machines

MDPI

Article

Real-Time Sliding Mode and Moving Sliding Mode Control of 3-DOF Linear Parallel Robot

Muhammet Aydın

Department of Mechatronics Engineering, Faculty of Engineering, Firat University, Elazig 23119, Türkiye; muhammeta@firat.edu.tr

Abstract: In this study, the design stages of a parallel robot with 3-DOF and linear motion are presented in detail. The position control of the linear robot, which was manufactured after the design process, was achieved in real time by using the inverse kinematic equations of the robot with the SMC method and the MSMC method. The highlights of the study are that the linear robot designed for a specific purpose has a unique design and the control of a 3-DOF linear robot with a MSMC method, which has not been found so far in the literature. In order to realize real-time position control, real-time SMC is implemented in the Matlab/Simulink R2017b environment. In addition, to increase the success of the SMC, the position control of the robot was also realized by a different method of moving the sliding surface. Humusoft MF634 card is compatible with Matlab and was used to transfer data in real time. Considering the point where the robot will be fixed, control was provided for the third region of the x-y cartesian coordinate system for position control. According to the results, the robot reached the desired reference values faster in the MSMC, with a difference of approximately 2 s. The robot reached the targeted values in 9.492 s with a maximum error of 0.06%, showing the effectiveness of the control mechanism. Considering the large size of the robot limbs, this result is satisfactory.

Keywords: real-time control; linear parallel robot; linear delta; position control; sliding mode control; moving sliding mode control

check for updates

Academic Editor: Dan Zhang

Received: 4 February 2025
Revised: 25 February 2025
Accepted: 26 February 2025
Published: 27 February 2025

Citation: Aydın, M. Real-Time Sliding Mode and Moving Sliding Mode Control of 3-DOF Linear Parallel Robot. *Machines* **2025**, *13*, 190. https://doi.org/10.3390/machines13030190

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Despite being used for millennia, parallel architecture has just lately been applied to robot manipulators. In this concept, several kinematic chains connect a stationary platform and a moving platform. These kinematic chain extensions move simultaneously, enabling the movable platform. Moreover, the load carried by the moving platform is maintained by several kinematic chains. It is known as a parallel architecture because of this layout. Unlike series manipulators, which have an open kinematic chain, parallel manipulators have a closed one. There are several advantages and disadvantages to parallel manipulators. Singularities, low skill capabilities, limited working volume, and difficult direct kinematic solutions are some of the drawbacks. The exceptional stiffness, minimal moving component inertia, lightweight construction, extraordinary precision, remarkable agility, and ease of handling inverse kinematics are only a few of the many advantages of parallel robots. The inverse kinematic solution simplifies robot control, making it easy to implement and administer [1].

If constructed correctly, the relative workspace of parallel robots can offer a significant safety benefit. Parallel robots demonstrate secure movement, and as the robots move closer to preset places, the actuators' forces increase. In contrast, the dangerous aspect is

introduced by the singularities and extremely high joint speeds that serial robots are prone to. Additionally, parallel robots provide the same sensitivity at a lower cost than serial robots. Serial robots may be challenging to utilize in confined locations since they require a greater area to move [2–5].

Usually, three-DOF manipulators are used, particularly in pick-and-place and machine operations scenarios. One famous example of a three-DOF robot is the Delta robot, which was developed by Clavel at the Ecole Polytechnique Lausanne [6]. Another member of this family is the three-DOF Neumann's Tricept robot [6]. Another prominent member of this group is the Orthoglide robot, which was created especially for applications involving machine-vehicle use. One of the notable characteristics of the Orthoglide robot is its consistent kinematic behavior within the relevant work area.

The Delta robot is one kind of parallel manipulator that has gained a lot of popularity recently all around the world. The system can move a platform, known as the end effector, to any desired location (x, y, and z) inside the working area thanks to three motors mounted on a stationary base. Each motor's matching arms are connected to the end effector and powered by parallelogram mechanisms. The fixed and movable platforms will always be parallel to one another thanks to this arrangement.

The Delta robot's technology allows for rapid acceleration and precise placement with minimal motor action. The overall rigidity of the system is determined by the joint clearances and the stiffness of the selected limbs. However, it is crucial to keep in mind that even minor differences in manufacturing quality could have an impact on the accuracy and repeatability of the system. Therefore, lowering tolerances and upholding high production standards are essential to achieving consistent and accurate performance.

The linear parallel robot is a scaled-down, three-DOF version of the Delta robot. The linear parallel robot was designed by Herve [6], and during rotation, each arm restricts one of the platform's two DOF. The three limbs of the linear parallel robot are moved by linear motors [6].

The synthesis of parallel manipulators, the robot's forward and inverse dynamics, dynamic simulations, robot optimization, and the matrix link between the robot's kinematics and on-board dynamics have all been the subject of prior research in the literature. The literature on parallel robots includes applications of model-free iterative learning control with non-repetitive trajectories for second-order MIMO nonlinear systems to the Delta robot, kinematic modeling and characterization of soft parallel robots, and studies on the development and analysis of the working volume of planar parallel robotic motion-based intelligent actuation. Intelligent controls have been developed for the Delta robot using recurrent fuzzy neural network descriptors and single-neuron PID algorithms based on fuzzy set-based methodology. These methods have been used to build robust control that is most appropriate for the uncertainty of the Delta robot. Some studies are purely theoretical, while others are simulation-based. There has not been much research conducted on linear parallel robot real-time control. None of these studies employed real-time SMC [7–32].

Within the scope of this study, the design stages, manufacturing stages, and position control of the 3-DOF linear parallel robot with a working volume of 20 $\times$ 20 $\times$ 20 cm have been carried out in real time with the SMC method and MSMC method. The linear parallel robot, which has been manufactured, does not have any previous analog and has a completely unique design. The coordinates of the gripper end of the robot have been controlled in real time with SMC and MSMC methods, which have not been seen in the literature before. The SMC and MSMC methods are implemented by utilizing the inverse and forward kinematic equations of the robot. This developed robot can be used in applications requiring high accuracy, in certain applications by adding to the ends of serial manipulators, in milling machines requiring high speed and high precision, in

motion simulations, in pick-and-place operations, in precise positioning applications, and in sorting, transport, and packaging processes in the food, pharmaceutical, electronics, and chemical industries.

2. Design and Manufacturing of 3-DOF Linear Parallel Robot

A 3-DOF parallel robot with a working space of 20 × 20 × 20 cm was designed. The front, top, and bottom views of the robot designed for this purpose are shown in Figure 1.

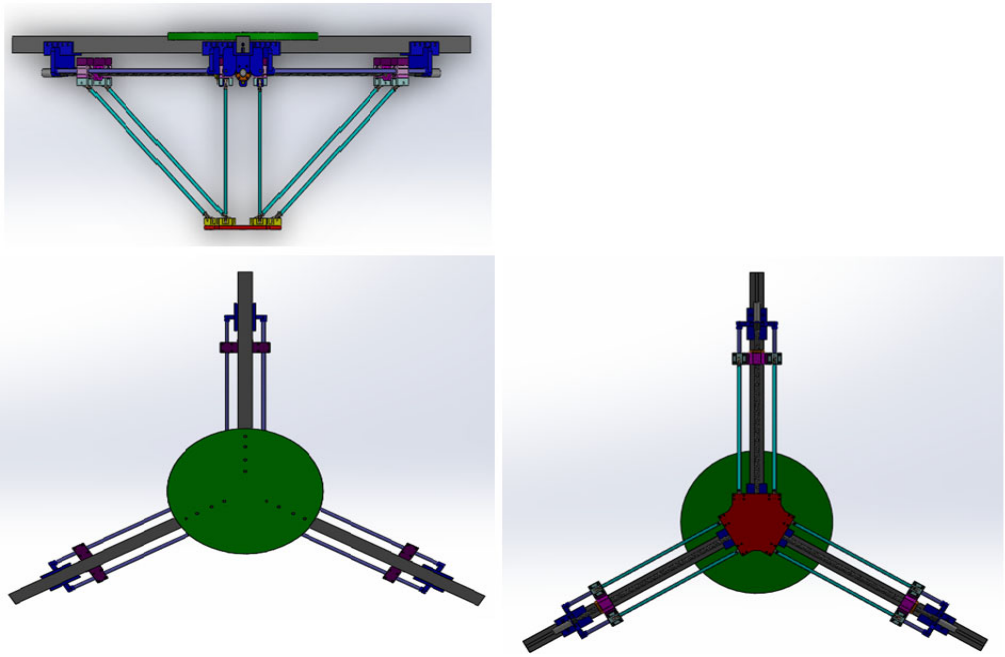


Figure 1. Front, top, and bottom views of the linear parallel robot design.

While designing the robot, the dimensions of elements 2, 8, 9, and 12 of the three-dimensional design of the parallel robot shown in Figure 2 were decided, in order to provide movement in the desired working space.

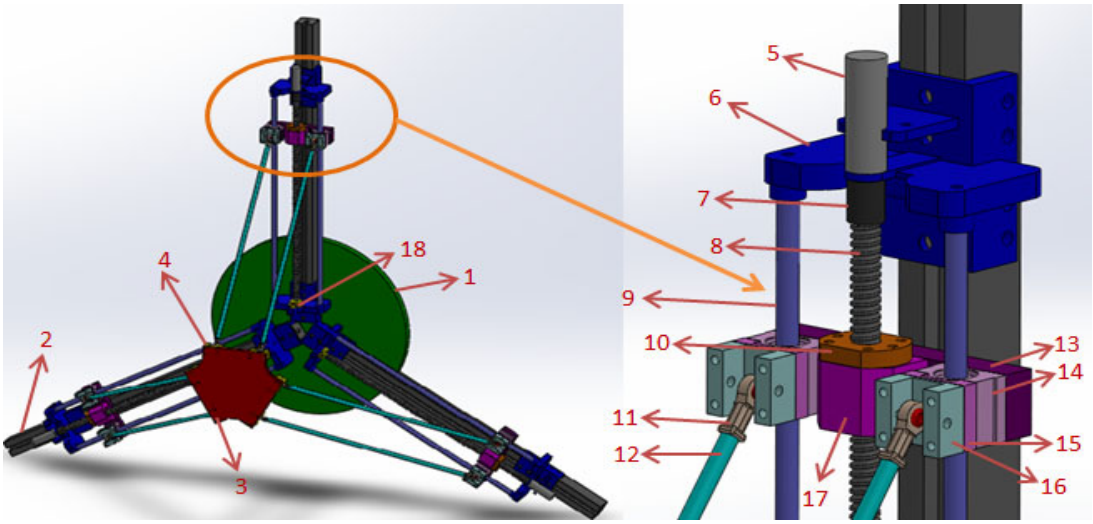


Figure 2. 3-DOF linear parallel robot solid model and its parts.

The names of the parts of the 3-DOF parallel robot seen in Figure 2 are detailed in Table 1 according to numbering.

Table 1. Part list.

Number	Part Name	Number	Part Name
1	Part holding the fixed parts	10	Ball screw nut
2	Fixed part	11	Spherical joint
3	End effector	12	Joint connection rod
4	Lower joint connection element	13	Unifying part
5	Motor	14	Shaft bearing element
6	Motor stabilizer part	15	Upper joint connection element-1
7	Coupling	16	Upper joint connection element-2
8	Ball screw	17	Ball screw-bearing nut
9	Linear shaft	18	End bearing element

Part 9 is a linear shaft that is 12 mm in diameter and 550 mm in length. The ball screw, part number 8, is 550 mm long, 16 mm in diameter, and 5 mm in pitch. Aluminum rods that were 520 mm long and 10 mm in diameter were used for part 12's parallelogram arms. The parts that would be employed were chosen based on their dimensions. The dimensions of part number 1, the end effector of the robot, lower joint connection element, upper joint connection element-1, and upper joint connection element-2 were determined through the completion of three-dimensional part designs.

After the completion of the design process, the production of the 3-DOF parallel robot was started. Firstly, the parts numbered 4, 5, 7, 10, 11, 14, and 17, whose dimensions were determined, were supplied. Afterward, the parts numbered 3, 13, 15, and 16, which were to be produced using plexiglass material in the three-dimensional CNC machine, were saved and produced as the dxf file extension. Some parts needed were produced from PLA material by a 3D printer. The parts of the ball screw, linear shaft, and parallelogram bars were machined on the lathe and made ready for the desired suitable dimensions. Both ends of the ball screws to be used were machined on the lathe to reduce the diameters suitable for the end bearing and coupling. In order to ensure proper bolt connections, both ends of the linear shafts were threaded with the help of a tap. Similarly, both ends of the parallelogram rods were threaded to provide spherical joint connections. After all the parts needed in the production phase were finalized, the assembly process was started. In the first stage, three fixed aluminum profiles were mounted to the fixed part holder, which forms the bottom base of the 3-DOF parallel robot, with appropriate bolt connections. In the next stage, the ball screw and end-bearing connections of each arm of the parallel robot were made. For a part designed to provide motor, coupling, and ball screw connections, PLA material was produced on a three-dimensional printer. Similarly, in order to ensure the bearing on the ball screws, a special part to fix the bearing elements was designed and produced on a three-dimensional printer. In the last stage, the connections between the joints of the gripper end of the parallel robot and the joints of the sliding elements on the ball screw were terminated using parallelogram bars. In order to prevent slipping, fiber nuts were used for better fixation in the connections. Thus, the production process of the 3-DOF parallel robot was finalized. Figure 3 shows an image of the fabrication of the linear parallel robot. The brands and models of the motors, gearboxes, and encoders used in the linear parallel robot system shown in Figure 3 are Maxon DCX22L, Maxon GPX22, and Maxon ENX16, respectively. The Maxon GPX22 is added to the front of the DC motor (Maxon DCX22L) to increase the amount of torque. Similarly, Maxon ENX16 is mounted on the back of the DC motor and selected to read the rotation of the motor instantaneously and transfer it to the interface program with the encoder blocks of the board in Matlab/Simulink R2017b via Humusoft MF634 data acquisition card (Humusoft s.r.o. Company, Prague, Czech Repuclic). Figure 4 shows the experimen-

tal setup of the robot, the hardware used, the hardware connection cables, and the computer interface.

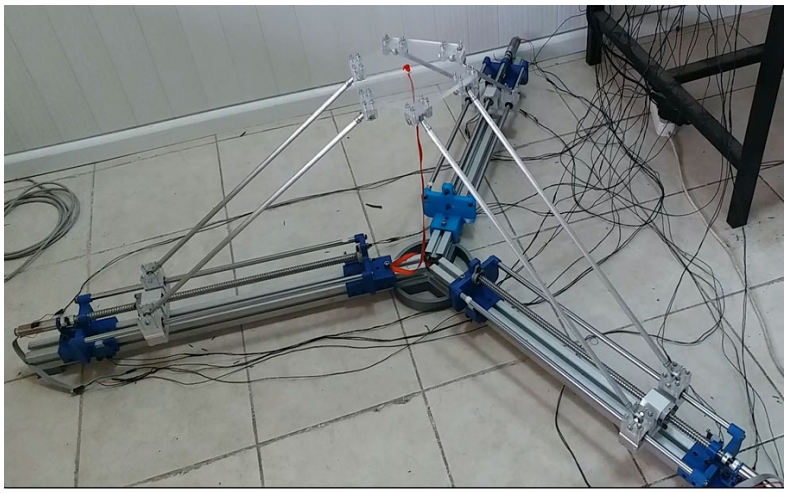


Figure 3. A final manufactured version of the linear parallel robot.

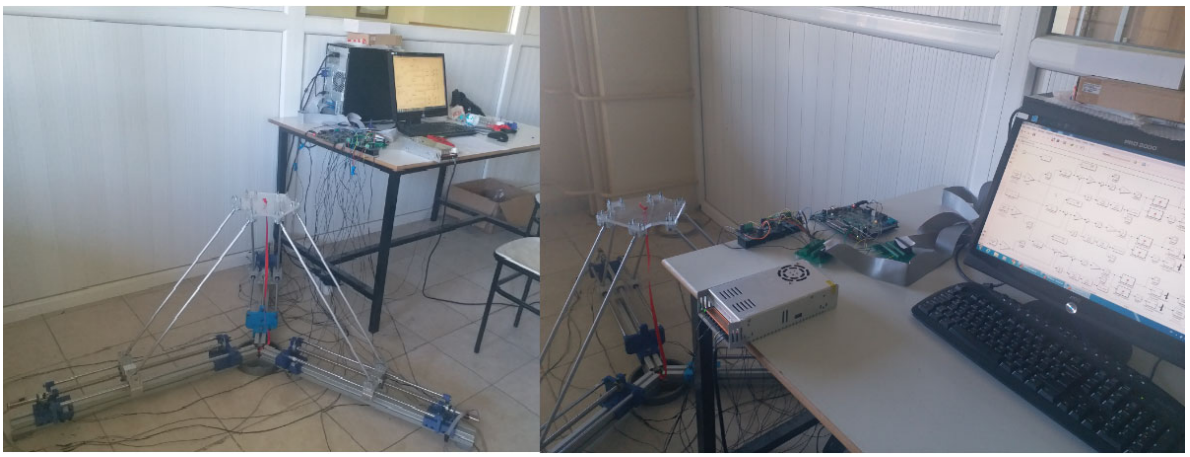


Figure 4. Experimental setup of the linear parallel robot.

3. Control of the 3-DOF Linear Parallel Robot with SMC and MSMC

SMC is a dependable nonlinear control method. It is not impacted by external disturbances like other control systems are. The vibrations on the sliding surface allow it to quickly and precisely approach the intended reference [33]. In scenarios when the system's parameters are dynamic or unpredictable, or when the system is impacted by external disturbances, SMC provides long-term controllability when the boundary values of the system are known.

Finding the sliding surface's position and creating a control rule that guarantees reaching it are essential steps before putting SMC into practice. The phase trajectory section during this operation is called the reaching mode, and the reaching time indicates the amount of time needed to reach the sliding surface. The plant is vulnerable to outside noise and parameter uncertainties while in the reaching mode [34]. The system enters the sliding mode after reaching the sliding surface. In this mode, unpredictable characteristics and outside factors have no effect on the system's course. When SMC is used, the system's high-frequency dynamics—which show up as chattering—become visible. The oscillations close to the equilibrium point that the system is attempting to reach are the cause of this phenomenon.

Use the sign function to control expression in sliding mode.

$$u = -K \operatorname{sign}(S) \tag{1}$$

can be expressed as the slip surface function (S), which in this instance is given by Equation (1) and depends on the error (e) and how it changes over time (de) in relation to the response of the system.

$$S = Ce + de \tag{2}$$

Stability theory has a very important role in control applications. There are many methods for analyzing the control stability of dynamic systems. The Lyapunov stability theorem, which is the most preferred method, is used to analyze the stability of the control method.

The sliding surface given in Figure 5 is defined by Equation (2). It consists of error and derivative of error variables. During runtime, S is positive if the state is above S = 0. Therefore, $\dot{S}$ must be negative so that the state variables are forced towards the sliding surface S = 0. When the state is below S = 0, S is negative. Therefore, $\dot{S}$ must be positive so that the state variables are forced towards the sliding surface (S = 0). For this reason, the following inequality exists in both cases [35].

$$S\dot{S} < 0 \tag{3}$$

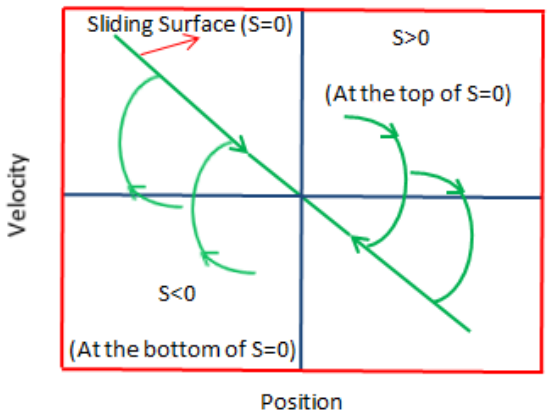


Figure 5. The sliding surface in the phase plane.

A sufficient condition for the system to be at the equilibrium point is the existence of a Lyapunov function $V(x)$ satisfying the following properties. $V(x) > 0$ must be locally positive definite, and $\dot{V}(x) \leq 0$ must be locally negative definite. In this study, the Lyapunov function $V(x)$ is chosen as follows.

$$V(x) = \frac{1}{2}\dot{S}^2 \tag{4}$$

The chosen Lyapunov function depends on the sliding surface (S) and will always be positive whether S is negative or positive. Therefore, the condition V(x) > 0 is always satisfied. Since S varies with time, it is a differentiable expression with respect to time. If the derivative of V(x) is obtained, the following expression is derived.

$$\dot{V}(x) = S\dot{S} \tag{5}$$

It was previously stated that $S\dot{S} < 0$ due to the condition of forced orientation to the sliding surface (S = 0) due to the structure of the sliding mode control. In this case, $\dot{V}(x) < 0$

is always satisfied. Since both sufficient conditions are fulfilled, it is concluded that the control system is asymptotically stable [35].

Equation (2) is used to calculate the sliding surface in the SMC method. When the SMC method is applied, K and C values are taken as −1 and −15, as can be seen in Figure 4. In the MSMC method, the value of C, which is the slope of the sliding surface, is preferred as a variable depending on the error and the derivative of the error as expressed in Equation (6). In this way, the slope of the sliding surface of the MSMC is not a constant value but constantly changing. The constant values K, C_1, and C_2 in the MSMC structure are taken as −1, 10, and −12 respectively.

$$C = C_1 - C_2(de/e) \tag{6}$$

A chattering control signal is produced by the sign function in the SMC expression. Several functions are employed in place of the sign function to address this issue. In this study, the popular saturation function is applied in place of the sign function in the MSMC method. The control signal with saturation function utilized in the MSMC structure is depicted in Equation (7).

$$u = -K\,\mathrm{sat}(S) \tag{7}$$

To generate the control signal, it is necessary to generate the error and the derivative of the error for each arm of the robot from the reference values and encoder data received from the motors. For this purpose, depending on the limb dimensions of the designed robot, it is necessary to know how the progress in each arm on the horizontal affects the gripper end of the robot. For this reason, the kinematic equations of the linear robot have been obtained in previous studies. The kinematic equations for the 3-DOF linear parallel robot have been obtained in previous research [36]. If this reference is examined, it can be seen in detail how the inverse and straight kinematic equations of the linear parallel robot are calculated. The inverse kinematic equations of the robot, which are obtained depending on the gripper tip and fixed limb values, giving the horizontal progress in each arm are presented below. These horizontal displacements determine how long the motor in each arm will run.

$$q_1 = \sqrt{L_2^2 - y_p^2 - z_p^2} + x_p + L_3 \tag{8}$$

$$q_2 = \sqrt{L_2^2 - z_p^2 - \frac{3}{4}x_p^2 - \frac{\sqrt{3}x_p y_p}{2} - \frac{1}{4}y_p^2} + \left(\frac{2L_3 - x_p + \sqrt{3}y_p}{2}\right) \tag{9}$$

$$q_3 = \sqrt{L_2^2 - z_p^2 - \frac{3}{4}x_p^2 + \frac{\sqrt{3}x_p y_p}{2} - \frac{1}{4}y_p^2} + \left(\frac{2L_3 - x_p + \sqrt{3}y_p}{2}\right) \tag{10}$$

The parameters in Equations (8)–(10) are shown on the robot in Figure 6.

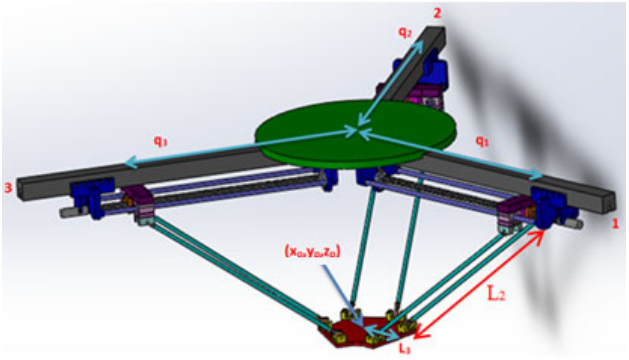


Figure 6. Robot geometric parameters.

The forward kinematic equations of the robot, which are obtained depending on the fixed limbs of the robot, q_1, q_2, and q_3, using the inverse kinematic equations of the robot, are given below. These inverse and forward kinematic equations were used during the control of the robot to convert the robot gripper tip reference values into horizontal motion and to translate the realized horizontal motion into robot gripper tip coordinates. The block diagrams of the SMC and MSMC implemented in Matlab/Simulink R2017b real-time toolbox environment using these equations are given in Figures 7 and 8.

$$x_p = \frac{(q_2 - q_1)(q_2 + q_1 - 2L_3) + \sqrt{3}y_p(L_3 - q_2)}{(3L_3 - 2q_1 - q_2)} \tag{11}$$

$$y_p = \frac{(3L_3 - 2q_1 - q_2)(q_3 - q_1)(q_3 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)} - \frac{(3L_3 - 2q_1 - q_3)(q_2 - q_1)(q_2 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)} \tag{12}$$

$$\begin{aligned} z_p^2 = {} & L_2^2 - L_3^2 - 2(L_3 - q_1)\Bigg(\frac{(q_2 - q_1)(q_2 + q_1 - 2L_3)}{(3L_3 - 2q_1 - q_2)} - \frac{\sqrt{3}(q_2 - L_3)\left(\frac{(3L_3 - 2q_1 - q_2)(q_3 - q_1)(q_3 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)}\right)}{(3L_3 - 2q_1 - q_2)} \\ & + \frac{\sqrt{3}(q_2 - L_3)\left(\frac{(3L_3 - 2q_1 - q_3)(q_2 - q_1)(q_2 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)}\right)}{(3L_3 - 2q_1 - q_2)}\Bigg) + 2q_1L_3 - q_1^2 - \Bigg(\frac{(q_2 - q_1)(q_2 + q_1 - 2L_3)}{(3L_3 - 2q_1 - q_2)} \\ & - \frac{\sqrt{3}(q_2 - L_3)\left(\frac{(3L_3 - 2q_1 - q_2)(q_3 - q_1)(q_3 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)}\right)}{(3L_3 - 2q_1 - q_2)} + \frac{\sqrt{3}(q_2 - L_3)\left(\frac{(3L_3 - 2q_1 - q_3)(q_2 - q_1)(q_2 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)}\right)}{(3L_3 - 2q_1 - q_2)}\Bigg)^2 \\ & - \Bigg(\frac{(3L_3 - 2q_1 - q_2)(q_3 - q_1)(q_3 + q_1 - 2L_3) - (3L_3 - 2q_1 - q_3)(q_2 - q_1)(q_2 + q_1 - 2L_3)}{\sqrt{3}(L_3 - q_3)(3L_3 - 2q_1 - q_2) - \sqrt{3}(q_2 - L_3)(3L_3 - 2q_1 - q_3)}\Bigg)^2 \end{aligned} \tag{13}$$

The block design for the 3-DOF linear parallel robot's implementation of real-time SMC is shown in Figure 7. This program was created using the real-time toolbox in the Matlab/Simulink environment. To facilitate data transfer, the Humusoft MF634 data gathering card—which is compatible with Matlab—was utilized. The encoder data has been moved to the software via the encoder blocks on the card. The sampling rate was used as 0.001. PWM (Pulse-Width Modulation) signals were sent to each motor via digital output blocks. The PWM blocks used a PWM frequency of 4000 Hz. Because of the card's features, the PWM signals were created between 10% and 90%, and the saturation block was used to achieve this. The SMC, the 3-DOF linear parallel robot, and the actual hardware parts of the system could all be controlled and communicated with in real time thanks to these parts and configurations.

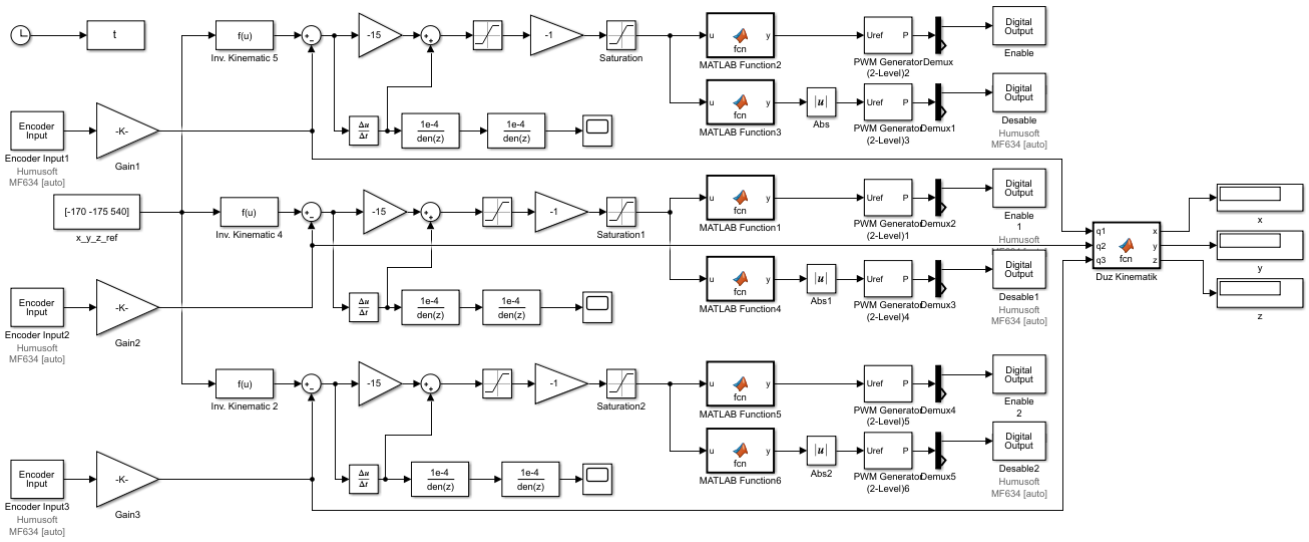


Figure 7. SMC Matlab/Simulink real-time interface program for the 3-DOF linear parallel robot.

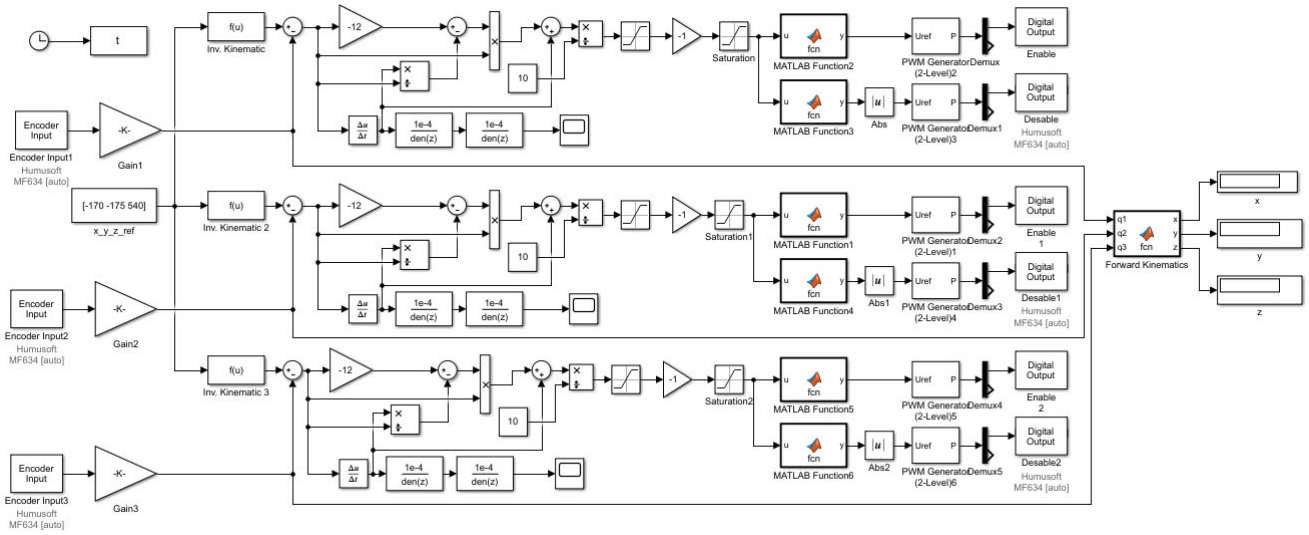


Figure 8. MSMC Matlab/Simulink real-time interface program for the 3-DOF linear parallel robot.

The internal logic of the interface software is as follows. First, reference points are established for the robot's end effector. Once the program is complete, the distances of each motor with the identifiers q_1, q_2, and q_3 are determined using inverse kinematic equations. To give these values, the encoders provide feedback and the motors are activated. The difference between the reference values and the input feedback is sent to the sliding mode controls. The PWM signals required by motors are produced in this way. When the relevant reference levels are attained, the robot stops completely. The reference values employed in this inquiry are x = −170 mm, y = −175 mm, and z = 540 mm, as shown in Figures 7 and 8.

Because there are three motors in the system, each motor has its own SMC mechanism. The SMC structures of the motors are the same since they are identical. The same methods were applied in the moving sliding mode control application in Figure 8.

4. Results and Discussion

The SMC method was applied on the first motor by determining the distance that the part on the ball screw should cover by rotating the motor with the help of the inverse kinematic equations of the robot according to the reference values x = −170 mm, y = −175 mm, and z = 540 mm. These reference values represent one of the far distance points in the robot's workspace. In Figure 9, the drive amount of the first motor (q_1) to meet the determined references are presented together in order to compare the results obtained with SMC and MSMC methods. "Drive amount of the first motor" refers to the progression of the elements on the ball screw driven by the first motor. In other words, the distance moving on the ball screw given in Figure 6 is equivalent to q_1. In the naming of Figure 9, q_1 is expressed in brackets. As a result of the analyses, it was observed that while the feed rate of the first motor reached 57.55 in 1.418 s in the SMC method, this value reached 57.61 in 1.25 s in the MSMC method. Accordingly, it is determined that the MSMC method provides the targeted feed amount of the first motor on the ball screw in a shorter time compared to SMC.

For SMC and MSMC control, Figure 10 displays the PWM signals given to the first motor in order to produce the intended reference value (q_1) on the ball screw. The graphs show that the PWM signal in the SMC approach continuously varies between 0.9 and −0.9. The so-called chattering in the motor was brought on by this circumstance. The PWM signal in the MSMC technique is known to be fixed at 0.01277, which is between 1.258 s and nearly zero. The findings show that the MSMC approach takes less time and produces the desired reference value more silently.

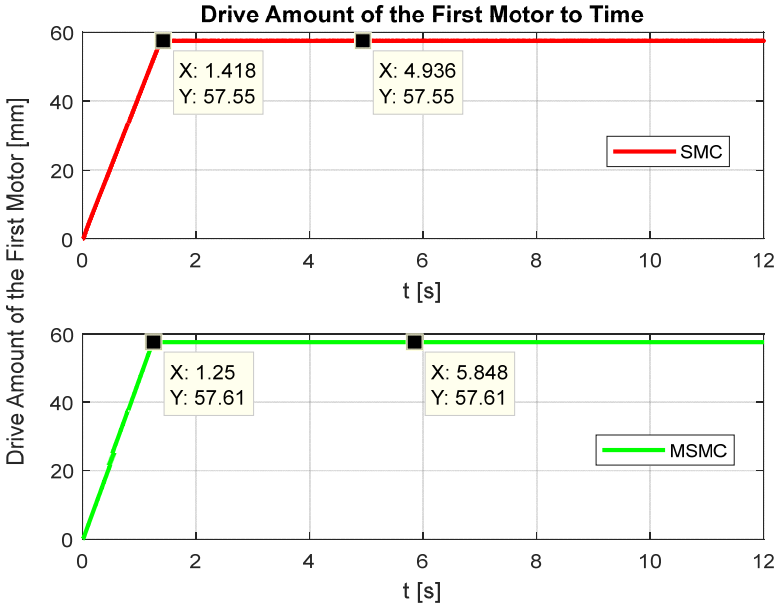


Figure 9. Drive amount of the first motor (q_1).

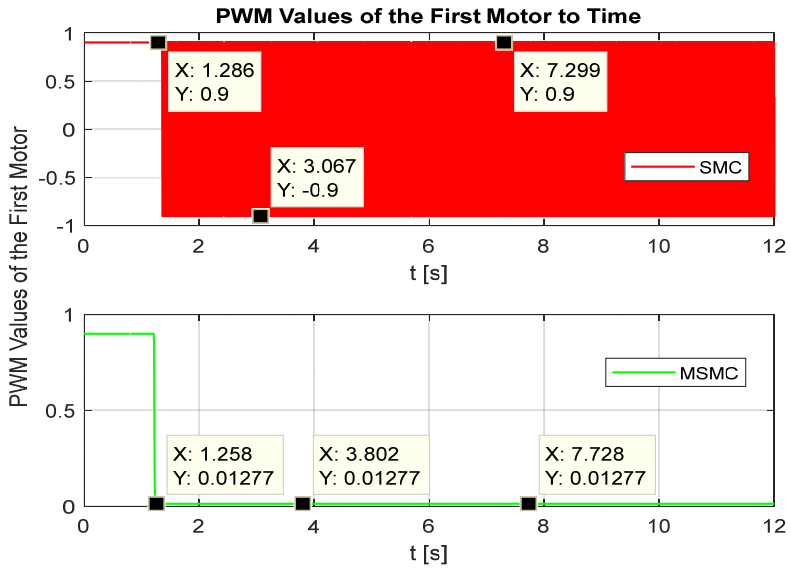


Figure 10. PWM values of the first motor to time.

In the second motor, SMC is used, which involves the calculation of how much the motor should move for the distance (q_2) to be reached on the ball screw according to the given reference values. Figure 11 shows the result graphs of SMC and MSMC control for the second motor to provide the progress (q_2) on the ball screw. How much progress the second motor will provide is calculated with the help of the inverse kinematic equations of the robot using the reference values desired to be traveled by the end effector of the robot. It is understood from the graphs that the progress value on the second screw shaft of 3.256 mm is reached in 0.2074 s in the SMC method, and the progress value on the second screw shaft of 3.194 mm is reached in 0.0994 s in the MSMC method. From these results, it is observed that a more successful result is obtained in the MSMC method, and the desired value is achieved earlier in the MSMC method.

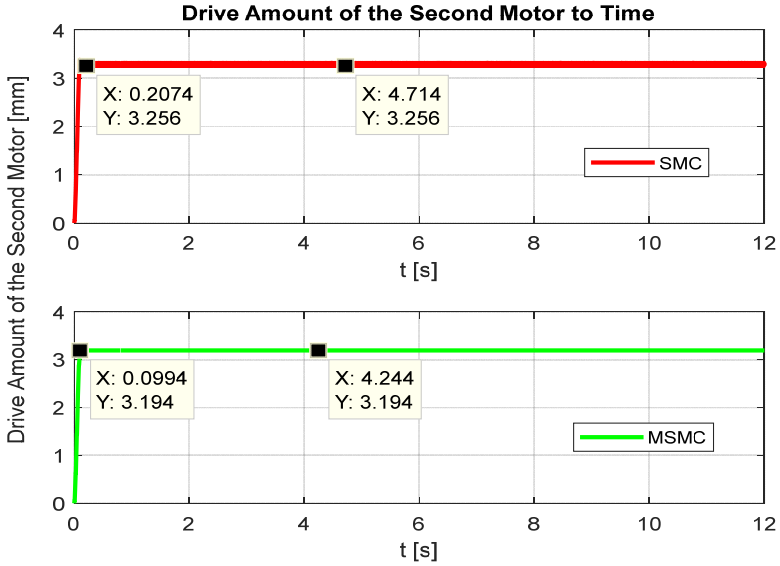


Figure 11. Drive amount of the second motor (q_2).

The second motor's PWM values for SMC and MSMC techniques are displayed in Figure 12. Analysis of the plots shows that beginning at 0.0994 s, the PWM signal in the MSMC technique stays constant at 0.00649, or roughly zero. The motor PWM values in the SMC technique fluctuate continuously between ±0.9. These findings demonstrate that the MSMC approach achieves the intended reference value without causing the motor to vibrate, whereas the SMC approach results in noise and oscillations in the motor. As a result, the motor is exposed to continuous running. The MSMC approach gave the motor faster and more comfortable control.

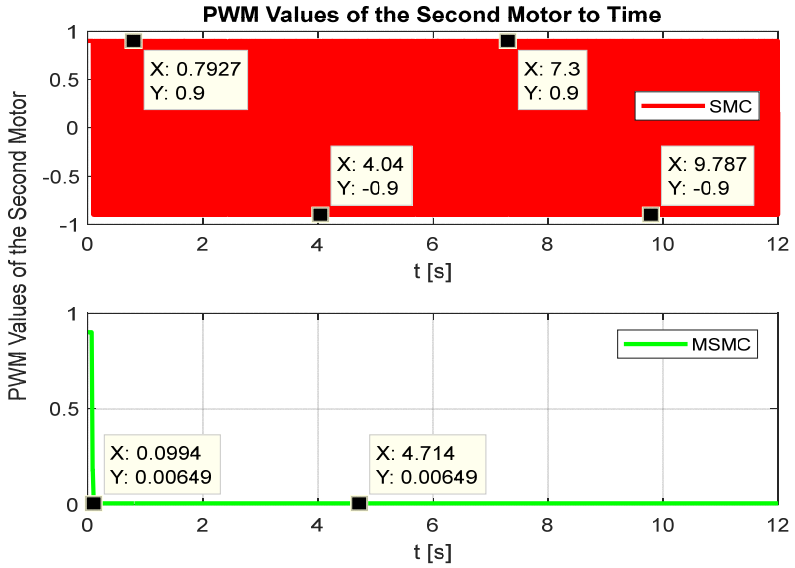


Figure 12. PWM values of the second motor to time.

In order to obtain the appropriate displacement on the third ball screw according to the reference values given for the end effector of the robot, SMC methods were also applied to the third motor in order to provide the value (q_3) calculated from the inverse kinematic equations of the robot. Figure 13 presents the graphs of the drive amount (q_3) on the ball screw shaft with respect to time as a result of the operation of the third motor. In the graphs, the results obtained after applying the SMC method and MSMC method are given together. While 450.5 mm feed value was reached in 11.65 s in the SMC method, this feed value was

reached in 11.52 s in the MSMC method. When the results are analyzed, it is obvious that the desired feed rate is reached 0.13 s earlier with the MSMC method.

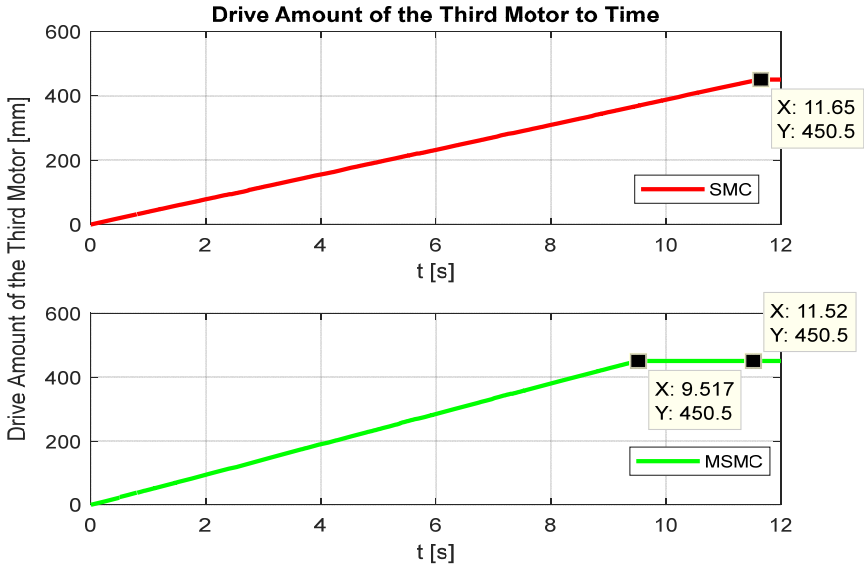


Figure 13. Drive amount of the third motor (q_3).

Figure 14 illustrates the PWM signals applied to the third motor while achieving the desired movement on the third screw shaft. The PWM signals for the SMC and MSMC methods are presented separately. When examining the PWM signals, it is observed that the signal for the SMC method oscillates between ± 0.9 from 11.54 s onwards. In contrast, for the MSMC method, the control PWM signal reaches a value of 0.00006555 after 9.517 s and stabilizes. As evident from the SMC signal, a more oscillatory behavior, characterized by vibrations and noise, is present. Consequently, the motor experiences chattering and is unnecessarily strained. On the other hand, the MSMC method achieves the desired steady, vibration-free, and noise-free state in a shorter time, thereby preventing undue stress on the motor.

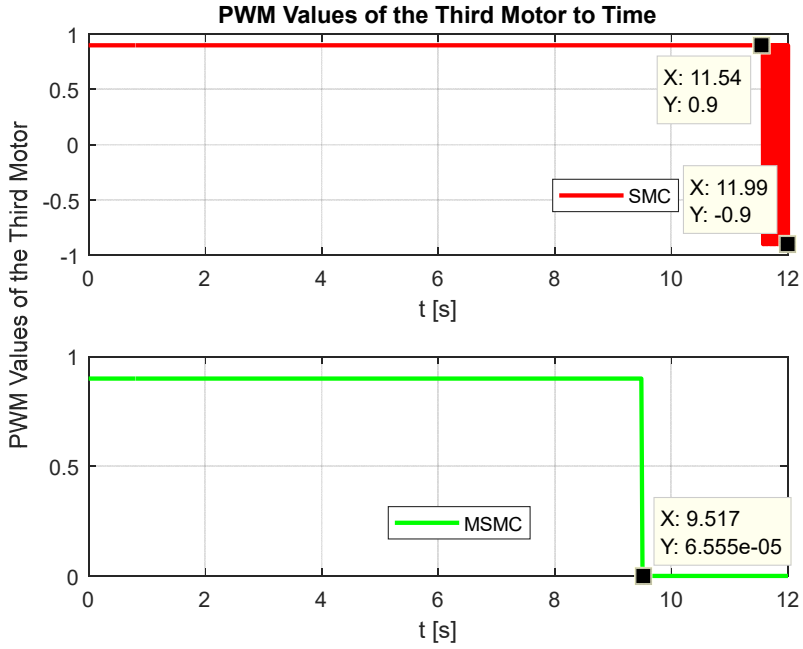


Figure 14. PWM values of the third motor to time.

Figure 15 shows the results of SMC and MSMC together for x reference value −170 mm. When the results are analyzed, it is seen that the SMC method reaches the reference value x in 11.51 s. In the MSMC method, it is understood that it reaches the x value in 9.489 s. As can be seen from the graph in Figure 16, the error amount in the SMC method is 0.01336. In the MSMC method, the error amount was obtained as 0.02905. While the percentage relative error is 0.0079 in the SMC method, it is calculated as 0.0171 in the MSMC method. As can be seen from the results, the SMC method reaches the reference value x with a smaller margin of error but has a delay of 2.021 s. Considering that the margins of error are quite small in both cases, the importance of the reaching time increases. For this reason, it can be said that MSMC provides better control since it reaches the reference value earlier with a time difference of 2.021 s.

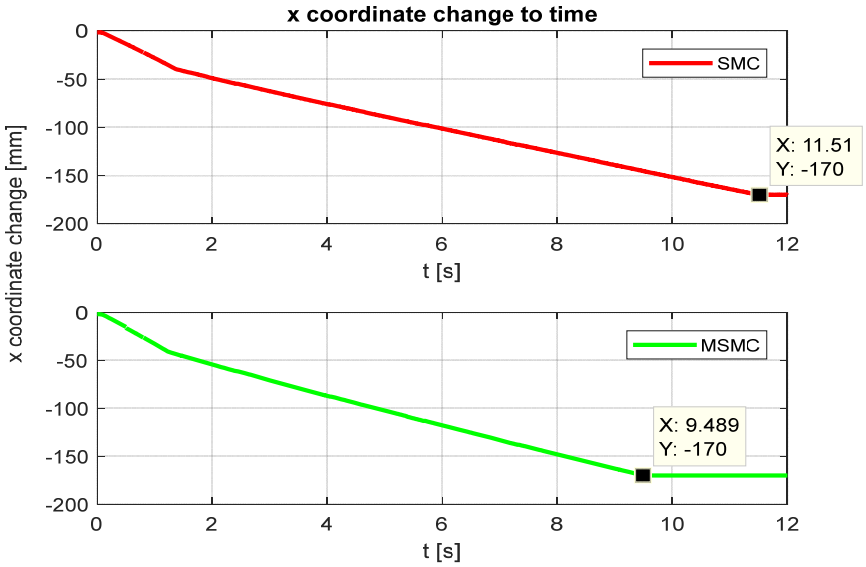


Figure 15. *x* coordinate change to time.

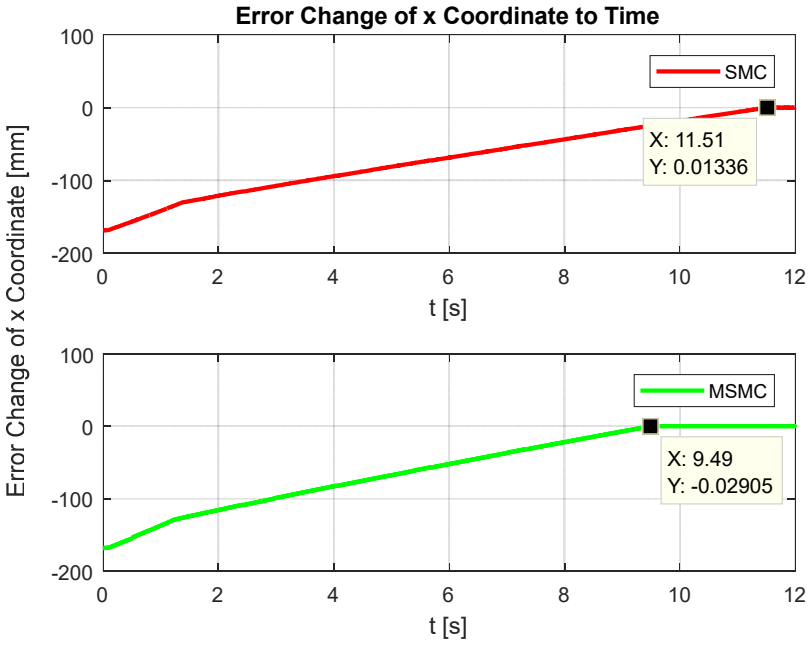


Figure 16. Error change of *x* coordinate to time.

The graphs of the y-coordinate value produced following the application of SMC and MSMC are displayed in Figure 17. The outcomes of the SMC and MSMC methods are

displayed, as the graphs demonstrate. The findings demonstrate that the SMC approach takes 11.51 s to attain the target reference y value. In the MSMC method, the reference y value is reached after 9.505 s. As can be seen from the error amounts of the y value in Figure 18, the error amount in the MSMC method is 0.09483. However, the amount of error in the SMC method was found to be 0.03224. The percentage relative error is 0.0542 in the MSMC method and 0.0184 in the SMC method. While the SMC method should be preferred in cases where a precise operation is required, the MSMC method, which fulfills the desired reference value in an early time of about 2 s with a small error excess of 0.0358 percent, should be preferred in studies where time is important. In addition, considering the situations in PWM signals, it is clear that the MSMC method produces a better result, considering that the motors do not vibrate and do not work unnecessarily.

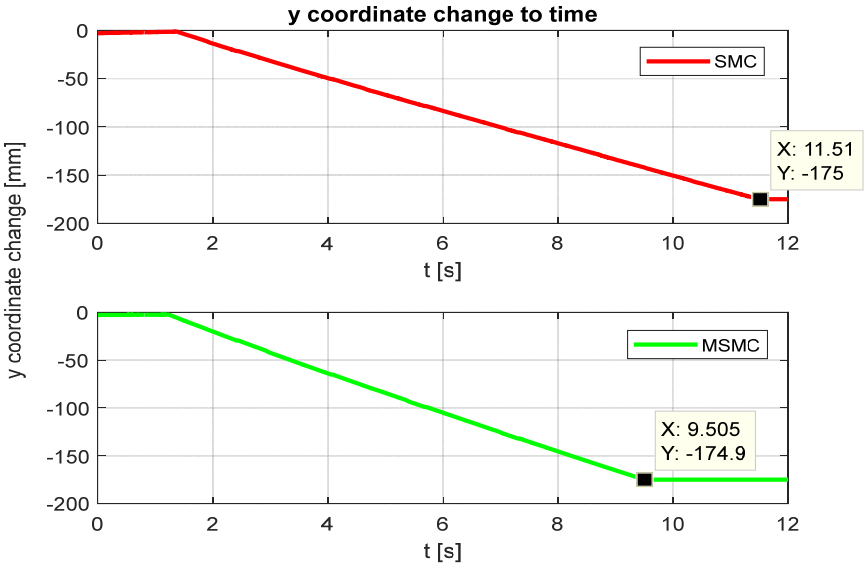


Figure 17. *y* coordinate change to time.

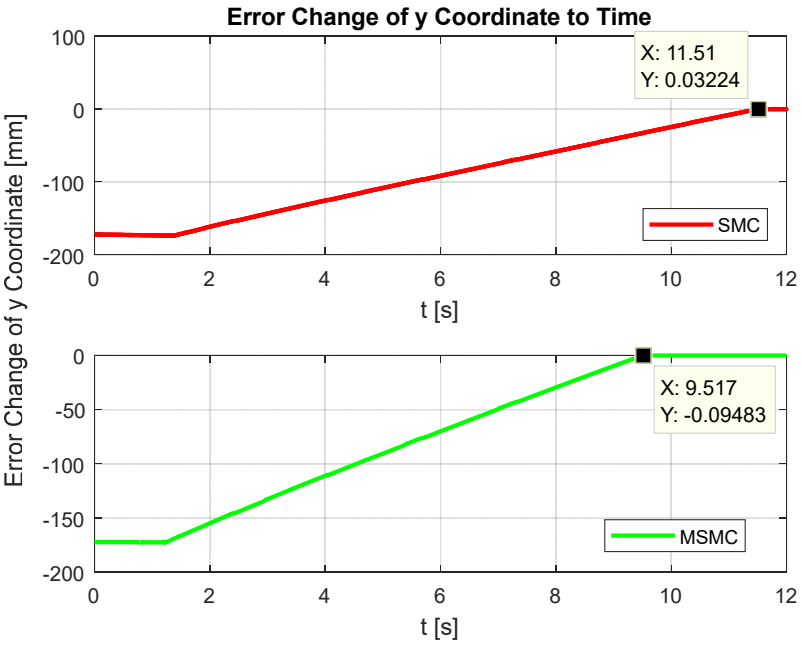


Figure 18. Error change of *y* coordinates to time.

The z-coordinate value graphs obtained after controlling the system with SMC methods are presented in Figure 19. These graphs display the z-coordinate values resulting from

the SMC and MSMC control methods. Upon examining the results, it is observed that the desired reference z value is achieved in 11.49 s with the SMC method, whereas the MSMC method reaches the desired z value in 9.502 s. When analyzing the error graph in Figure 20, it is found that the error amount for the SMC method is 0.0054, whereas the error for the MSMC method remains at 0.0021. The percentage relative errors for the SMC and MSMC methods are calculated as 0.001 and 0.0004, respectively. With a margin of error of 0.0004% and a time difference of 1.988 s, the MSMC method achieves the desired reference z value faster than the SMC method.

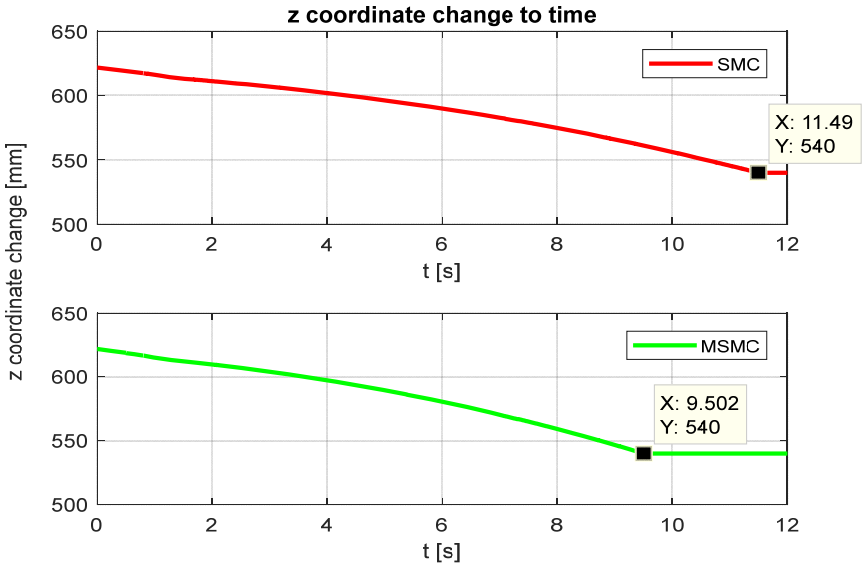


Figure 19. *z* coordinate change to time.

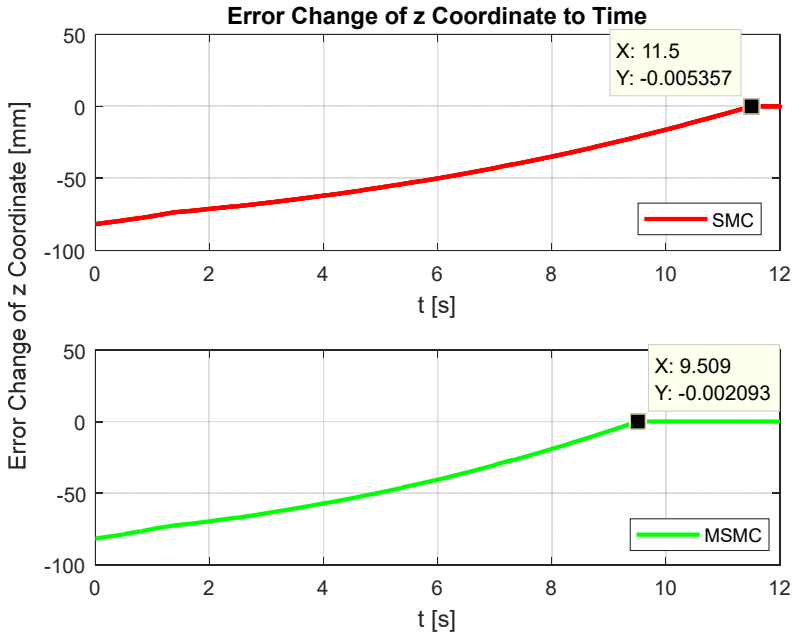


Figure 20. Error change of *z* coordinate to time.

Figure 21 shows the changes in the robot motion at one second intervals of the position control application performed on the robot as shown in Video Supplementary. At the end of about 10 s, the progress of the robot on each ball screw (q_1, q_2, and q_3) is seen to provide the desired reference value from the initial position.



Figure 21. Application images.

Figures 22–24 show the sliding surfaces of the SMC and MSMC methods applied to each engine, respectively. From the graph in Figure 22, it is seen that the sliding surface of

the SMC method is not fixed and constantly shifts between ±626 values, while the sliding surface of the MSMC method remains at a constant value. There is a similar situation in Figure 23. While the SMC method fluctuates at the same values, the sliding surface value of the MSMC method remains at a constant value. In the control of the last motor, the sliding surfaces stabilized at the end of 11.74 s for the SMC method and 9.517 s for the MSMC method.

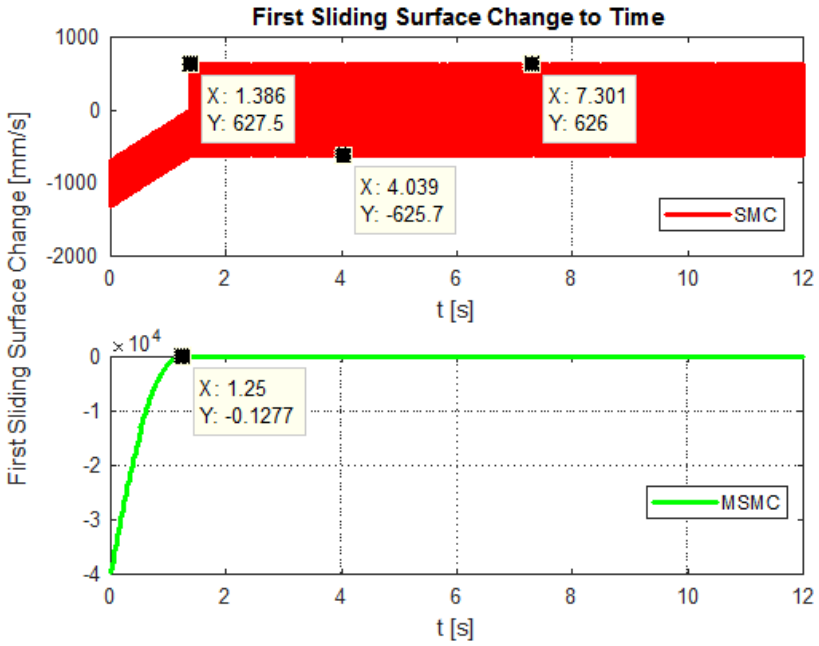


Figure 22. The first sliding surface change to time.

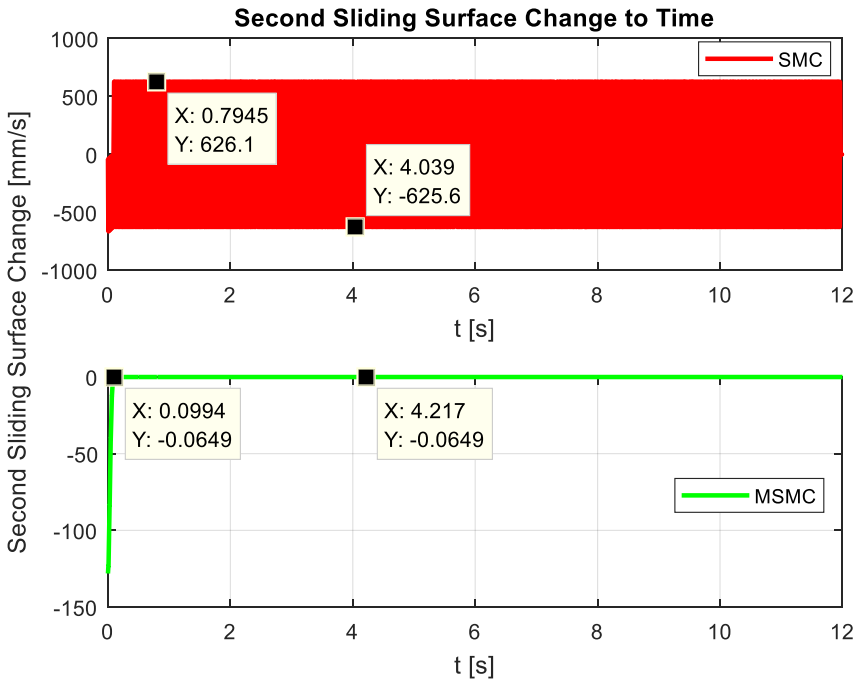


Figure 23. Second sliding surface change to time.

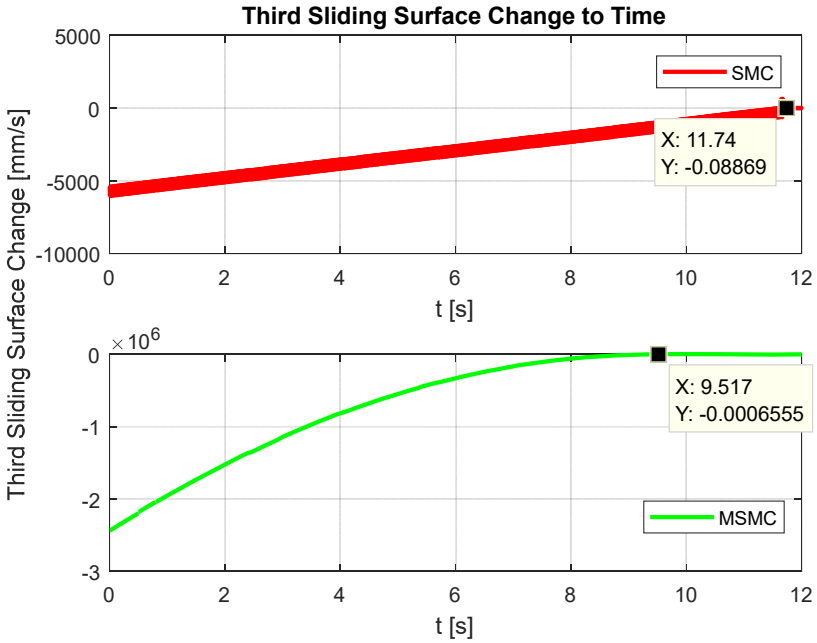


Figure 24. Third sliding surface change to time.

The robot's path from the starting position to the intended reference is depicted in Figure 25. Figure 25 displays the path taken by the robot's gripper end during the SMC and MSMC methods. The SMC method operates more noisily and vibrates more than the MSMC approach, which operates more steadily and without noise, even if the graph appears to show that the trajectories followed by these two methods are similar. Because the MSMC method enables vibration-free, noise-free control and gives the needed references sooner, better control has been realized as a result.

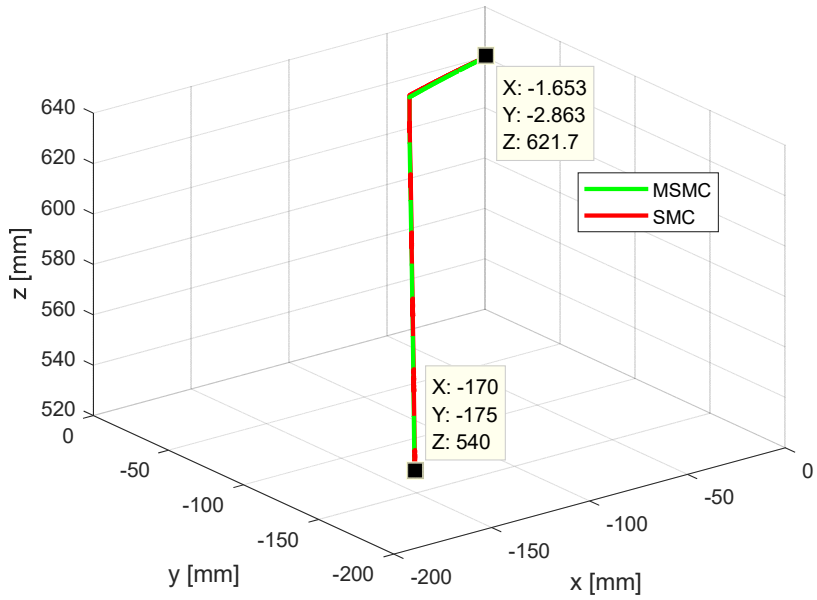


Figure 25. The trajectory of the robot while reaching the desired reference.

The trajectory of the case study for the non-linear trajectory is presented in Figure 26. It is clear from the values that the study area is the first region in the coordinate axis set.

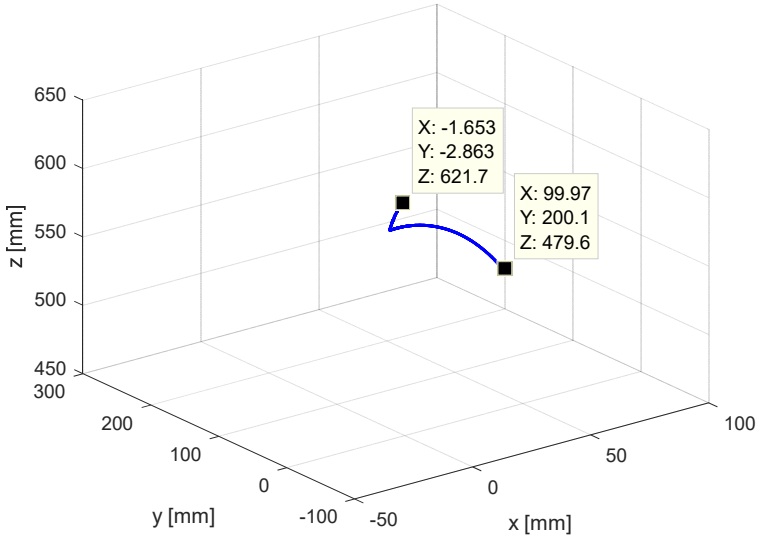


Figure 26. Nonlinear trajectory example for the system.

5. Conclusions

This study focuses on the uniqueness of the 3-DOF linear parallel robot with a working volume of (20, 20, 20) cm by taking a completely different approach and the real-time position control of the robot with MSMC, which has not been realized in the literature so far. Firstly, the three-dimensional design of the robot was realized appropriately and then the production phase was started. With the addition of motors, encoders, and data acquisition card (Humusoft MF634) hardware parts, the control of the robot was started.

The robot was controlled using two techniques. First, the Matlab/Simulink real-time toolbox was used to implement the SMC method for $(-170, -175, 540)$ mm robot control. The MSMC approach was then used to operate the robot. The interface program enabled a connection between the robot and the data acquisition card, and the PWM control signal enabled the motors to move. The inverse kinematic equations were used to create reference values q_1, q_2, and q_3 for horizontal motions. To achieve the required q_1, q_2, and q_3 values, the motors were managed using SMC and MSMC techniques. To determine the coordinate coordinates of the robot's gripper end, the forward kinematic equations were fed the actual q_1, q_2, and q_3 values that were produced as outputs. In contrast to the MSMC approach, the SMC method operates with a delay of roughly 2 s and offers more accuracy with smaller error margins. The MSMC approach, on the other hand, is a better option in time-sensitive applications since it achieves the intended reference values more quickly and with fewer errors. Furthermore, the MSMC technique improves PWM signal behavior by lowering motor vibrations, noise levels, and needless strain on the motor. As a result, it is clear that the MSMC approach yields better results than the SMC approach.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines13030190/s1, Working video of the robot.

Funding: This research received no external funding.

Data Availability Statement: All data used to support the findings of this study are included in the article.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DOF	Degree of freedom
MSMC	Moving sliding mode control
PLA	Polylactic acid
SMC	Sliding mode control

References

1. Sciavicco, L. *Modeling and Control of Robot Manipulator*; The McGraw-Hill Comp. Inc.: New York, NY, USA, 2012.
2. Merlet, J.P. On the Infinitesimal Motion of Parallel Manipulators in Singular Configurations. In Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 12–14 May 1992; Volume 1, pp. 320–325. [CrossRef]
3. Bi, Z.M.; Lang, S.Y.T. Joint workspace of parallel kinematic machines. *Robot. Comput. Integr. Manuf.* **2009**, *25*, 57–63. [CrossRef]
4. Merlet, J.P. Determination of the orientation workspace of parallel manipulators. *J. Intell. Robot. Syst.* **1995**, *13*, 143–160. [CrossRef]
5. Mustafa, M.; Misuari, R.; Daniyal, H. Forward kinematics of 3 degree of freedom delta robot. In Proceedings of the 2007 5th Student Conference on Research and Development, Selangor, Malaysia, 11–12 December 2007; pp. 1–4. [CrossRef]
6. Merlet, J.P. *Parallel Robots*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 31–34.
7. Bonev, I. ParalleMIC 2001, World Wide Web Site. Available online: www.parallemic.org/Reviews/Review002.html (accessed on 18 January 2025).
8. Stan, S.; Manic, M.; Maties, V.; Balan, R. Evolutionary Approach to Optimal Design of 3 DOF Translation Exoskeleton and Medical Parallel Robots. In Proceedings of the 2008 Conference on Human System Interactions, Krakow, Poland, 25–27 May 2008. [CrossRef]
9. Staicu, S. Dynamics analysis of the Star parallel manipulator. *Robot. Auton. Syst.* **2009**, *57*, 1057–1064. [CrossRef]
10. Sparacino, F.; Herve, J.M. Synthesis of parallel manipulators using Lie-groups Y-star and H-robot. In Proceedings of the 1993 IEEE/Tsukuba International Workshop on Advanced Robotics, Tsukuba, Japan, 8–9 November 1993; pp. 75–80. [CrossRef]
11. Wan, Y.; Wang, G.; Ji, S.; Wang, Z.; Liu, J. A Y-Star Robot: Optimal Design for the Specified Workspace. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Washington, DC, USA, 12–14 December 2008. [CrossRef]
12. Briot, S.; Pashkevich, A.; Chablat, D. Optimal Technology-Oriented Design of Parallel Robots for High Speed Machining Applications. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District, Anchorage, AK, USA, 3–8 May 2010. [CrossRef]
13. Enferadi, J.; Tootoonchi, A.A. Inverse dynamics analysis of a general spherical star-triangle parallel manipulator using the principle of virtual work. *Nonlinear Dyn.* **2010**, *61*, 419–434. [CrossRef]
14. Dolga, V.; Dolga, L. The structural synthesis of the parallel robots. *Mechanika* **2011**, *17*, 288–295. [CrossRef]
15. Huang, X.; Zhu, X.; Gu, G. Kinematic modeling and characterization of soft parallel robots. *IEEE Trans. Robot.* **2022**, *38*, 3792–3806. [CrossRef]
16. Singh, D.; Choudhury, R.; Singh, Y.; Mukherjee, M. Development and workspace analysis of smart actuation based planar parallel robotic motion stage. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *912*, 032063. [CrossRef]
17. Boudjedir, C.E.; Bouri, M.; Boukhetala, D. Model-free iterative learning control with nonrepetitive trajectories for second-order MIMO nonlinear systems—Application to a delta robot. *IEEE Trans. Ind. Electron.* **2020**, *68*, 7433–7443. [CrossRef]
18. Le, M.T.; Luong, H.T.; Tung, P.T.; Pham, C.T.; Nguyen, C.N. Evaluating the Quality of Intelligent Controllers for 3-DOF Delta Robot Control. *Int. J. Mech. Eng. Robot. Res.* **2021**, *10*, 542–552. [CrossRef]
19. Le, M.T.; Luong, H.T.; Loc, P.T.; Nguyen, C.N. Delta robot control using single neuron PID algorithms based on recurrent fuzzy neural network identifiers. *Int. J. Mech. Eng. Robot. Res.* **2020**, *9*, 1411–1418. [CrossRef]
20. Wu, L.; Zhao, R.; Li, Y.; Chen, Y.H. Optimal design of adaptive robust control for the delta robot with uncertainty: Fuzzy set-based approach. *Appl. Sci.* **2020**, *10*, 3472. [CrossRef]
21. Moustris, G.; Tzafestas, C. Modeling, Simulation and Control of the Double Delta Surgical Robot. *Machines* **2024**, *12*, 620. [CrossRef]
22. Hong, J.-H.; Lim, J.-H.; Lee, E.; Shin, D. The Workspace Analysis of the Delta Robot Using a Cross-Section Diagram Based on Zero Platform. *Machines* **2024**, *12*, 583. [CrossRef]
23. Garcia, M.; Esquen, A.-C.; Sabbagh, M.; Grace, D.; Schneider, E.; Ashuri, T.; Voicu, R.C.; Tekes, A.; Amiri Moghadam, A.A. Soft Robots: Computational Design, Fabrication, and Position Control of a Novel 3-DOF Soft Robot. *Machines* **2024**, *12*, 539. [CrossRef]
24. Huang, X.; Rendon-Morales, E.; Aviles-Espinosa, R. ROMI: Design and Experimental Evaluation of a Linear Delta Robotic System for High-Precision Applications. *Machines* **2023**, *11*, 1072. [CrossRef]

25. Portillo-Vélez, R.d.J.; Burgos-Castro, I.A.; Vásquez-Santacruz, J.A.; Marín-Urías, L.F. Integrated Conceptual Mechatronic Design of a Delta Robot. *Machines* **2022**, *10*, 186. [CrossRef]
26. Altuzarra, O.; Urizar, M.; Hernández, A.; Amezua, E. Parallel Continuum Delta: On the Performance Analysis of Flexible Quasi-Translational Robots. *Appl. Sci.* **2024**, *14*, 9744. [CrossRef]
27. Brahmia, A.; Kerboua, A.; Kelaiaia, R.; Latreche, A. Tolerance Synthesis of Delta-like Parallel Robots Using a Nonlinear Optimisation Method. *Appl. Sci.* **2023**, *13*, 10703. [CrossRef]
28. Cretescu, N.; Neagoe, M.; Saulescu, R. Dynamic Analysis of a Delta Parallel Robot with Flexible Links and Joint Clearances. *Appl. Sci.* **2023**, *13*, 6693. [CrossRef]
29. Che, H.; Shi, C.; Wang, Y. Transmission Angle Analysis of a Type of Parallel Manipulators. *Appl. Sci.* **2022**, *12*, 9468. [CrossRef]
30. Malyshev, D.; Rybak, L.; Carbone, G.; Semenenko, T.; Nozdracheva, A. Optimal Design of a Parallel Manipulator for Aliquoting of Biomaterials Considering Workspace and Singularity Zones. *Appl. Sci.* **2022**, *12*, 2070. [CrossRef]
31. Li, I.-H.; Chiang, H.-H.; Lee, L.-W. Development of A Linear Delta Robot with Three Horizontal-Axial Pneumatic Actuators for 3-DOF Trajectory Tracking. *Appl. Sci.* **2020**, *10*, 3526. [CrossRef]
32. Yigit, T.; Celik, H. Dimension Synthesis and Optimized FOPID Control of the Delta Robot with Moth Swarm Algorithm. *Arab. J. Sci. Eng.* **2023**, *48*, 6889–6902. [CrossRef]
33. Young, K.D.; Utkin, V.I.; Ozguner, U.A. Control engineer's guide to sliding mode control. *IEEE Trans. Control Syst. Technol.* **1999**, *7*, 328–342. [CrossRef]
34. Edwards, C.; Spurgeon, S.K. *Sliding Mode Control: Theory and Applications*; Taylor and Francis: New York, NY, USA, 1998.
35. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1991; Volume 199.
36. Aydın, M. Chapter 11 PID Position Control of Star Parallel Robot. In *Current Debates in Engineering Sciences*; Firat University: Elazig, Türkiye, 2022; pp. 193–212.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.