

Original Research Paper

Trajectory Tracking Control of UR5 Robot Manipulator Using Fuzzy Gain Scheduling Terminal Sliding Mode Controller

¹Andualem Welabo and ²Gebremichael Tesfamariamr

¹Electromechanical (mechatronics) engineering, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia

²Control Engineering, Mekelle University, Mekelle, Ethiopia

Article history

Received: 14-03-2020

Revised: 08-05-2020

Accepted: 22-07-2020

Corresponding Author:

Andualem Welabo
Electromechanical
(mechatronics) engineering,
Addis Ababa Science and
Technology University, Addis
Ababa, Ethiopia
E-mail: anduw7585@gmail.com

Abstract: In dealing with the trajectory tracking control of robotic manipulator many scholars have had to work in a sliding mode controller. However, SMC has two major limitations one is chattering the other is asymptotical convergence. The chattering problem occurs in traditional SMC due to the switching function of the discontinuous controller and the constant gain parameter of K . Though in this study a Fuzzy Gain Scheduling Terminal Sliding Mode (FGSTSM) controller with a hyperbolic tangent function instead of signum function, is proposed and applied for tracking control of the UR5 robot manipulator. Moreover, the mathematical model of the UR5 robot manipulator using the Newton-Euler algorithm was developed on Maple software of version Maple 18.2. Hence the trajectory tracking control of the UR5 robot manipulator simulation was conducted on MATLAB/SIMULINK of version Matlab R2016a. Then for simulation purposes, a curve (arc) was taken as the desired trajectory for the robot manipulator to track and finally the result is compared with the conventional SMC controller.

Keywords: UR5 Robot, Fuzzy Gain Scheduling, Terminal Sliding Mode Controller, Trajectory Tracking

Introduction

Robots have been traditionally employed in closed environments separate from human beings because they have been heavy, bulky and dangerous. In recent years, a lot of development has been made to robots that are safe enough to interact with human beings. In the future, one might envision service robots co-existing with humans. Such robots are known as Collaborative Robots (COBOTS). Universal Robot is the most prominent type of manipulator manufacturer in the collaborative robot world. A series of robotic manipulators including UR3, UR5 and UR10 were developed by the universal robot and now they are widely used in different applications. For instance, this robot is widely used in industry, in academic and now researchers tried to implement it in hospitals for COBOT surgery.

Trajectory tracking control of robot manipulators plays an important role in the control theory due to the wide development of engineering practice. The trajectory tracking control or the motion control of the robotic manipulators have a very significant influence on the overall manipulator performance. Robotic manipulators

are always exposed to unmodeled dynamics including unknown gravitational and friction torque, disturbances and uncertain inertia. All these un-modeled robotic manipulator parametric uncertainties are addressed by the kinematic and dynamic modeling of the robotic system.

Literature Review

Though to control such a highly disturbed, MIMO, coupled and non-linear dynamic system demand a controller which is robust against uncertainties, fast response and simple for design. Liu and Zhang (2012) have introduced a fuzzy sliding mode controller first for two Degrees Of Freedom (2 DOF) robot arm and then tested on 4 DOF SCARA. This controller first developed for under actuated SCARA robot. Though this article, used a sliding mode controller to improve the control accuracy and the robustness of the robotic manipulator and the fuzzy logic control was used to approximate various uncertainties and to alleviate the chattering effect of the SMC.

A simple fuzzy sliding mode control to achieve the best trajectory tracking for the robot manipulator was reported by (Soltanpour *et al.*, 2014). The principal of

the proposed method by the author, applied the feedback linearization technique, as a result, the known dynamics of the robot's manipulator is removed; and then, to overcome the remaining uncertainties, a classic sliding mode control was used.

3-DOF spatial manipulator trajectory tracking control was reported by (Wan *et al.*, 2016). To realize a high accuracy trajectory tracking a novel sliding mode controller with PID was proposed in this article. This paper significantly attunes the chattering effect of the Sliding mode controller by introducing the switching function, however, the system stability in both globally and asymptotically analyzed theoretically. PSO based neuro-fuzzy sliding mode control strategy for two Degrees Of Freedom (2DOF) rigid robot manipulator was developed by (Vijay and Debashisha Jena, 2017). The formulation of a coupling Artificial Neuro-Fuzzy Inference System (ANFIS) with Sliding Mode Control (SMC) approach was the main concern of the paper. In the beginning, SMC with Proportional Integral Derivative (PID) sliding surface is adapted to control the robot manipulator, but the sliding surface parameters are obtained by minimizing quadratic performance indices using Particle Swarm Optimization (PSO).

Moreover, a Fractional-Order Nonsingular Terminal Sliding Mode (CFONTSM) based on Time Delay Estimation (TDE) approach was presented by (Liu and Wang, 2012) for practical tracking control robot manipulator. This control scheme doesn't require detailed information about the robot dynamics, leading to a better model-free nature of TDE, however, this paper considers only lumped uncertainties, under this condition FONTSM surface and fast-TSM-type reaching law ensure fast convergence and high tracking precision. Simulation and experiment was carried out using two degrees of freedom robot arm for the purpose of testing and validating the proposed controller. An adaptive robust sliding mode tracking controller for a 6 degree-of-freedom industrial assembly robot with parametric uncertainties and external disturbances was developed by (Pan *et al.*, 2018). Then the proposed dynamic control algorithms have been designed and simulated in MATLAB toolboxes. However, the proposed control of this paper was compared with a commonly used Proportional Derivative (PD) controller with fixed control parameters and a conventional sliding mode controller without adaptive and robust terms.

Task space absolutely continuous Terminal Sliding Mode (TSM) controllers with the finite time convergence property of the trajectory tracking by n-DOF rigid robotic manipulator were proposed by (Galicki, 2016). The terminal sliding mode controller on this research, makes it possible to simultaneously apply both the first and second-order sliding mode controller with their advantages. This work addresses the problem of the accurate task space control subject to finite-time convergence and dynamic equations uncertainty of a rigid robotic manipulator was also considered.

Having the good characteristics of the sliding mode, researchers have focused on finding a new continuous SMC controller that can be converged in finite time globally. The discontinuous terminal sliding mode controller has been developed by (Yu and Zhihong, 1998; 2002). However, this discontinuous TSM controller has been widely used for robotic manipulator applications. Nevertheless, this system is highly singular due to the fractional power in a negative error signal of the sliding variable of the TSM controller. To overcome the singularity problem related to the sliding function of the TSM controller, Nonsingular TSMC has developed by (Feng *et al.*, 2002). Nonetheless in this study to reduce the chattering of the discontinuous control a boundary layer approach was applied, however, the finite-time stability property was lost. Latter on a continuous terminal sliding mode controller was developed by (Yu *et al.*, 2005). This approach has tried to address global finite-time stability adequately. However in this study, a constant tuning switching gain has existed, this is another factor that contributes to the formulations of the chattering phenomenon.

Latter on a continuous terminal sliding mode controller was developed by (Yu *et al.*, 2005). This approach has tried to address global finite-time stability adequately. However in this study, a constant tuning switching gain has existed, this is another factor that contributes to the formulations of the chattering phenomenon. Regardless of VSS in the above reviewed articles most controllers have some major problems have been considered for this study, among those the chattering phenomenon, asymptotic convergence and a constant or a fixed tunable reaching control gain (K) are mentioned. So to meet the demand, many scholars have been developed different types of controller for robotic manipulator. Though to overcome the above main drawbacks of a continuous finite time terminal sliding mode controller, in this study FGSTSM controller is developed for trajectory tracking control of UR5.

Design, modeling and control of lightweight and flexible robotic manipulators recently become more attractive among the robotics community because lightweight and flexible robotic manipulators have a great potential in the design of high performance industrial robotic manipulators since they allow high-speed operation and low energy consumption. Thus, in this research, the kinematic and dynamic model analysis of the Universal Robot 5 (UR5) robotic manipulator is developed by using MATLAB and MAPLE software. Henceforth a fuzzy gain scheduling TSM controller is adopted for trajectory tracking problems. Finally, the trajectory tracking control of the UR5 robotic manipulator will be tested by simulating the robotic system using the FGSTSM controller and the result will be verified by comparing the result with other conventional SMC controllers in different scenarios.

This article is organized as follows: Section 2 discussed the development of the kinematic and dynamic model of the UR5 robot. Section 3 formulates the conventional and the proposed fuzzy gain scheduling terminal sliding mode controller. Section 4 in this section simulation results will be discussed. Section 5 will conclude the paperwork.

Mathematical Model of 6-DoF UR5

The science that deals with motion without considering the force causes the motion is known as kinematics. Hence, the study of the kinematics of robotic manipulators refers to all the geometrical and time-based properties of the motion. Therefore, a kinematic model of the 6-DOF UR5 robotic man manipulator firstly developed.

Kinematic Modeling of UR5

From the mechanical structure point of view, the UR5 robot manipulator is an open kinematic chain as shown in Fig. 1, which connected by 6 revolute joints.

Robot arm always expressed kinematically by giving the values of four quantities for each link. The first two quantities describe the link itself and the other two represent the link’s connection to a neighboring link. In the UR5 robot manipulator (i.e., all joints are revolute) the *joint variable* is θ , the other quantities are constant and called *link parameters*. Though, joint variables

and the link parameters of the UR5 robot arm from Fig. 2 as presented by (Kebria *et al.*, 2016) is summarized in Table 1. Therefore the final results of the individual homogeneous transformation matrix are obtained as A_i^{i-1} :

$$T_6^0 = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4 * A_6^5 \tag{1}$$

$$T_6^0 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The first 3x3 matrix represents orientation or the robotic manipulator, whereas the last column represents the position of the end-effector.

Table 1: The DH parameter values of the UR5 robot

LINK <i>l</i>	θ_l	d_l (m)	a_l (m)	α_l (rad)
1	θ_1	0.08916	0	$\pi/2$
2	θ_2	0.00000	-0.425	0
3	θ_3	0.00000	-0.392	0
4	θ_4	0.10915	0.000	$\pi/2$
5	θ_5	0.09465	0.000	$-\pi/2$
6	θ_6	0.0823	0.000	0



Fig. 1: Six DOF Universal Robot (UR5)

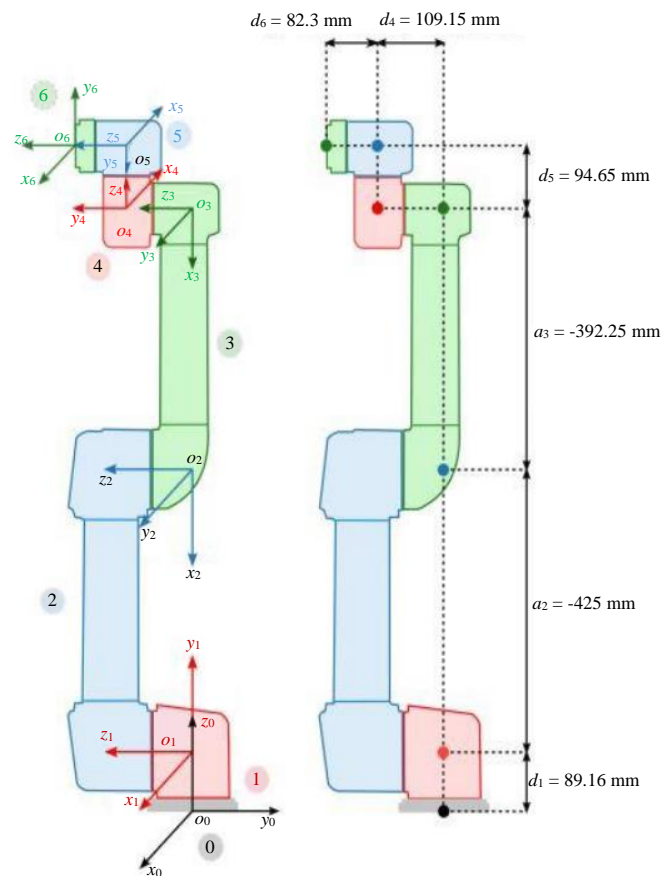


Fig. 2: The coordinate frame assignment at each joints and the DH parameters for each link of the UR5 robotic manipulator (Kufieta, 2014)

Dynamic Modeling of UR5 Robotic Manipulator

Modeling the dynamics of the robot manipulator means deriving a comprehensive relation between force and motion. The robot dynamic behavior can be expressed in terms of robot configuration and its derivatives in relation to the joint torques. This relation can be described by a set of differential equations, known as the equation of motion (dynamic model) which governs the dynamic response of the whole robotic system. There are two different algorithms, to develop the dynamic model of the robotic manipulator: The Newton-Euler and the Lagrange-Euler algorithm. This paper will drive the dynamic model of UR5 robot manipulator based on the Newton-Euler formulation.

Hence, the dynamic equation (equation of motion) develop the relationship between the joint torques τ , joints positions q , joint velocities \dot{q} and joint accelerations \ddot{q} of the robot manipulator. Therefore the general equation of motion expressed in the form:

$$\tau = M(q)\ddot{q} + C(\dot{q}, q) + g(q) \tag{3}$$

where, the matrix $M(q)$ is called the joint space inertial (mass) matrix, which gives the relationship between the joint force/torque and the resulting joints accelerations.

The vector $C(\dot{q}, q)$ is the centrifugal and Coriolis vectors of torques and the vector $g(q)$ is the vector of gravitational joint torques.

By rearranging the above equation of motion it is possible to find, the forward dynamics equation and expressed as:

$$\ddot{q} = M^{-1}(q)\tau - C(\dot{q}, q) - g(q) \tag{4}$$

Numerous literature on analytical mechanics have provides several approaches to develop the dynamic equation for a robot manipulator. These approaches include recursive Newton-Euler (Luh *et al.*, 1980), Euler-Lagrange

Paul (1981), Kane's equation (Kane and Levinson, 1983) and so on. Thus, in this paper the dynamic model of the manipulator formulated based on the recursive Newton-Euler method is established in (Sciavicco and Siciliano, 2012) and (Luh *et al.*, 1980) for the manipulators having revolute joints. Moreover, the dynamic formulation (the forward and backward recursion) which govern the robotic system computed by an automated software called Maple symbolically.

A Continuous Terminal Sliding Mode Controller Analysis

The non-linear controller provides an acceptable control performance with stability and robustness for a non-linear systems. Terminal Sliding mode control has the same robust properties like conventional sliding mode controller and which is one of a non-linear type controller and is generally a non-linear control scheme and recognized as the most powerful, robust and simple controller.

System Description and Problem Formulation

Therefore, to recall the dynamic model of a robotic manipulator (i.e., six degrees of freedom) is obtained in the previous section and can be expressed as:

$$M(q)\ddot{q} = \tau - [C(\dot{q}, q) + G(q)] \quad (5)$$

In Equation (3) (the manipulator dynamics) represents the dynamic behavior of the robotic manipulator, but the expression is highly non-linear, coupled and complex second-order system of the form:

$$\dot{x}_2(t) = f_i(x, t) + b_i(x, t)u_i(t) \quad (6)$$

where, $i = 1, 2, 3 \dots n$.

Let's consider x as a state variables by assigning:

$$q_i = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^T \quad (7)$$

$$\dot{q}_i = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4 \quad \dot{q}_5 \quad \dot{q}_6]^T \quad (8)$$

$$x_i = q_i; x_{i+p} = \dot{q}_i, i = 1, 2, \dots, p \quad (9)$$

Then, the state variables (state vectors) of the robotic manipulators can be defined in this form:

$$x_1 = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^T$$

$$x_2 = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4 \quad \dot{q}_5 \quad \dot{q}_6]^T \quad (10)$$

$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$

$$\ddot{q} = \dot{x}_2 = [\ddot{q}_1 \quad \ddot{q}_2 \quad \ddot{q}_3 \quad \ddot{q}_4 \quad \ddot{q}_5 \quad \ddot{q}_6]^T \quad (11)$$

where, $t \in R$ is the time, $x(t) \in R^{2n}$ is a state variables, $u(t) \in R^n$ is the control input of the system and they are defined as:

$$X(t) = [x_1(t), x_2(t), \dots, x_{2n}(t)]^T$$

$$u(t) = [u_1(t), u_2(t), \dots, u_6(t)]^T \quad (12)$$

Therefore the dynamic model in Equation (3) of the robot manipulator can be expressed as:

$$\dot{x} = \begin{bmatrix} x^2 \\ -M(x^1)^{-1} [C(x^1, x^2) + G(x^1)] \\ 0 \\ M(x^1)^{-1} \tau \end{bmatrix} \quad (13)$$

Sliding Mode Controller Revision

Sliding mode controller derivation has two phases:

- Phase 1 Sliding surface design
- Phase 2 Control input design

Therefore to discuss the first phase, first, let's define the sliding surface:

$$s(t) \in R^n$$

A classic form for the sliding surface is the following, which depends on just a single scalar parameter λ :

$$S = \left(\frac{d}{dt} + \lambda \right) e \quad (14)$$

where, e and \dot{e} are position and velocity error respectively and defined as:

$$e(t) = (q_d - q)$$

$$\dot{e}(t) = (\dot{q}_d - \dot{q}) \quad (15)$$

Thus, the sliding surface equation is defined as:

$$s(t) = \dot{e} + \lambda e = 0 \quad \lambda > 0. \quad (16)$$

where, λ is a positive diagonal matrix.

The next phase, (phase 2), compute the control action that drives the system trajectories on to the sliding manifold. In another way, the control is capable of drives the s variable to zero in finite time.

Therefore in case of sliding mode control, the controller fulfills the following conditions:

$$s(t) = 0, \quad \dot{s}(t) = 0. \quad (17)$$

In Equation (14) the sliding surface equation, the sliding surface can be accomplished when $s(t)$ gets zero (Slotine and Li, 1991). The reaching condition expressed by the Lyapunov function candidate makes the state vectors approach the defined sliding surface. The Lyapunov function, in Sliding mode expressed as:

$$V(x,t) = \frac{1}{2} \times s^2(x,t) = 0 \quad (18)$$

Form the above Lyapunov function, the reaching condition is taken when:

$$\dot{V}(x,t) = s(x,t) \times \dot{s}(x,t) < 0 \quad (19)$$

where, $s(x, t)$ is switching function and $\dot{s}(x, t)$ is a reaching law.

A constant rate, exponential and a power rate reaching law is introduced by (Liu and Wang, 2012) and among those exponential reaching law is taken in these studies and which is expressed as:

$$\dot{s}(t) = k_1 \operatorname{sgn}(s(t)) + k_2 S(t), \quad (20)$$

where, $k_1, k_2 > 0$.

Though in this study an exponential reaching control law is selected for both conventional sliding mode controller and terminal sliding mode controller.

Then the total torque input u becomes as:

$$u(t) = (-f(x,t) + \dot{x}_{2d}(t) + \lambda x_2(t) + \lambda x_{2d}(t) + k_1 \operatorname{sgn}(s(t))) / b(x,t) \quad (21)$$

Substitute the derived manipulator dynamics into the general second-order non-linear equation in Equation (13) and expressed in the form:

$$\ddot{q} = M^{-1}(q)[C(\dot{q},q) + G(q)] + M^{-1}(q)\tau(t) \quad (22)$$

$$\begin{aligned} f(x,t) &= M^{-1}(q)[C(\dot{q},q) + G(q)] \\ b(x,t) &= M^{-1}(q) \quad u(t) = \tau(t) \end{aligned} \quad (23)$$

The control input torque $\tau(t)$ expressed in the form:

$$M^{-1}(q)\tau(t) = \ddot{q} - M^{-1}(q)[C(\dot{q},q) + G(q)] \quad (24)$$

In the sliding surface or on the sliding plane where $S(t) = 0$ and $\dot{S}(t) = 0$, the equivalent control torque $\tau_o(t)$ is

modeled to keep the states on the sliding phase and move along with it. In the reaching phase where $S(t) \neq 0$ the discontinuous control torque $\tau_c(t)$ is designed to satisfy the reaching phase condition.

The Continuous terminal sliding mode block diagram as shown in Fig. 3, the total control input torque (τ_{tot}) to the manipulator is the sum of equivalent control torque and sliding mode control:

$$\tau_{tot}(t) = \tau_o(t) + \tau_c(t) \quad (25)$$

Since, $\tau_o(t)$ is an equivalent control torque and $\tau_c(t)$ is the sliding mode control torque or the discontinuous part:

$$\begin{aligned} \tau_{tot}(t) &= M(q)(\dot{x}_{2d}(t) + \lambda x_{2d}(t) - \lambda x_2(t) \\ &+ k_1 \operatorname{sgn}(s(t))) \\ &- [C(\dot{q},q) + G(q)] \end{aligned} \quad (26)$$

Therefore, Equation (25) is consist of both the equivalent control and the sliding mode control torque and expressed as:

$$\begin{aligned} \tau_{tot}(t) &= M(q)(\dot{x}_{2d}(t) + \lambda x_{2d}(t) - \lambda x_2(t) \\ &- [C(\dot{q},q) + G(q)]) + M(q) \\ &\times [k_1 \operatorname{sgn}(s(t)) + k_2 (S(t))] \end{aligned} \quad (27)$$

This implies:

$$\begin{aligned} \tau_o(t) &= M(q)[\lambda_1 (\dot{q}_d - \dot{q}) + \ddot{q}_d] \\ &- [C(\dot{q},q) + G(q)] \end{aligned} \quad (28)$$

and, the discontinuous control torque become as:

$$\tau_c(t) = M(q) \times [k_1 \operatorname{sgn}(s(t)) + k_2 (S(t))] \quad (29)$$

Terminal Sliding Mode Controller Design

The robotics activities in industries nowadays become more complicated and advanced, though to meet these requirements a new continuous terminal sliding mode is proposed in this paper with fuzzy gain scheduling controller. Therefore, a new continuous terminal sliding mode is developed in this paper to control the trajectory tracking of UR5 robotic manipulator with global finite-time stability is used. For the above second-order nonlinear system in Equation (22), the sliding variable or sliding function

of the terminal sliding mode controller is taken as developed by (Yu *et al.*, 2005):

$$s(t) = e(t) + \beta \text{sign}(\dot{e}(t))^\gamma = 0 \tag{30}$$

where, $1 < \gamma < 2$ and $\beta > 0$, $S(t) = [S_1(t) \dots S_n(t)]^T$ and $\beta = \text{diag}[\beta_{11} \dots \beta_{nn}]$.

Intended for simplicity some notations are introduced for the analysis of TSMC by (Haimo, 1986). Similarly as a sliding mode controller the control input or a control torque for terminal sliding mode controller is the same as given in Equation (25). Then the equivalent and the discontinuous controller is obtained as follow:

$$\text{sign}(e(t))^\gamma = \begin{bmatrix} |e_1(t)|^\gamma \text{sign}(e_1(t)) \dots |e_n(t)|^\gamma \text{sign}(e_n(t)) \end{bmatrix}^T \tag{31}$$

$$|e(t)|^\gamma = \begin{bmatrix} |e_1(t)|^\gamma \dots |e_n(t)|^\gamma \end{bmatrix} \tag{32}$$

The simplified form of terminal sliding function is expressed as:

$$s(t) = e(t) + |\dot{e}(t)|^\gamma \beta \text{sign}(\dot{e}(t)) = 0 \tag{33}$$

Then the next step is to solve the equivalent control torque on the sliding phase ($\dot{S} = 0$). Then differentiating the terminal sliding variable will become as:

$$\dot{S}(t) = \dot{e}(t) + \gamma \beta |\dot{e}(t)|^{\gamma-1} \text{sign}(\dot{e}(t)) \ddot{e}(t) \tag{34}$$

Therefore an equivalent control torque can be expressed as:

$$\tau_{equ}(t) = [C(\dot{q}, q) + G(q)] + M(q)(\ddot{q}_d) + M(q)\beta^{-1}\gamma^{-1}\text{sign}(\dot{e})^{2-\gamma} \tag{35}$$

Moreover, for the reaching phase, an exponential reaching law is taken and exponential reaching control law is described as follow:

$$\dot{s}(t) = k_1 \text{sgn}(s(t)) + k_2 S(t), \quad k_1, k_2 > 0 \tag{36}$$

Therefore, the discontinuous (the switching) control torque is obtained as:

$$\tau_c(t) = M(q) \times k_1 \text{sgn}(S(t)) + k_2 S(t) \tag{37}$$

However, the main source of chattering is sign function in the discontinuous controller, though to come

chattering, a hyperbolic tangent function (*tanh*) is adopted in the switching control instead of signum function and the equation rewritten as follow:

$$\tau_c(t) = M(q) \times k_1 \tanh(S(t)) + k_2 S(t) \tag{38}$$

Then finally the total control torque input can be expressed as:

$$\tau_{tot}(t) = \tau_{equ}(t) + \tau_{diaC}(t) \tag{39}$$

The Lyapunov stability and a finite-time convergence were also computed according to the Lyapunov stability function by (Yu *et al.*, 2005). The controller scheme for the terminal sliding mode controller is depicted below in Fig. 3.

Fuzzy Inference System Design for Gain Scheduling of TSM Controller

If a nonlinear dynamics system plants are changes its operating point due to unknown uncertainties and disturbance, it may be conceivable to alter the parameters of the controller according to each operating point. This is known as gain scheduling. As it has been proven by many scholars that the fuzzy logic controller is appropriate for gain scheduling. The fuzzy logic controller consists of four main general steps of (i.e., fuzzification, inference rule and defuzzification).

The inputs for the fuzzy inference system are selected to be the terminal sliding variable $S(t)$ and change in the terminal sliding variable $\dot{S}(t)$ and the output would become the switching gain $K(t)$ as depicted in Fig. 4. The universal discourse for the inputs is taken from the results of the simulation response, whereas the output variable is taken by trial and error for the optimum position error and torque response. Form the input universal discourse of $S(t)$ and $\dot{S}(t)$ the membership functions are designed for both inputs parameters as Negative Big (NB), Negative (N), Zero (ZO), Positive (P) and Positive Big (PB).

The fuzzy controller $K(t)$ output membership function also classified as Very Small (VS), Small (S), Medium (ME), Large (L) and Very Large (VL). All input membership functions are taken as triangular and uniformly distributed shape. Thus in this paper, a Prod-Sum and a constant Takagi Sugeno Fuzzy inference system are used to design the rule base.

The linguistic controller contains rules in the *if-then* format is designed on MATLAB/Simulink according to the existence of the sliding mode ($S(t) \dot{S}(t) < 0$). Though as long as the state trajectories are on the sliding mode the switching gain should be small, whereas when the states are far from the sliding mode due to unknown model uncertainties and disturbance the switching gain should be increased.

The surface of the rule base for the six sliding variable and its derivatives are depicted as follows.

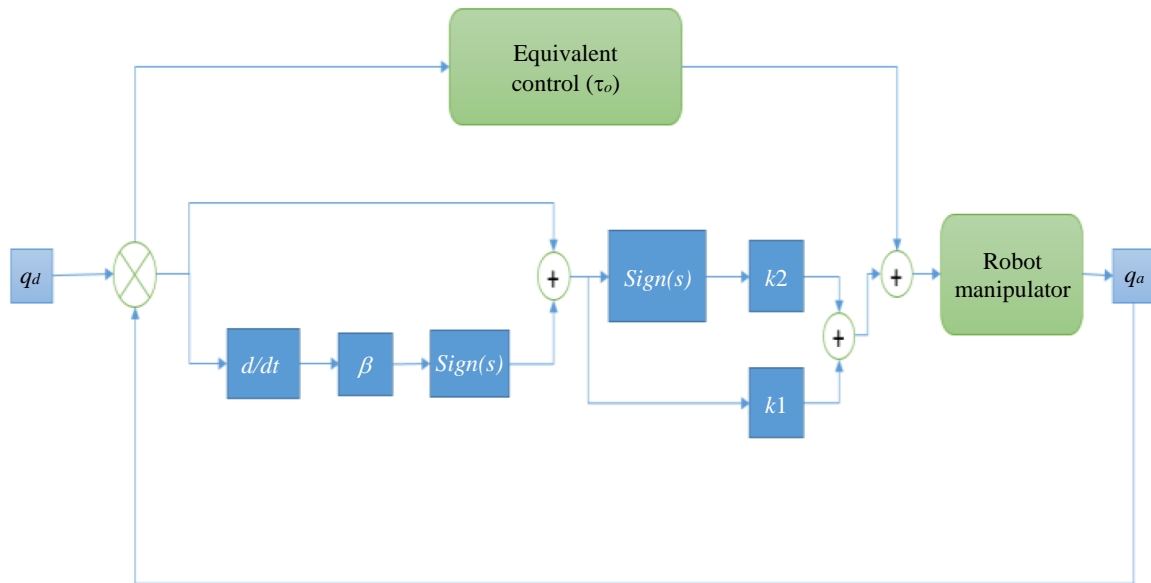


Fig. 3: Continuous terminal sliding mode controller schematic diagram

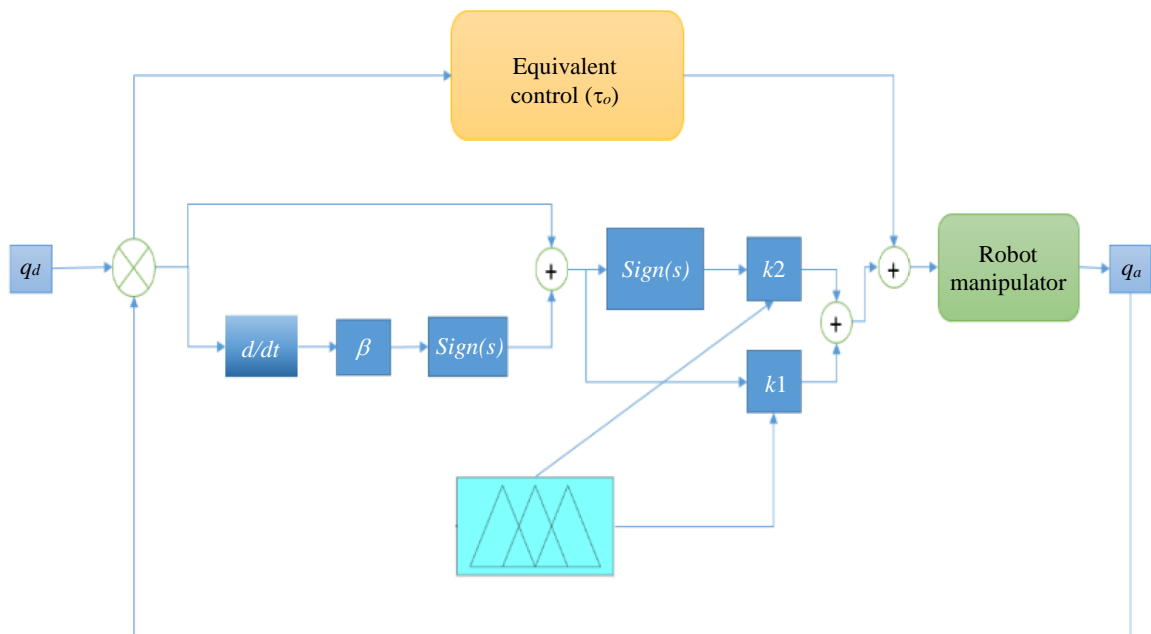


Fig. 4: An adaptive fuzzy gain scheduling terminal sliding mode control scheme

Simulation Result and Discussion

The level-2 MATLAB S-function API allows the any user to create custom block by using MATLAB R2016a language with multiple inputs and multiple outputs. Thus, the dynamics of the UR5 robot manipulator is implemented in SIMULINK using level-2 MATLAB S-Function.

For solving the differential equations commonly *ode 45* is used, but sometimes automatically it switched to *ode 15* solver. At each step time of the simulation, the

state vector is sent from SIMULINK to the MATLAB interface via to workspace block. Finally, the simulation result is plotted graphically, by Scope block or 3D Graph.

Open-Loop Simulation

In most cases of the open-loop system as demonstrated in Fig. 5, the output response has no effect on the input to the control process. Therefore, the velocity and position outputs of the UR5 robot manipulator with zero torque input is studied. Thus, simulation was initially done under a

zero torque input situation. As a result unlike the other joints, joint 5 and 6 shows highly disturbed and chaotic properties as shown in the Fig. 6.

Zero torque input is taken as:

$$\tau = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

The result of the instability behavior of the two joints is caused by gravitational force. Therefore, applying gravity compensating torque for the manipulator will result in a stable or undisturbed system response as shown in the Fig. 7.

The gravity compensating torque is found as:

$$\tau = [0 \ -51.4896 \ -14.6133 \ 0 \ 0 \ 0]^T \text{ N.m}$$

A Sliding Mode Controller for Tracking Control of UR5

The Sliding mode controller internal structure is also depicted in Fig. 8 and under this block basically, there is two main control gains which is the sliding surface gain and the reaching control gain and the saturation function block.

Moreover, the same joint trajectory path is used for both (i.e., sliding mode controller and for fuzzy gain scheduling terminal sliding mode controller) and the initial and final joint positions are expressed as follows:

$$q_i = [\pi/6 \ -\pi/6 \ 0 \ \pi/4 \ \pi/8 \ 0]^T \text{ rad}$$

$$q_f = [\pi \ -\pi/2 \ -\pi/4 \ 0 \ \pi/4 \ 0]^T \text{ rad}$$

The sliding Surface gain and the reaching control gain is taken as following for trajectory tracking control of the UR5 robot manipulator:

$$\lambda = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} 19 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 0 & 0 & 0 \\ 0 & 0 & 0 & 18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 0 & 0 & 18 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

Finally, after the simulation is done, the 3D graph plot the actual and desired trajectory path of the robot manipulator, here in the Fig. 9 it seems both having the same color, however, the graph is consists of the blue line and red line, where the blue one is the reference input trajectory while the red line represents the actual path of the robot manipulator.

Then the results of the position error, velocity error and the total control input torque in *nm* to the system are depicted as follows.

As can be seen in the above-depicted diagrams, on the position and velocity error response of the UR5 robot manipulator in Fig. 10 to 12 resulted in an unsatisfactory error convergence in finite time, however in the 3D graph of the actual and the desired trajectory path looks accurately matched. Nevertheless, the system error exhibited large error relative to the terminal sliding mode controller. The sliding function of the traditional sliding mode control also becomes chatter due to a high frequency switching action of the discontinuous controller.

The control torque input in Fig. 13 also has a chattering phenomenon, which is an undesired and unwanted condition on the traditional sliding mode controller due to a high frequency action of the discontinuous controller, the chattering phenomenon is most difficult to implement practically on the actuator. This problem is caused by two main reasons one is tuning the switching controller constantly at sliding mode and at reaching condition when the system is at sliding mode the switching control gain should be small or less, whereas when the system at reaching condition the switching gain should become large to overcome disturbance.

The second case that produces chattering problem is taking the linear sliding function. Though to alleviate the above major problems, next a fuzzy gain scheduling terminal sliding mode controller is used and simulated the result in the next coming chapter.

Note that in Fig. 14 *qa* stands for actual angular positions and *qd* represents the desired angular position of the applied to the joints in rad.)

Such that, in Fig. 15, *Xa, Ya, Za* stands for actual angular positions and *Xd, Yd, Zd* represents the desired angular position of the applied to the joints in meter.

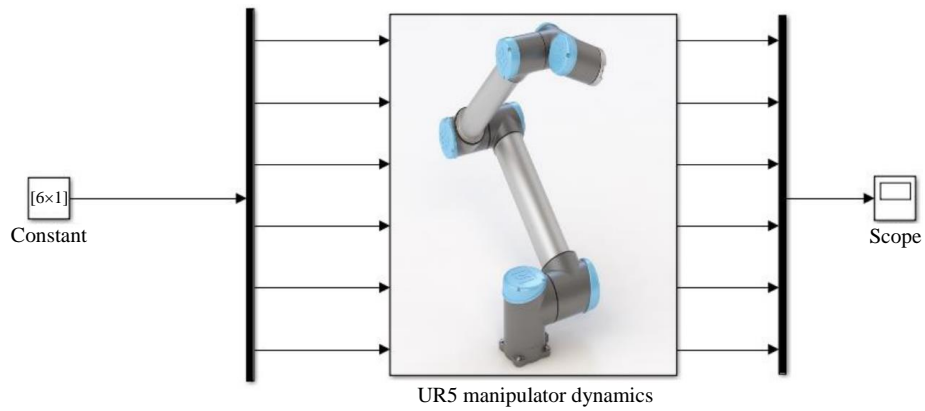


Fig. 5: The open-loop dynamic model of the UR5 robot

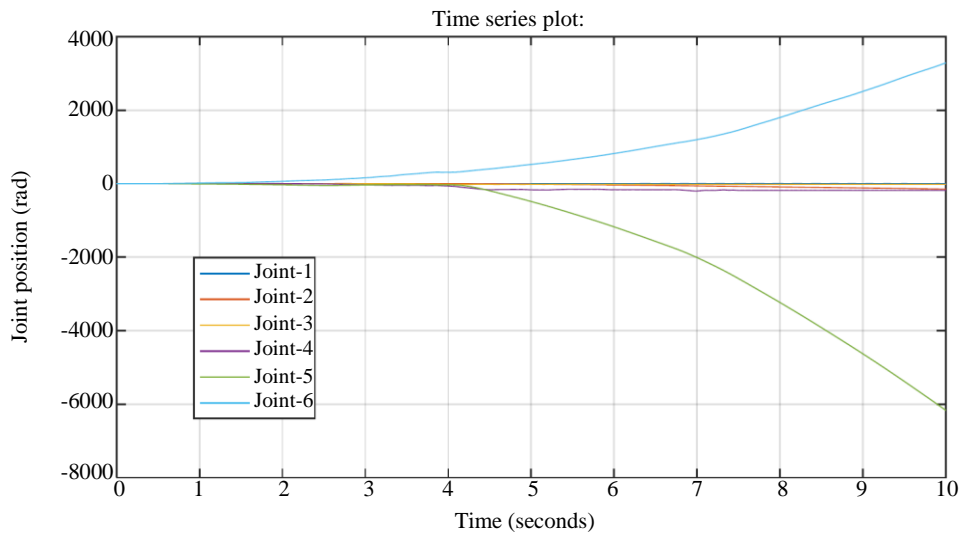


Fig. 6: Joint position response of the robotic manipulator under zero torque

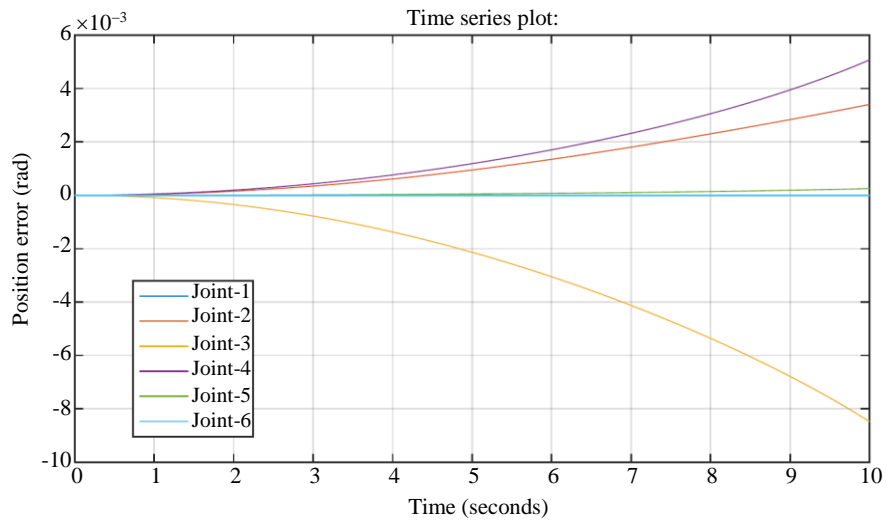


Fig. 7: Manipulator position response with gravity compensating torque

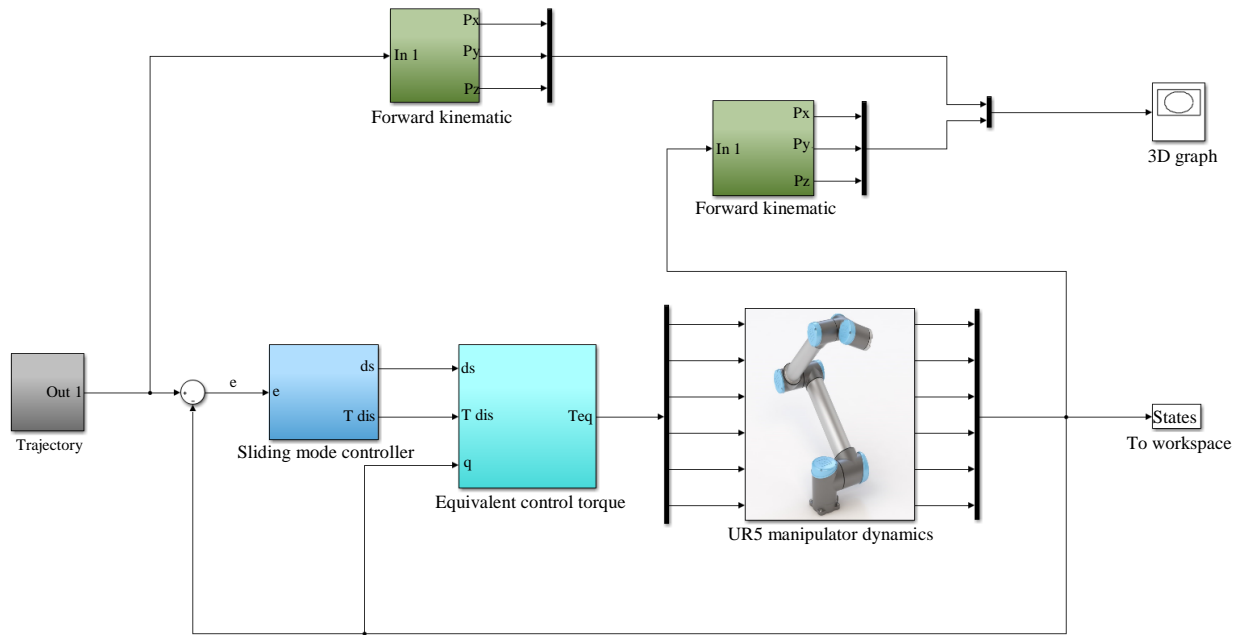


Fig. 8: The convention Sliding mode control of UR5 robot manipulator Simulink model

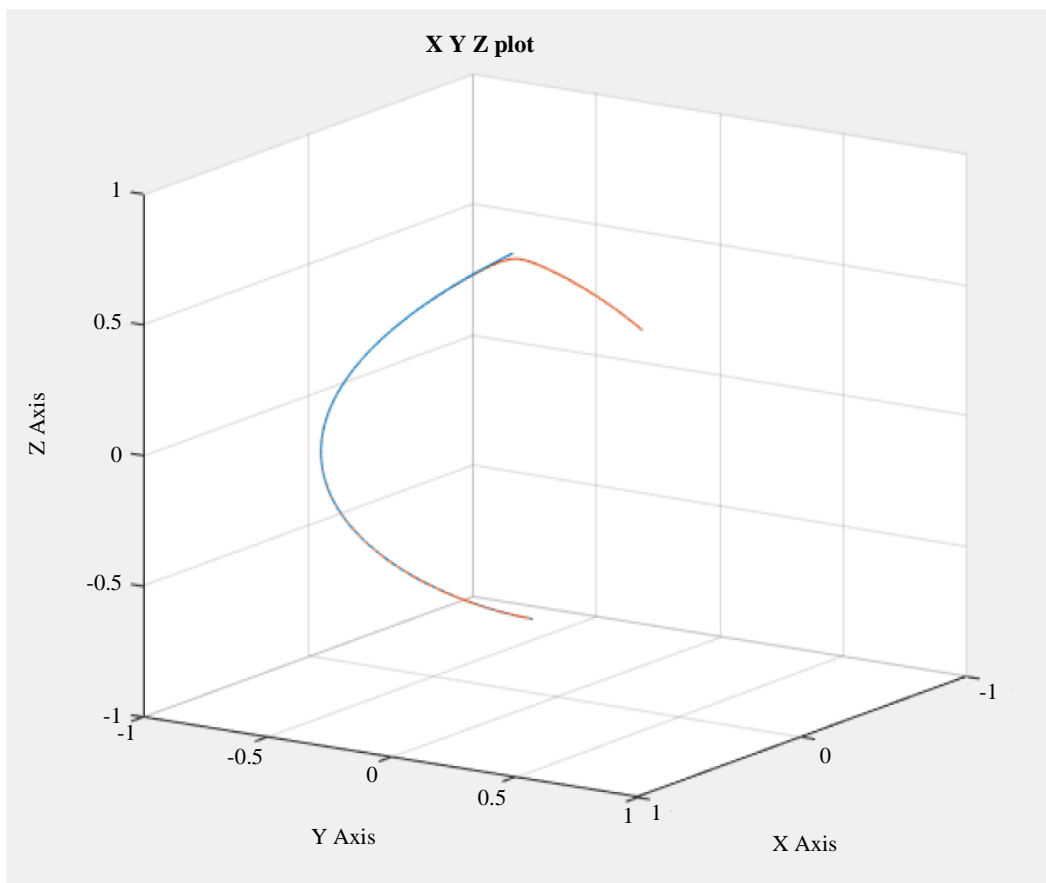


Fig. 9: The diagram shows both the actual (tracked) and desired

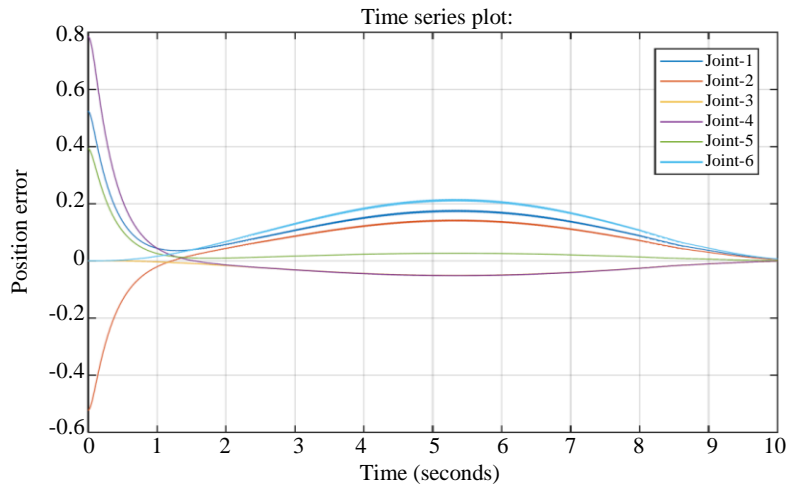


Fig. 10: Joints position error of the UR5 robot manipulator joints under conventional SMC

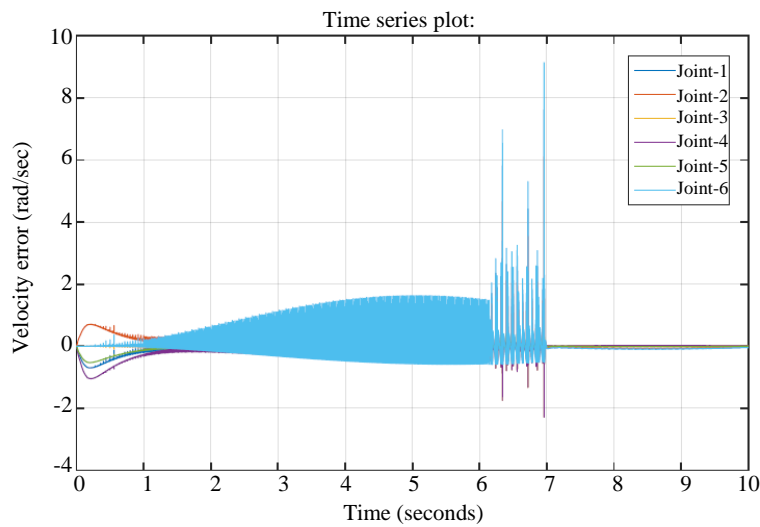


Fig. 11: Joints velocity error of the UR5 robot manipulator

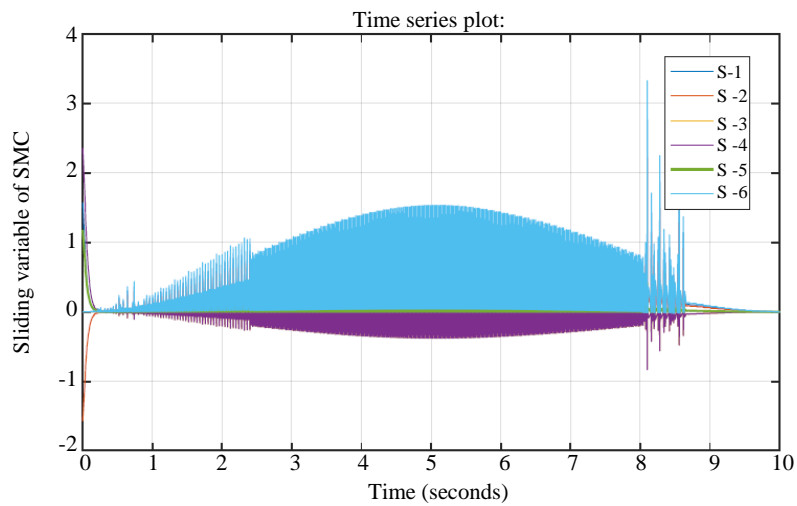
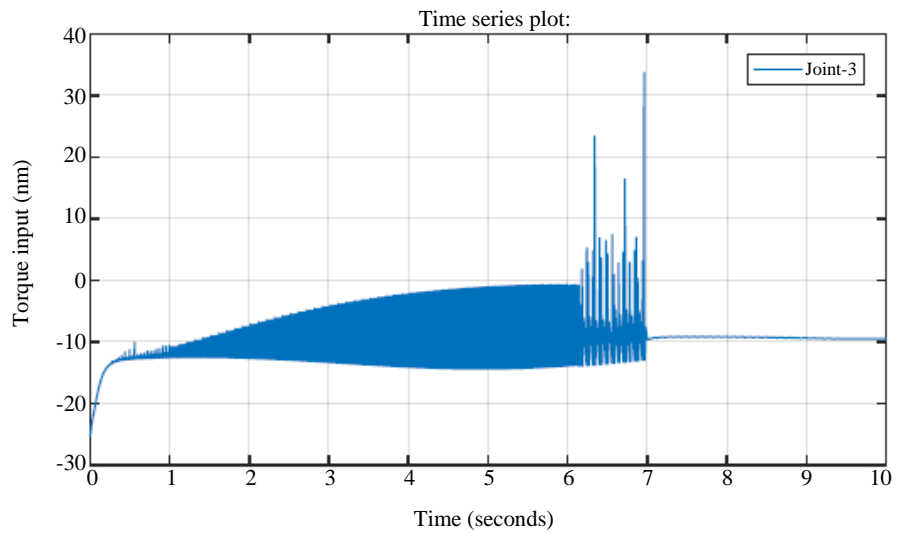
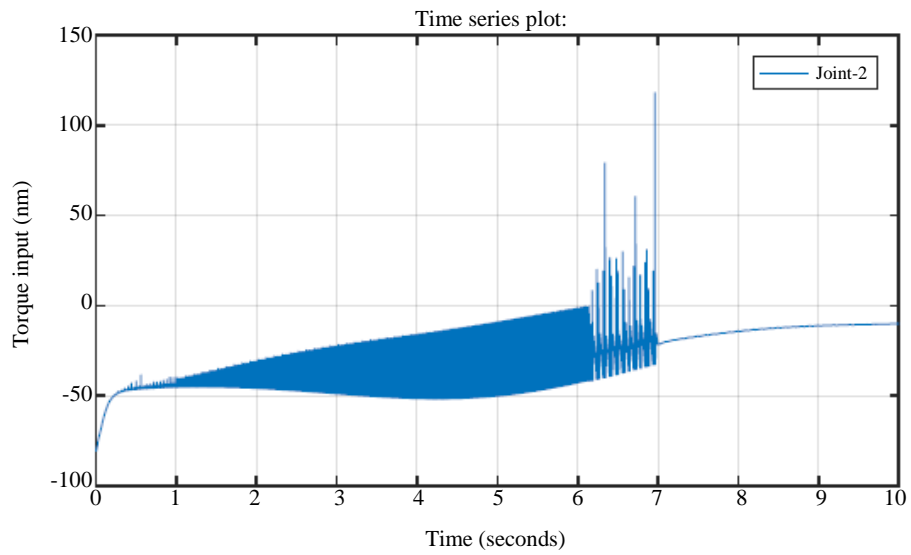
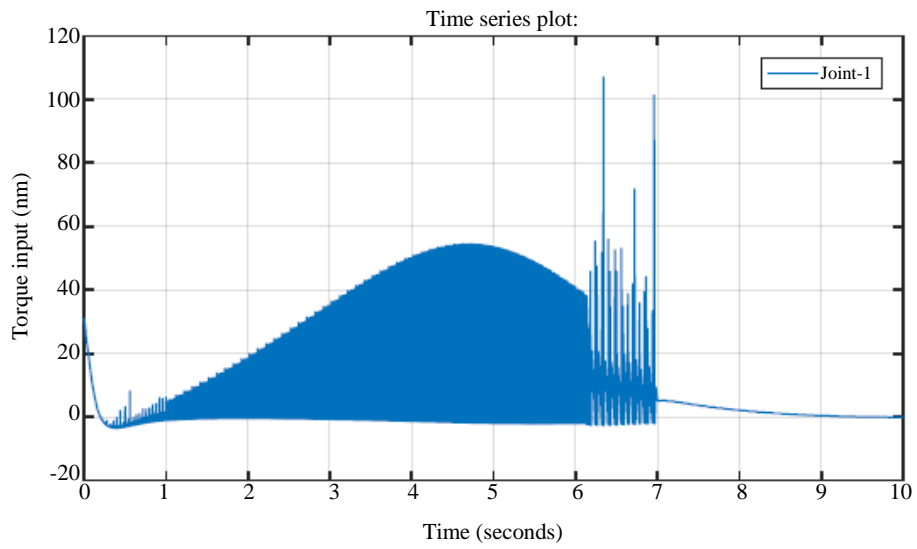


Fig. 12: Sliding variable S of the conventional SMC controller for UR5 control



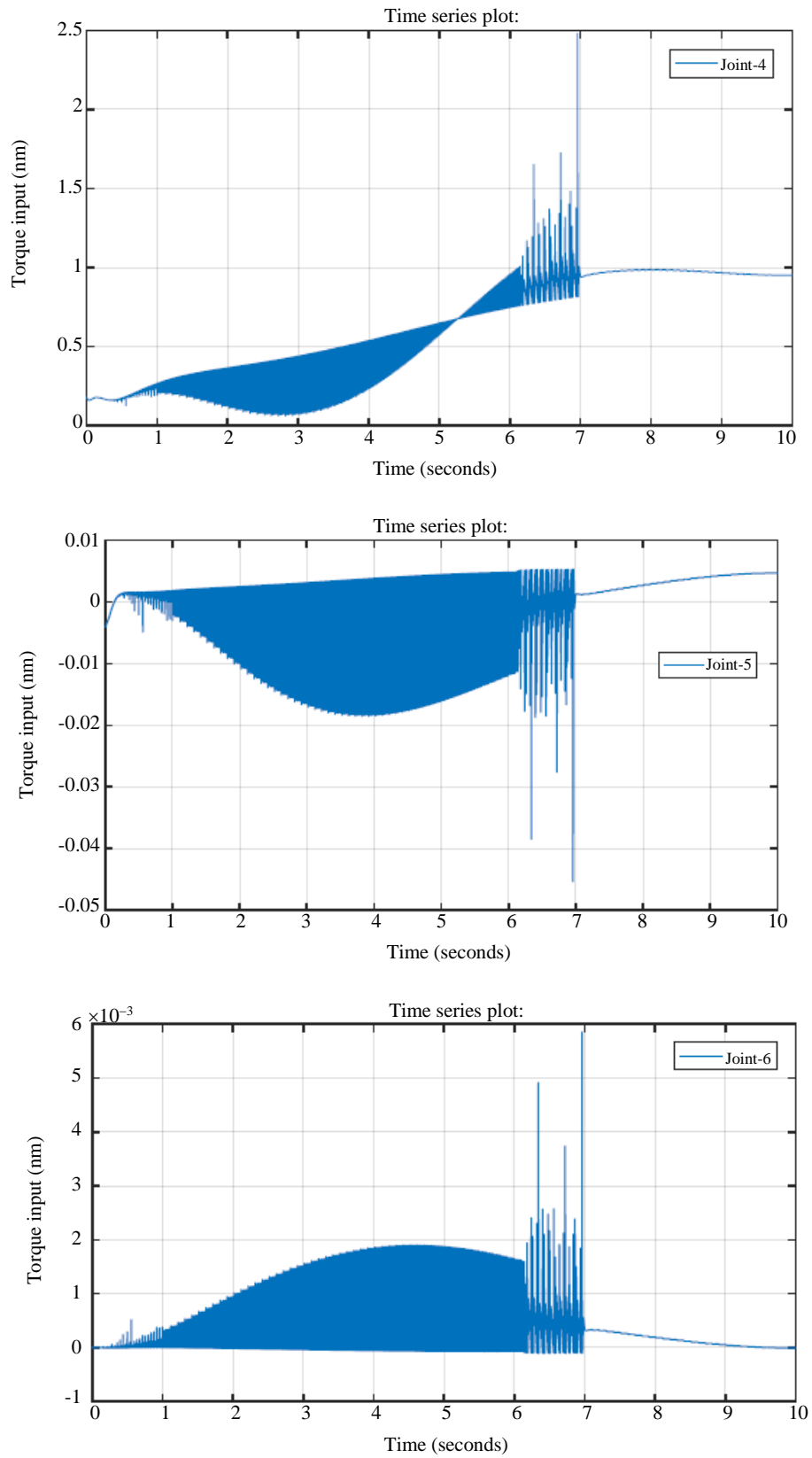


Fig. 13: The control input Torque for each joint of the robot manipulator

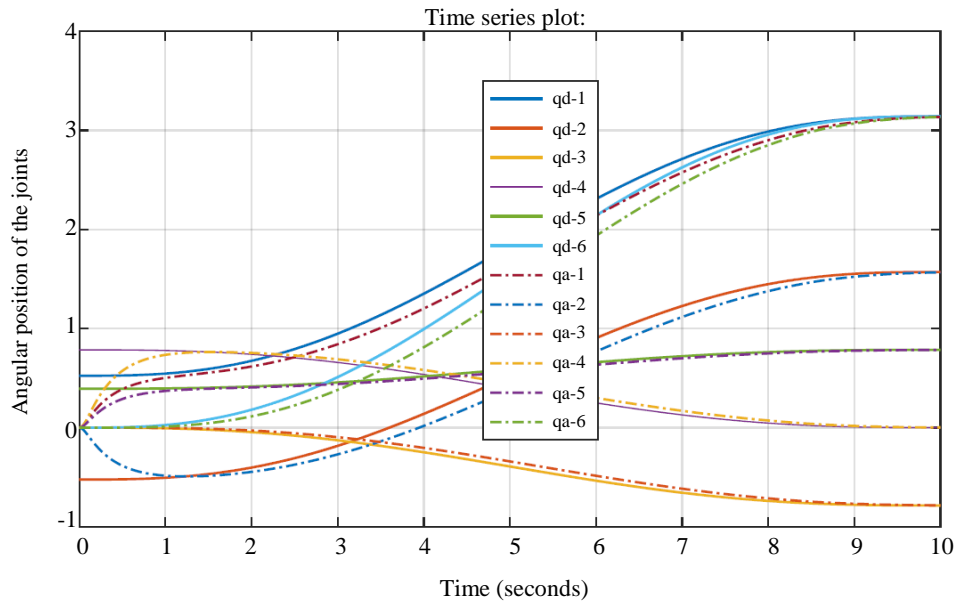


Fig. 14: The actual and desired joint positions of UR5 robot manipulator under SMC controller

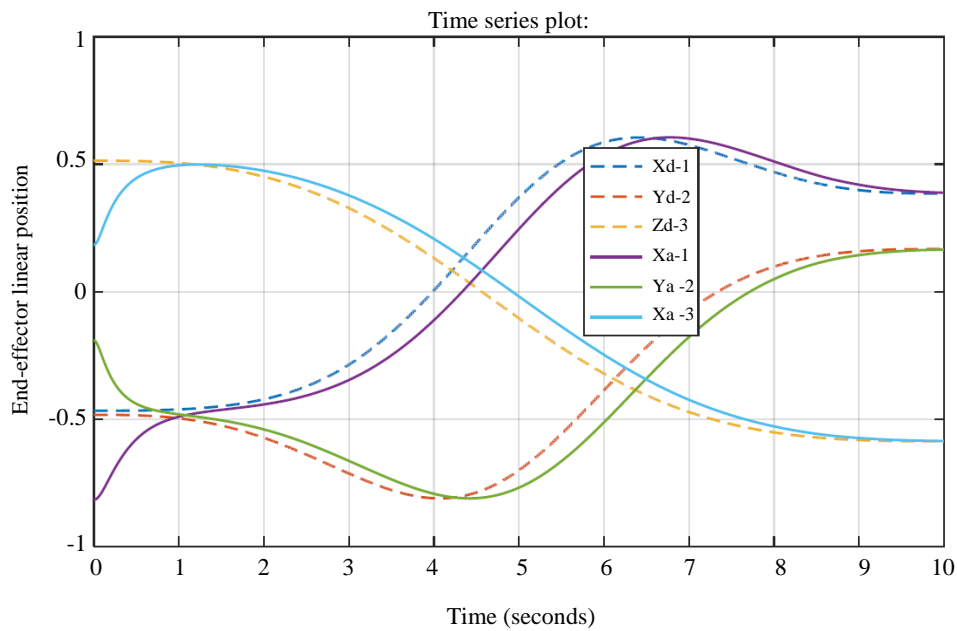


Fig. 15: The end-effector linear positions of UR5 robot manipulator under SMC controller

Fuzzy Gain Scheduling Terminal Sliding Mode Controller for Tracking Control of UR5

In this section, a fuzzy gain scheduling controller is tested to control the trajectory of UR5 robot manipulator. The basic control approach of FGSTSM controller is reviewed in the previous chapter. As per the mathematical formulation, the FGSTSM controller was implemented in

Simulink. The performance of the controller also measured as the position and velocity error and the control input torque. As per the mathematical formulation, the FGSTSM controller was implemented in Simulink. The performance of the controller also measured as the position and velocity error and the control input torque.

The sliding Surface gains of a terminal sliding mode controller is taken as follow:

$$\beta = \begin{bmatrix} 0.42 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.21 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.55 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.62 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.49 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.32 \end{bmatrix}$$

$$\gamma = \begin{bmatrix} 1.88 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.88 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.65 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.48 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.59 \end{bmatrix}$$

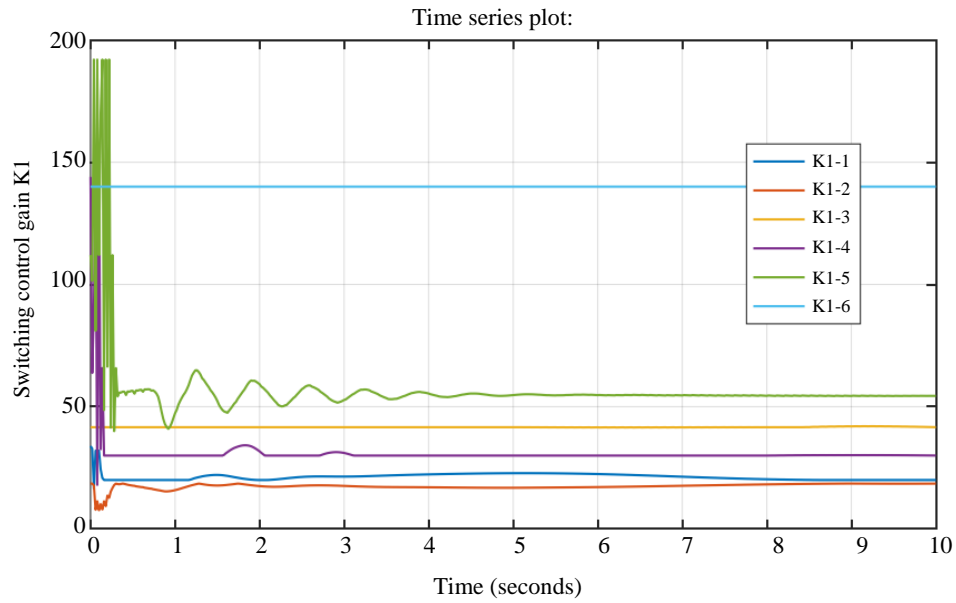


Fig. 16: An adaptive switching gain for tuning terminal Sliding Mode controller

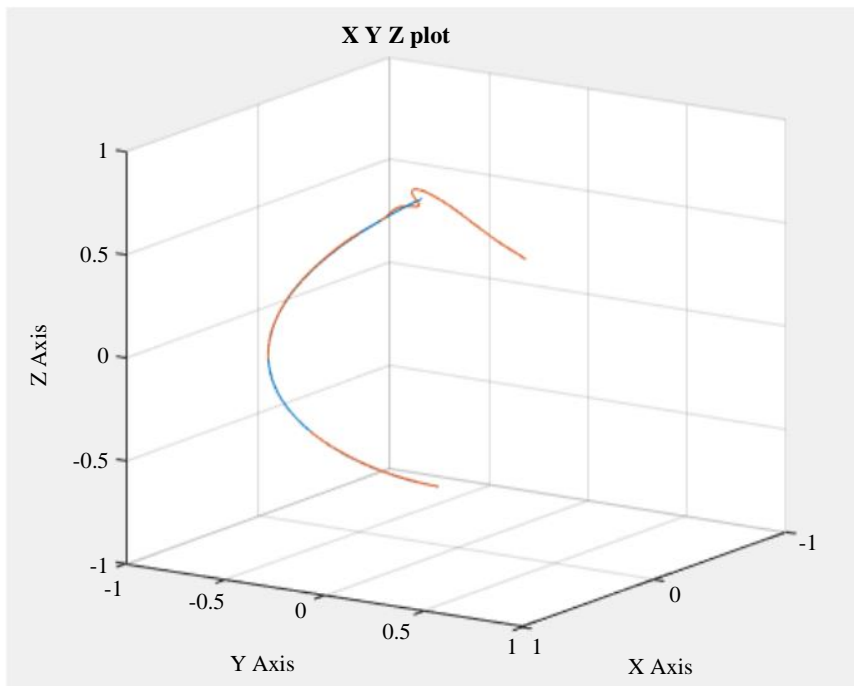


Fig. 17: The desired and actual trajectory path of UR5 using FGSTSM controller

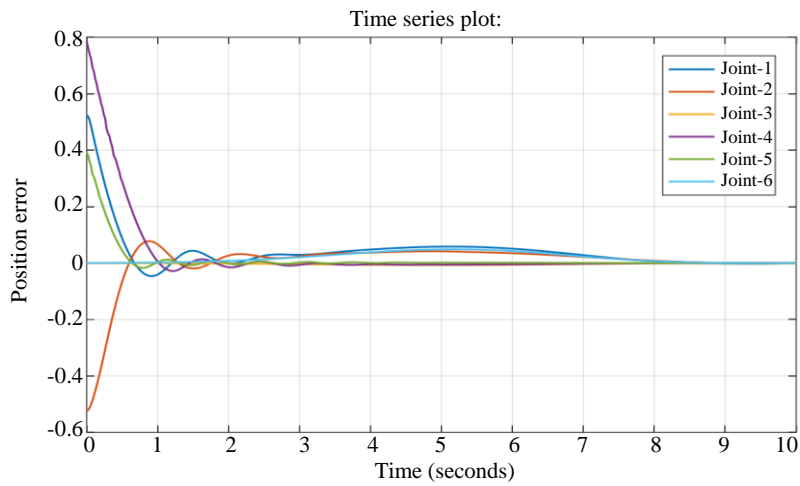


Fig. 18: Position error of the FGSTSM controller of UR5

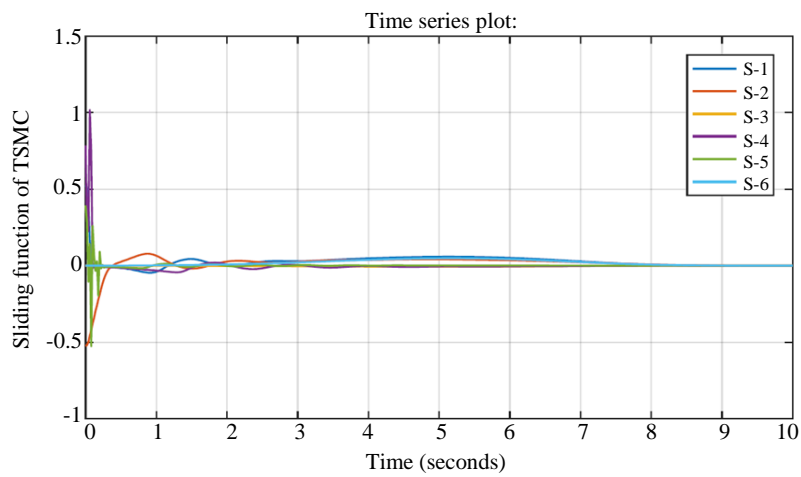


Fig. 19: The sliding function S of terminal sliding mode controller FOR UR5 robot manipulator

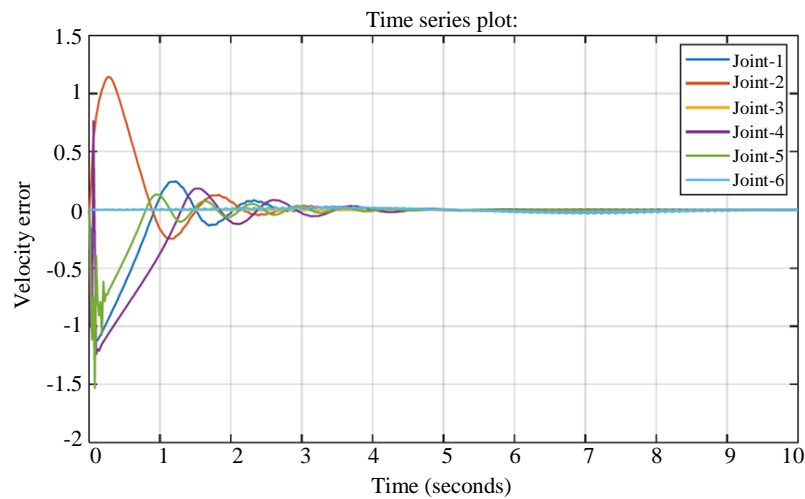
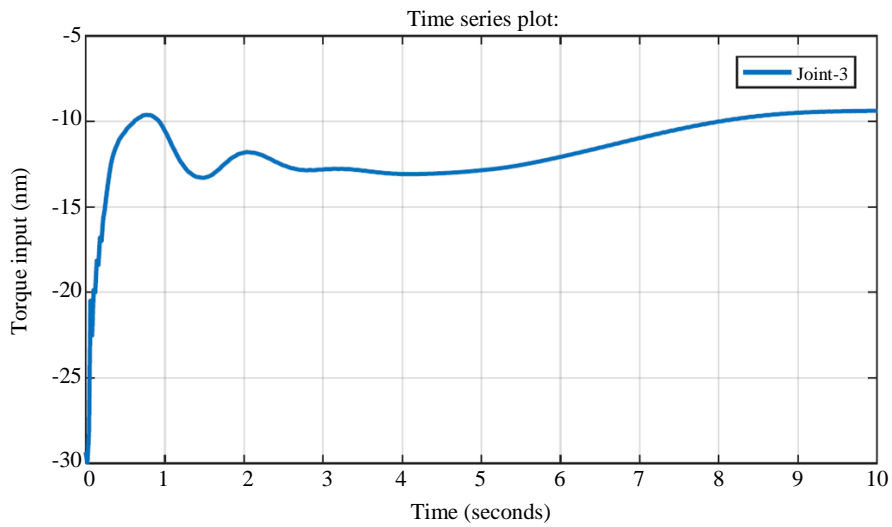
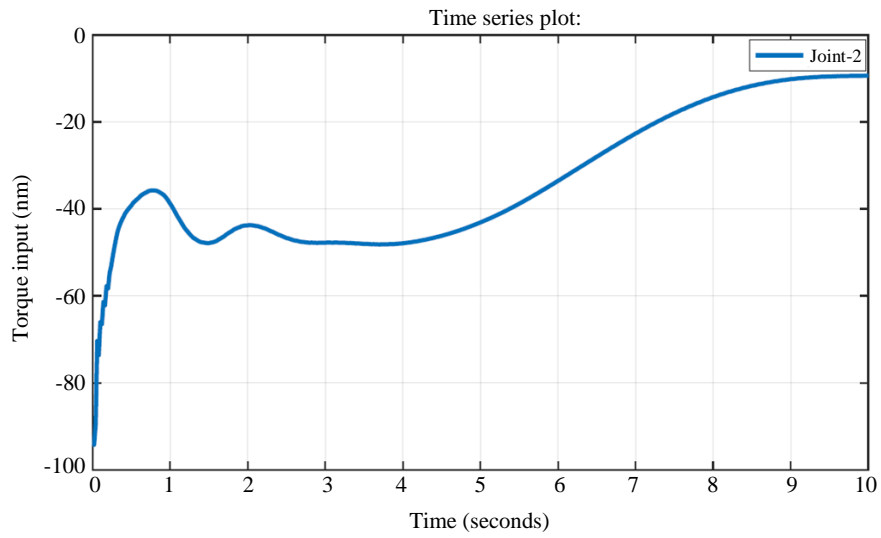
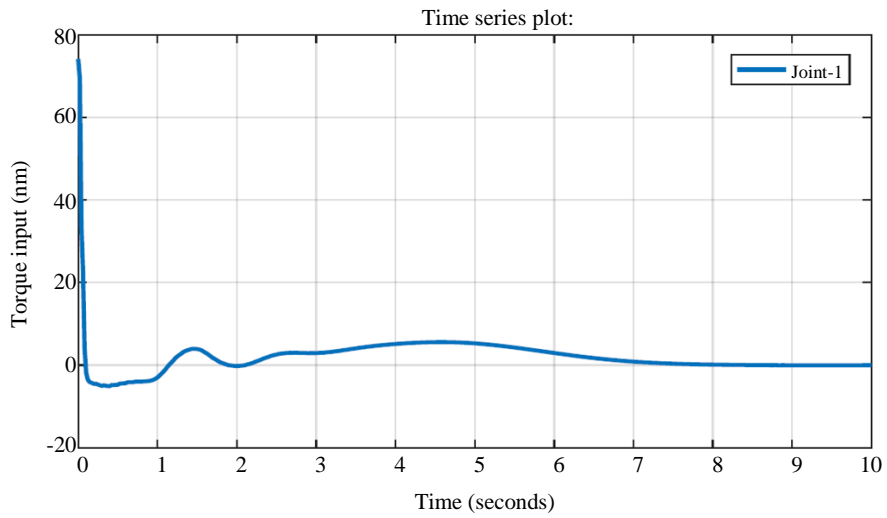


Fig. 20: Velocity error of FGSTSM controller of the UR5 robot manipulator



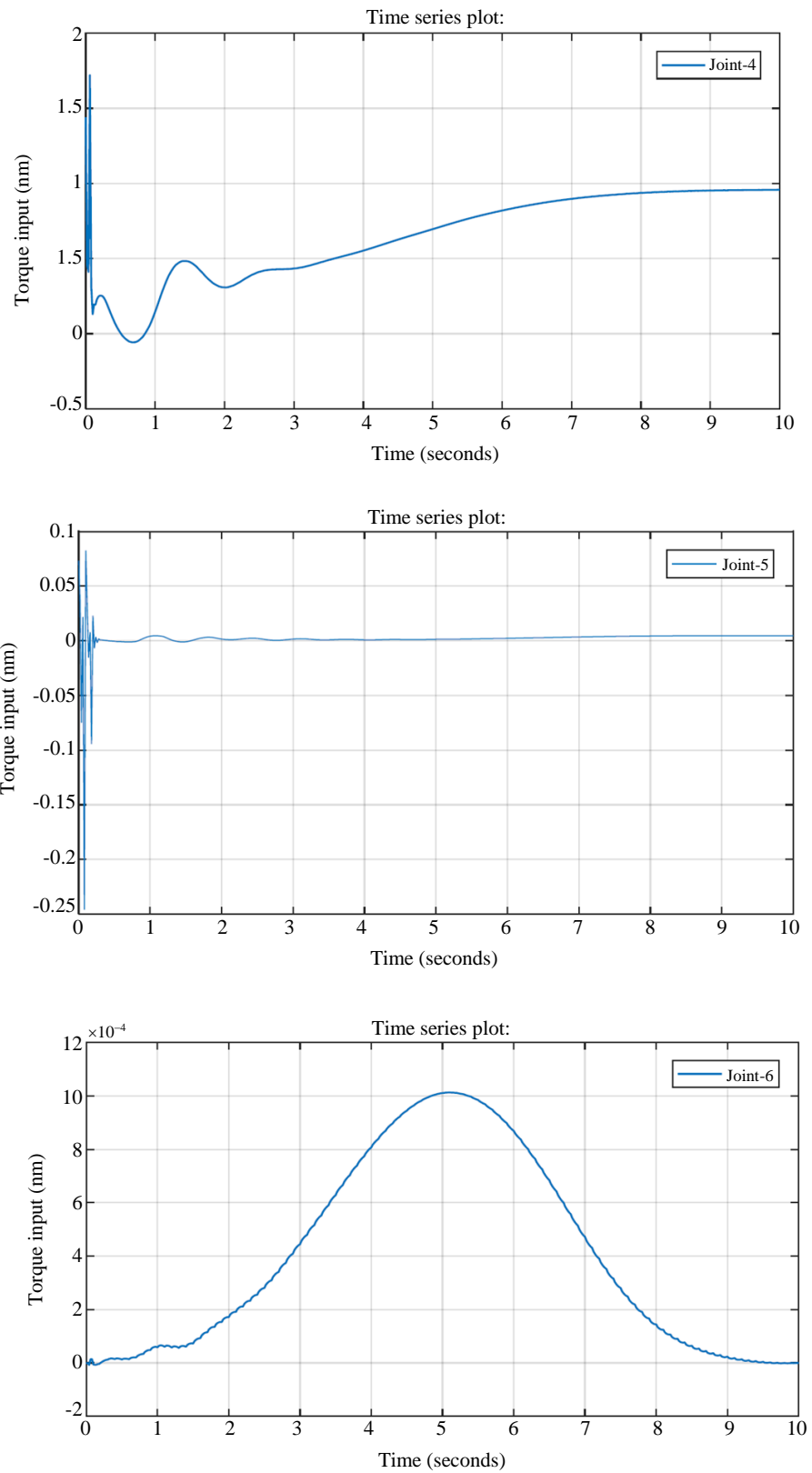


Fig. 21: Torque input for each joint of UR5 robot manipulator from the FGSTSM controller

However, switching control gain K_1 is obtained and tuned by fuzzy inference system adaptively and the switching control gain K_2 are tuned manually and the tuning parameters K_2 are used as the given below:

$$K_2 = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 & 0 & 43 \end{bmatrix}$$

In Fig. 22 the broken lines represent the reference x , y , z position, whereas the solid lines of the graph indicate the tracked linear position of the UR5 manipulator.

In Fig. 23 all the broken lines represent the actual tracked joint position, whereas the solid line represents the reference joint position of the UR5 robot manipulator.

Fig. 17 showed the results of the actual and the reference trajectory path that the manipulator has moved on under the FGSTSM controller. Hence the red colored line represents the actual path that the robot moved whereas the blue one is the reference trajectory path given to the plant.

In this study the conventional SMC controller and the developed Fuzzy gain Scheduling Terminal sliding mode controller have compared, accordingly the FGSTSM controller exhibits a fast tracking under equal simulation time, a better tracking accuracy and alleviate the chattering problem. Various responses of the proposed fuzzy gain scheduling terminal sliding mode controller are presented above. Firstly as it is shown in the position error graph in Fig. 18 of the FGSTSM controller the position error converges to the equilibrium in finite time.

Moreover, the other two basic responses from the trajectory tracking control of UR5 using FGSTSM controller are the sliding function (s) and its derivatives (\dot{s}) of TSMC. The response of the sliding function and its derivatives of TSM controller are demonstrated in Fig. 19 and Fig. 20 respectively. As it can be seen clearly

both exhibits a small amounts of steady state error as well as fast finite time convergence.

The position error also highly minimized when it is compared to the sliding mode controller. The other prominent contribution of the proposed FGSTSM controller is alleviating the chattering problem in the system. The chattering phenomenon is eliminated by tuning the gain adaptively using the fuzzy inference system and using the nonlinear sliding variable (i.e., terminal sliding mode controller) as shown in the Fig. 21. The other main advantage of tuning the switching control gain are used to handle system uncertainties by increasing the switching gain automatically. However, as it is reviewed in the previous chapter the implementation of the FGSTSM to the tracking control of the UR5 robot manipulator is relatively complicated.

In this study the conventional SMC controller and the developed Fuzzy gain Scheduling Terminal sliding mode controller have compared, accordingly the FGSTSM controller exhibits a fast tracking under equal simulation time, a better tracking accuracy and alleviate the chattering problem.

He *et al.* (2016) developed AFSM controller with nonlinear observer for redundant manipulator. The developed control approach have resulted a superior output in tracking performance and alleviating the chattering phenomenon. However, the main shortcoming of this scholars is related to the tuning parameters of the controller to attune uncertainties, hence in order to get a better controller performance they have increased the sliding mode and the reaching mode controller gain to the highest tuning parameter, sometimes this scenario might be impractical.

Neila and Tarak (2011) an adaptive terminal sliding mode controller was developed for robotic manipulator, adaptive control algorithm was utilized to estimate bounds of uncertainties, this leads the controller approaches to be complex. Though the tracking error and the chattering problem bring about satisfactory for 2 DOF robot arm.

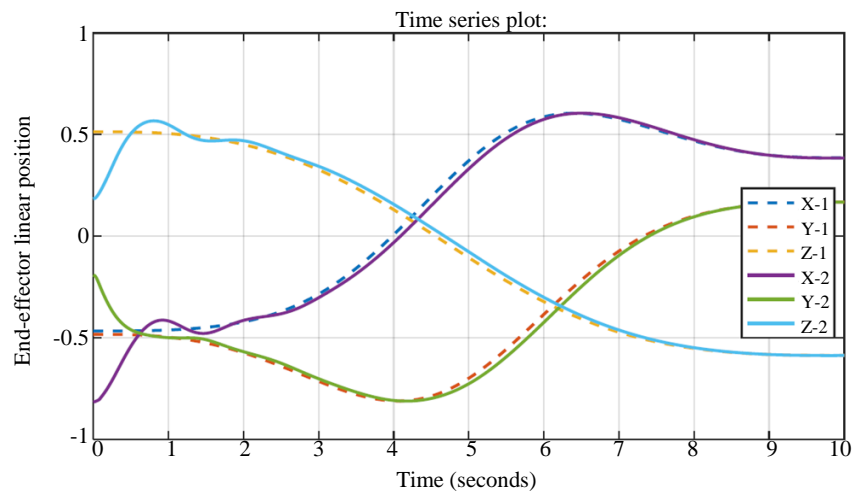


Fig. 22: The linear end-effector positions (in meter) of UR5 robot manipulator under FGSTSM controller

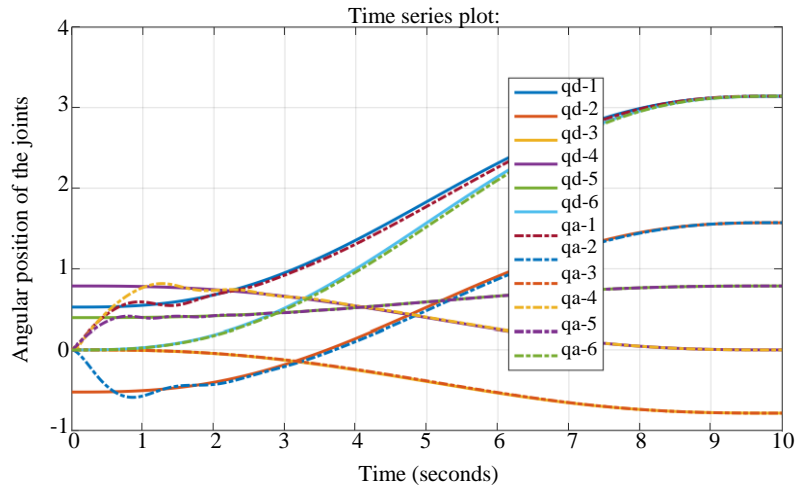


Fig. 23: Joint positions (in rad) of UR5 robot manipulator under FGSTSM controller

Mustafa *et al.* (2017) unlikely this paper was tried to remove the effect of lumped uncertainties and disturbance for trajectory tracking of robotic manipulator. Under this work the control torque input wasn't considered.

Conclusion

In this study, a new fuzzy gain scheduling terminal sliding mode controller was developed for the tracking control UR5 robotic manipulator. After validating the open loop system stability, trajectory tracking control by the conventional sliding mode controller and FGSTSM controller was developed and simulated. After validating the open-loop system stability, trajectory tracking control by the conventional sliding mode controller and FGSTSM controller was developed and simulated. Simulation results demonstrate the proposed FGSTSM controller is effectively alleviated the undesired chattering in the torque input to the manipulator ensuring a better tracking performance. On the other hand, the proposed controller successfully tune the switching gain K adaptively as demonstrated in the Fig. 16, this definitely will help the controller to handle undesired uncertainties as well as helpful for alleviating chattering.

In this study, the hyperbolic tangent function was used in the discontinuous (switching) controller for FGSTSM control instead of *sign* function, which also has contributed a lot in smoothing the control signal. Hence, in general, the conventional SMC has resulted in chattered control input due to a constant switching gain and the sign function of the discontinuous controller in moderate tracking error. Moreover, SMC converges to the equilibrium point asymptotically. Whereas in the proposed controller the chattering problem in the control input is completely alleviated and the state trajectories converge to the equilibrium points with global and finite-time stability.

Acknowledgment

First and above all, I thank the lord almighty, to whom I owe my very existence for providing me this opportunity and granting me the capability to proceed successfully. I would like to pass my genuine gratitude to my advisor Gebremichael T. (PhD) for the continuous support of my MSc. Study and related research, for his patience, motivation and immense knowledge.

Author's Contribution

Andualem Welabo: Developed the mathematical model, formulate the control algorithm and simulate.

Gebremichael Tesfamariamr: Construct the theoretical framework and supervised the research work.

Ethics

This article is an original research paper. There are no ethical issues that may arise after the publication of this manuscript.

References

- Feng, Y, Yu X., & Man, Z. (2002). Non-singular terminal sliding mode control of rigid manipulators. *Automatica*, 38; (2159-2167).
- Galicki, M. (2016). Finite-time trajectory tracking control in a task space of robotic manipulators. *Automatica*, 67, 165-170.
- Haimo, V. T. (1986). Finite time controllers. *SIAM Journal on Control and Optimization*, 24(4), 760-770.
- He, J., Luo, M., Zhang, Q., Zhao, J., & Xu, L. (2016). Adaptive fuzzy sliding mode controller with nonlinear observer for redundant manipulators handling varying external force. *Journal of Bionic Engineering*, 13(4), 600-611.

Kane, T. R., & Levinson, D. A. (1983). The use of Kane's dynamical equations in robotics. *The International Journal of Robotics Research*, 2(3), 3-21.

Kebria, P. M., Al-Wais, S., Abdi, H., & Nahavandi, S. (2016, October). Kinematic and dynamic modelling of UR5 manipulator. In 2016 IEEE international conference on systems, man and cybernetics (SMC) (pp. 004229-004234). IEEE.

Kufieta, K. (2014). Force estimation in robotic manipulators: Modeling, simulation and experiments. Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology.

Liu, H., & Zhang, T. (2012). Fuzzy sliding mode control of robotic manipulators with kinematic and dynamic uncertainties. *Journal of dynamic systems, measurement and control*, 134(6).

Liu, J., & Wang, X. (2012). Advanced sliding mode control for mechanical systems (pp. 31-35). Beijing: Springer.

Luh, J.Y, Walker M.W., & Paul, R.P.C. (1980). On-line computational scheme for mechanical manipulators. *J. Dyn. Sys, Meas, Control*, 102; (69-76).

Mustafa, A., Dhar, N. K., Agrawal, P., & Yerma, N. K. (2017, April). Adaptive backstepping sliding mode control based on nonlinear disturbance observer for trajectory tracking of robotic manipulator. In 2017 2nd International Conference on Control and Robotics Engineering (ICCCE) (pp. 29-34). IEEE.

Neila, M. B. R., & Tarak, D. (2011). Adaptive terminal sliding mode control for rigid robotic manipulators. *International Journal of Automation and Computing*, 8(2), 215-220.

Pan, L., Bao, G., Xu, F., & Zhang, L. (2018). Adaptive robust sliding mode trajectory tracking control for 6 degree-of-freedom industrial assembly robot with disturbances. *Assembly Automation*.

Paul, R. P. (1981). *Robot manipulators: mathematics, programming and control: the computer control of robot manipulators*. Richard Paul.

Sciavicco, L., & Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.

Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice hall.

Soltanpour, M. R., Khooban, M. H., & Soltani, M. (2014). Robust fuzzy sliding mode control for tracking the robot manipulator in joint space and in presence of uncertainties. *Robotica*, 32(3), 433.

Vijay, M., & Jena, D. (2017). PSO based neuro fuzzy sliding mode control for a robot manipulator. *Journal of Electrical Systems and Information Technology*, 4(1), 243-256.

Wan, K., Wu, A., Liu, H., & Dong, N. (2016, June). Trajectory tracking of 3-DOF spatial robot manipulator. In 2016 12th World Congress on Intelligent Control and Automation (WCICA) (pp. 3259-3264). IEEE.

Yu, S., Yu, X., Shirinzadeh, B., & Man, Z. (2005). Continuous finite-time control for robotic manipulators with terminal sliding mode. *Automatica*, 41(11), 1957-1964.

Yu, X., & Zhihong, M. (1998). Multi-input uncertain linear systems with terminal sliding-mode control. *Automatica*, 34(3), 389-392.

Yu, X. & Zhihong, M. (2002). Fast terminal sliding-mode control design for nonlinear dynamical systems. *Trans. Circuits Syst. I Fundamental Theory Applic*, 49; (261-264).

Appendix

Forward Kinematics

$$A_{01} = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & 0.0890 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{12} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & -0.425\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & -0.425\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{23} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & -0.392\cos(q_3) \\ \sin(q_2) & \cos(q_2) & 0 & -0.392\sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{34} = \begin{bmatrix} \cos(q_4) & 0 & \sin(q_4) & 0 \\ \sin(q_4) & 0 & -\cos(q_4) & 0 \\ 0 & 1 & 0 & 0.1091 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{45} = \begin{bmatrix} \cos(q_5) & 0 & -\sin(q_5) & 0 \\ \sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & -1 & 0 & 0.09465 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{56} = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0.0823 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The resulted general transformation matrix is a 4x4:

$$T_6^0 = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4 * A_6^5$$

Dynamics Equation

$$\tau = M(q)\ddot{q} + C(\dot{q}, q) + g(q)$$

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix}$$

$$m_{11} = \cos((2*q_4 + 2*q_3 + 2*q_2 - q_5)) * 0.436519680000000548e-4 + \cos((q_5 + 2*q_4 + 2*q_3 + 2*q_2)) * (-0.436519680000000548e-4) + \sin((q_4 + q_3)) * (-0.584507089999999896e-1) + \cos((q_4 + 2*q_3 + 2*q_2)) * 0.211758236813575085e-21 + \sin((-2*q_5 + q_4)) * (-0.867361737988403547e18) + \sin((2*q_5 + q_4)) * (0.867361737988403547e-18) + \cos((2*q_3 + 2*q_2)) * (-0.780236711999999935e4) + \sin((q_5 + q_4 + q_3)) * (0.199484800000000018e-3) + \sin((q_5 + q_4 + 2*q_3 + 2*q_2)) * (-0.184197223680000180e-3) + \sin((-q_5 + q_4 + 2*q_3 + 2*q_2)) * 0.184197223680000180e-$$

$$3 + \cos(q_5) * (-0.204647935999999913e-3) + \sin((-q_5 + q_4 + q_3 + 2*q_2)) * 0.199484800000000018e-3 + \sin((q_5 + q_4 + q_3 + 2*q_2)) * (-0.199484800000000018e-3) + 0.114621301634378825e1 + \cos((2*q_5 + 2*q_4 + 2*q_3 + 2*q_2)) * (0.155137072200000389e-3) + \cos((-2*q_5 + 2*q_4 + 2*q_3 + 2*q_2)) * (-0.155137072199999955e-3) + \cos(q_3) * 0.633093845285000012e0 + \sin(q_3) * (-0.136782000000000006e-3) + \cos(q_4) * (-0.211758236813575085e-21) + \sin(q_4) * (-0.539713217244000010e-1) + \cos((2*q_2)) * 0.708461872500000034e0 + \sin((-q_5 + q_4 + q_3)) * 0.199484800000000018e-3 + \sin((q_5 + q_4)) * (-0.184197223680000180e-3) + \sin((-q_5 + q_4)) * 0.184197223680000180e-3 + \cos((q_3 + 2*q_2)) * 0.633093845285000012e0 + \sin((q_3 + 2*q_2)) * (-0.136782000000000006e-3) + \cos((2*q_5)) * 0.310274144400000344e-3 + \sin((q_4 + 2*q_3 + 2*q_2)) * (0.539713217244000010e-1) + \sin((q_4 + q_3 + 2*q_2)) * (-0.584507089999999896e-1) + \sin((2*q_5 + q_4 + 2*q_3 + 2*q_2)) * (0.867361737988403547e-18) + \sin((-2*q_5 + q_4 + 2*q_3 + 2*q_2)) * (0.867361737988403547e-18) + \cos((2*q_4 + 2*q_3 + 2*q_2)) * (0.550305141784999952e-2)$$