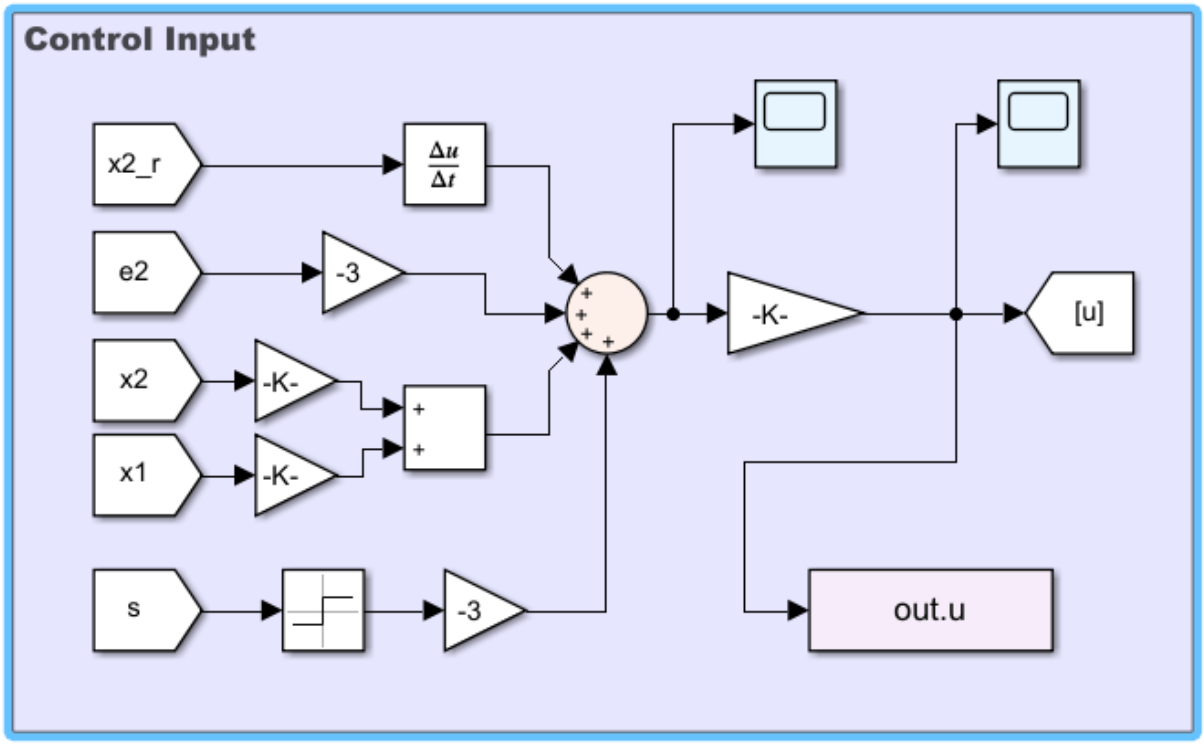
C6: System



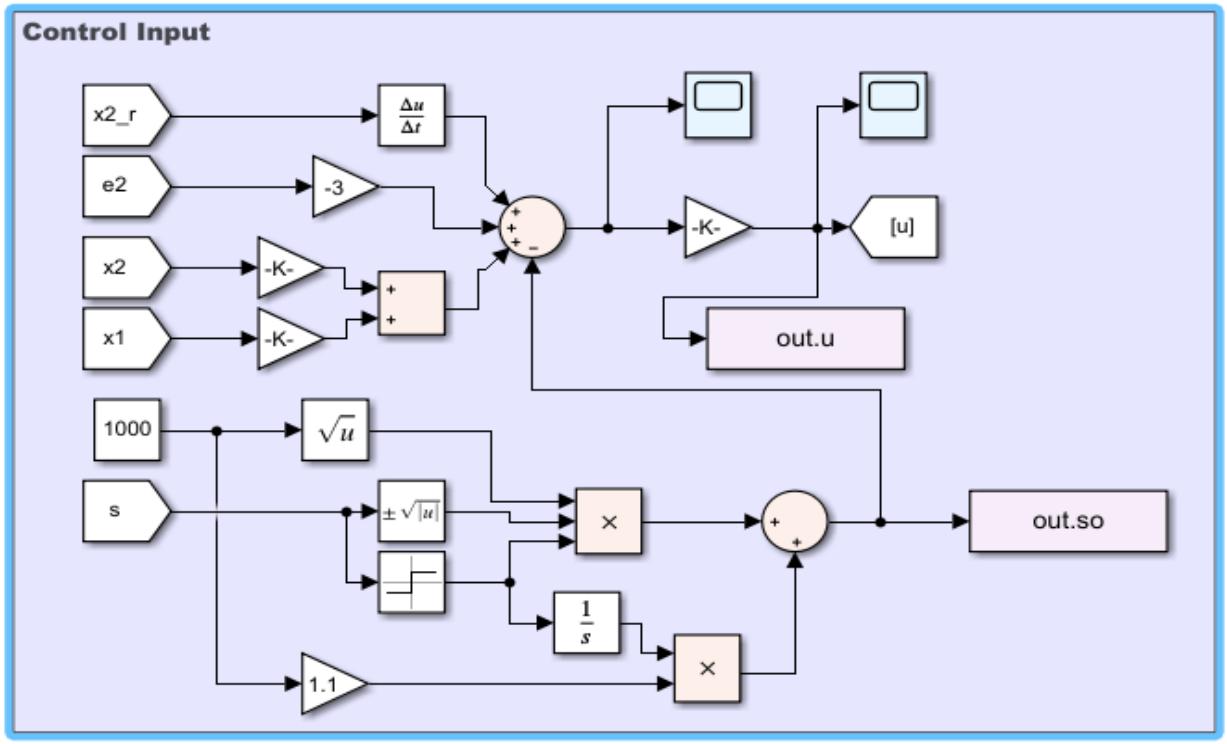
C7: PID Control Input



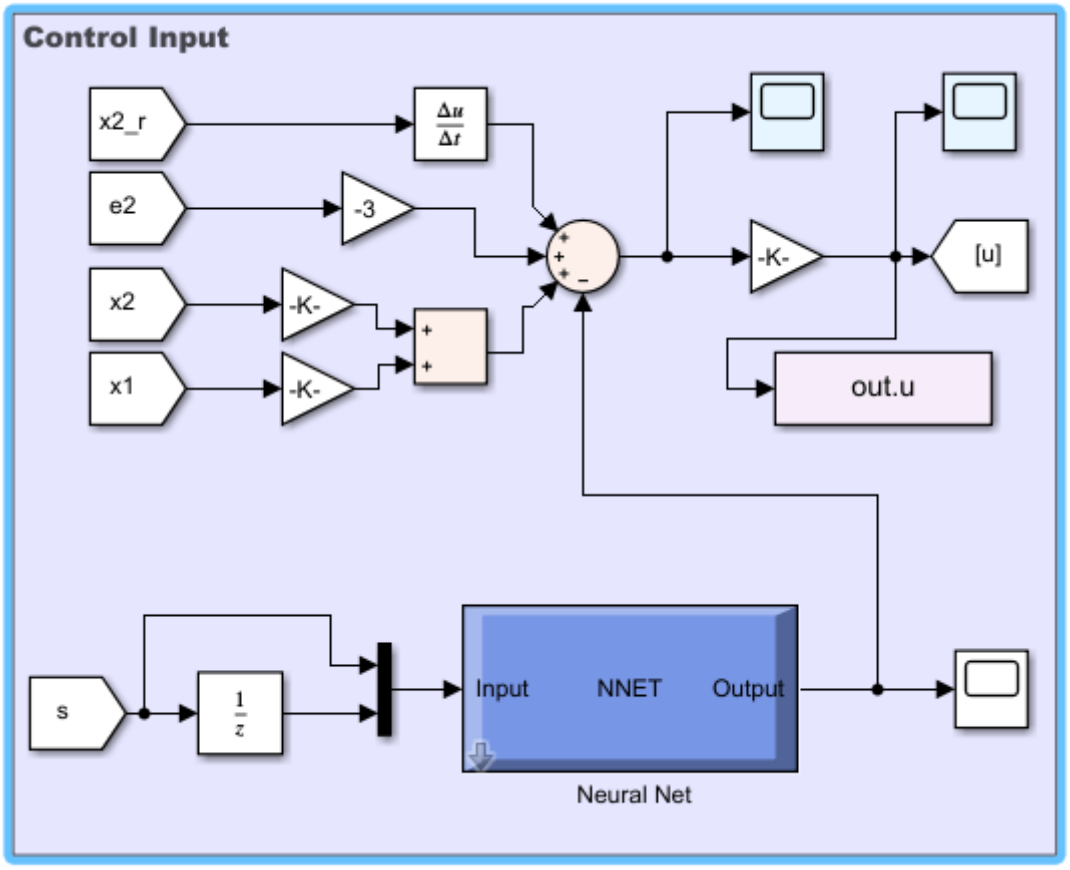
C8: Conventional Sliding Mode Control Input



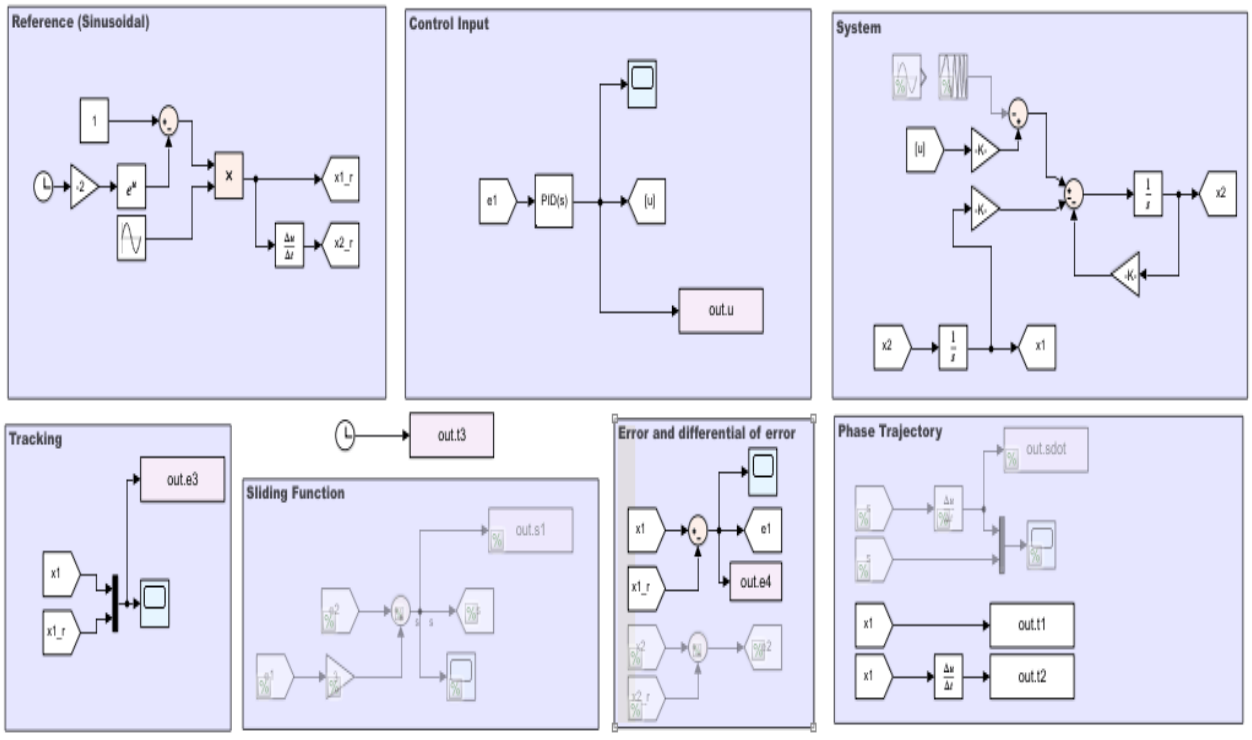
C9: Super-Twisting Sliding Mode Control Input



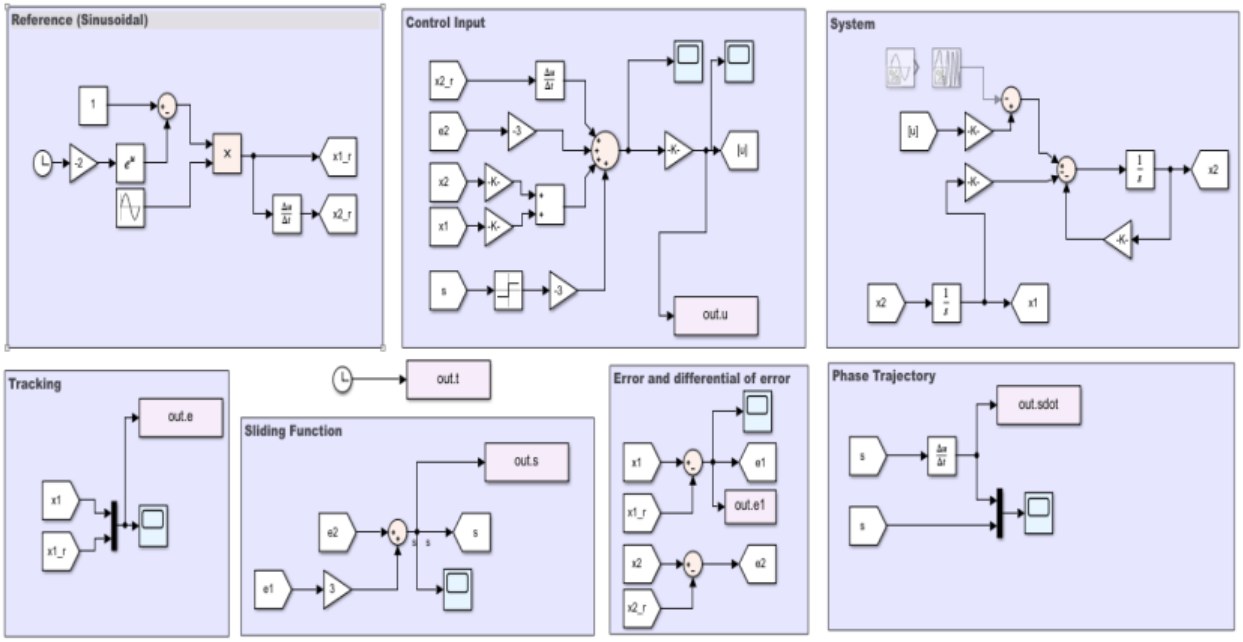
C10: Neural-Tuned Super-Twisting Sliding Mode Control Input



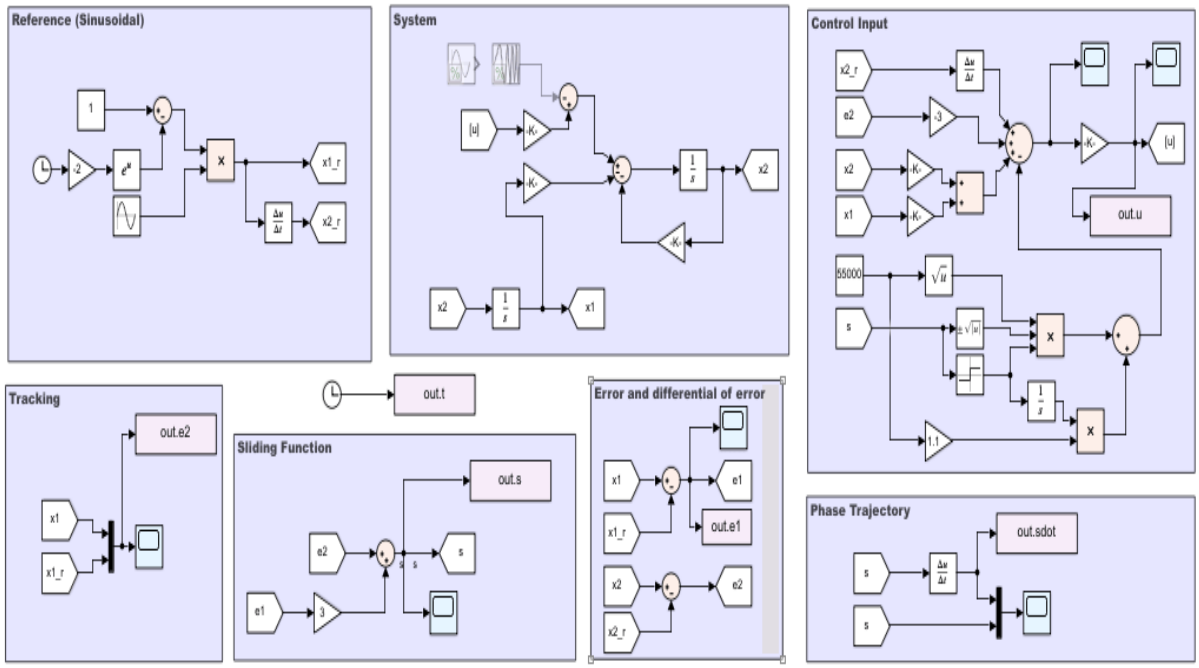
C11: Complete PID Simulation Setup



C12: Complete Conventional SMC Simulation Setup



C13: Complete Super-Twisting SMC Simulation Setup



C14: Complete Neural-tuned SMC Simulation Setup