Kinematic examination and replication of six grade of autonomy robotic manipulator with MATLAB

[1]Name of 1st Dr Vikram Patel
[1]Designation of 1st Associate Professor
[1]Name of Department of 1st Faculty of Engineering
[1]Name of organization of 1st Gokul Global University, Sidhpur, Patan, Gujarat – India

Abstract— For the purpose of managing a robotic manipulator with six degrees of freedom (6-DOF), the sliding mode control (SMC) method is proposed in this study. As a consequence, the suggested Sliding mode controller is devised to enhance the outcomes, and the sliding surface is built utilizing proportional and derivative constants to boost smoothness in the end effector moment. This study provides a thorough architecture of the proposed sliding mode controller (SMC) controller.

Keywords: kinematics, dynamics, Robotic manipulator control, PD controller, and Sliding Mode Control, MATLAB/Robotics.

INTRODUCTION

The proliferation of robots in various production processes can be attributed to significant advancements in information technology and mechanics. These advancements have resulted in enhanced productivity, better working conditions, and a rapid progression towards industrial automation. Beyond their applications in environments characterized by extreme temperatures, high intensity, dirt, disturbance, radiation, and contamination, robotic arms have found substantial use in the medical field [3]. This inclusion in the medical sector underscores the versatility and broad applicability of these technological marvels.

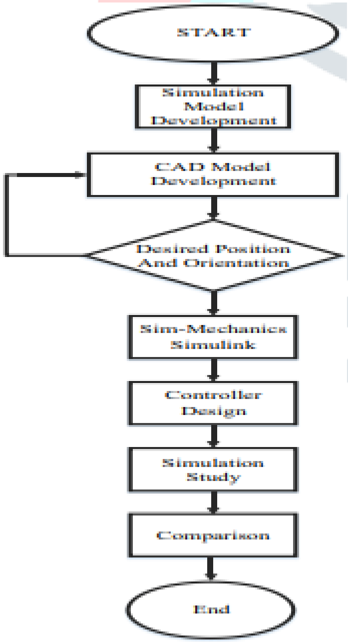
The versatility of robotic arms extends even further; they are now integrated into automated medical procedures, deep-sea research, advancements in nuclear science, and the realm of space exploration. It's evident that the robotic arm isn't just a mere tool; it's a fundamental component of modern automated control technology and has cemented its significance in today's industrial ecosystems. When compared to human efforts, robots streamline processes, thereby slashing the time required for producing results. As a direct consequence, robots exhibit higher productivity rates than humans. Given these advantages, precise control over robotic arms becomes paramount.

The robotic arm design basically composed of links, joints, body and end effector to hold or pick and place the objects and a controller to control the system to follow the given path. This system provides an automated approach to handling, reprogramming, and executing various tasks in a threedimensional space. In the current industrial setting, these robotic arms play a crucial role in manufacturing processes. Typically, they function in an open-loop system, where in they only require a single computation for determining their position, trajectory, and speed. A model of the robot will be developed to address and test issues related to forward kinematics, dynamics, torque, and trajectory planning. This testing process will help in reducing unnecessary risks and consequently, lead to financial savings.

In this study, we delve deeply into the kinematics and dynamics of a robotic arm with six degrees of freedom (6 DOF). Given the increasing array of applications for robotic arms in the automation industry—especially in functions like pick and place—it prompted us to propose both a ProportionalDerivative (PD) and a Sliding Mode Control (SMC) method to regulate the movements of the 6 DOF robotic manipulator. To develop and simulate the aforementioned robotic arm, we employ MATLAB. Initially, our simulations utilized a conventional PD calculated torque controller. However, this approach was found to have suboptimal performance, leading to the end effector taking an extended period to align with the reference trajectory. In response, we designed the recommended sliding mode controller to improve these results. To ensure fluidity in the movement of the end effector, a sliding surface harnessing both proportional and derivative constants has been instituted.

This article provides a thorough architecture of the suggested PD and SMC controller. Since the 6 DOF robot has 6 joints, a trajectory is allocated for each joint to follow in order to produce six distinct trajectory patterns. According to the findings obtained, the suggested controller (SMC) produces reliable and accurate results when the trajectory tracking error acquired using this SMC controller is compared with the PD calculated torque controller.[7] the PID controller of the robotic arm is created in such a way that it can travel from point to point with ease. For the packaging industry, sensors are attached to goods that are stored at various points along a line. The sensors will provide data on the accuracy of the end effector's contact with the item as well as the distance between them. If there is any inaccuracy, the PID controller will make an effort to minimize it and provide accurate control so that it may travel effortlessly between all positions and pick up things. [8] A 3DOF robotic system is described by segmenting it into subsystems depending on the angles in coordinate space where each subsystem is located. Switched system modelling is the name given to this form of modelling. The robotic arm reacts quite quickly to certain complex techniques like sliding mode control. A SMC-based arm can pick up an object from one location and deposit it at another one within a set amount of time.

[9] The authors highlighted the model-based switching functions design and employed an exoskeleton robotic arm. Transients and the system's chattering behavior were also examined prior to it. The torque control rule is significantly simpler than in earlier works, and the control algorithm is created so that the system is asymptotically stable under all circumstances. A hardware robotic arm's design is more expensive and difficult when done for experimental purposes. A lot of money would be lost as well if the hardware model failed. Therefore, experiments in this specific field of inquiry are decreasing. The only way to solve this issue is to create a virtual model. In the beginning of this study, a 6DOF robotic arm with six links and a circular base is designed in MATLAB; the detailed design stages are provided in figure.



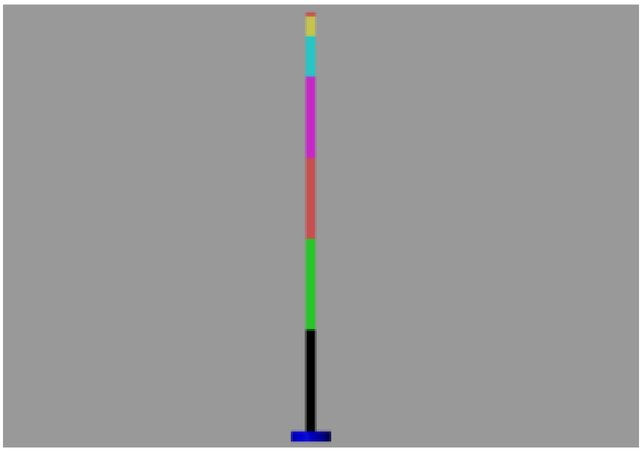
Flowchart for the design procedure of the project.

The goal is to shift the end effecter from its default position inside the work area to a specific location. Trapezoidal trajectory block was used to provide the necessary trajectory route to move the arm from the starting position to the finishing position. In order for the end effector to follow the reference trajectory, the controller now aims to generate sufficient torques at each joint.

The reaction time of traditional PD-computed torque control is poor and it takes longer to follow the course. The study's findings support the assertion that the SMC effectively controls the robotic arm. The goal is to create an SMC controller that follows the trajectory path without making any errors. The sliding surface in this work is built utilizing traditional PD controller gains. A system is deemed stable by a sliding mode controller if its trajectory is on the surface s=0. Therefore, the primary goal of the sliding mode controller's design is to shift the system trajectory to the surface s=0. If s = 0, then s is also zero. Here, s is the change in the sliding surface. A detailed design of the PD and SMC controller is provided in this work.

ROBOTIC ARM DESIGN IN MATLAB

The degree of freedom is a crucial metric for a robot, influenced by its design and directly tied to its adaptability. Typically, a robot with more degrees of freedom exhibits greater flexibility and mimics human movements more closely. However, an increase in degrees of freedom also results in a more intricate structural design. The procedure for design of a model in MATLAB is given below Launch the MATLAB Main window. Open a new model. then add the necessary robotic arm elements and put them together in the proper order. the parts are in the robotic toolbox. the MODEL is then saved. By using a Small Code in New script, construct the rigid body tree for the stored model, then save and run it. The designed robotic arm then has a rigid body tree. This tree is utilized in the Forward Kinematics Analysis and Controlling of Robotic Arms. Figure displays the robot model featuring six degrees of freedom. Joints 2, 3, 4, 5, and 6, which all rotate, lie on the same plane.



2D CAD Model of 6 degrees of freedom robot in MATLAB.

In this simulation, our main goal is to control the robot arm's path and its torque, especially when dealing with environmental challenges. Initially, we're designing a polishing robot with a flexible arm. Next, we analyze its movement to create its operating equations. We use MATLAB for 3D modeling and testing. We also use mechanical setups to study robotic dynamics and specialized tools to assist in various robot designs. These tools help with movement modeling and offer solutions in both 3D space and joint positions. Figure displays a CAD model made with MATLAB.

Kinematics Modeling

For accurate control of industrial robots in polishing operations, it's crucial to set up the robot arm's kinematic model. This model describes how each segment of the arm relates to the others. We use the D-H (Denavit-Hartenberg) parameters to explain how each joint of the robot connects. This D-H approach uses four specific measures to describe the relationship between one segment (i) and the one before it (i-1). Our study also uses transformation matrices based on the D-H parameters and foundational expressions about the robot's connections. As shown in Figure, we outline how two successive robot segments relate spatially, marking coordinates for each. After defining the coordinate system for each link, we use the D-H parameters to display how the coordinate systems of two adjacent joints relate.

Based on the idea that the robot system possesses six degrees of freedom, and by using the transformation matrix Ti (where i ranges from 1 to 6) based on the Denavit-Hartenberg (D-H) parameters, we can determine how each solid component connects to the preceding one through a joint. From this matrix, we can understand the position and orientation of the rigid body, adjust a vector's reference frame, and shift the vector. This understanding comes from considering the values tied to the robot's segments and the measurements linked to each rod's length.

The Denavit-Hartenberg (DH) parameters specific to this particular robotic arm can be found itemized in the provided.

Link ai (mm) αi (degree) di (mm) θi (degree) 1 0 0 0 Θ1 2 0 -90 0 Θ2 3 a2 0 d3 Θ3 4 a3 -90 d4 Θ4 5 0 90 d5 Θ5 6 0 -90 0 Θ6

Additionally, the transformation matrix that defines the relationship between two consecutive linkages can be derived by employing the Denavit-Hartenberg (DH) frame, as detailed below:

$$T_i = R_{Z,\theta_i} T_{Z,d_i} T_{x,a_i} R_{x,\alpha_i}$$

$$T_1 = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

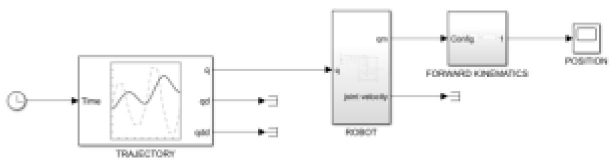
$$T_5^4 = \begin{bmatrix} \cos_5 & -\sin_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin_5 & \cos_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^5 = \begin{bmatrix} \cos_6 & -\sin_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin_6 & -\cos_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^6 * T_6^0$$

$$T_6^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & O_{6x} \\ r_{21} & r_{22} & r_{23} & O_{6y} \\ r_{31} & r_{32} & r_{33} & O_{6z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The analysis of robot kinematics establishes the relationship between the robot's links and joints in relation to its position and orientation. This kinematic examination is split into forward and inverse kinematics. In this article, we've executed forward kinematics for the robotic arm we designed. Figure displays the block diagram for the forward kinematics.



forward kinematics block of the designed robot

Dynamic Model

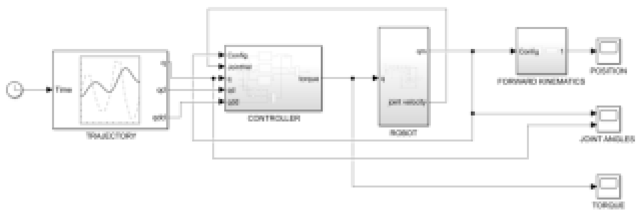
In the given context, M(q) represents the positive inertia matrix. h q q (,) isThe central force is coupled with the Coriolis force vector. Additionally, G(q) denotes the gravitational force vector, where q signifies the position of the joint, and τ indicates the torque at the joint. Within the scope of this research paper, a design featuring six degrees of freedom has been crafted. The individual masses for the six joints are provided as are their corresponding lengths

are given as 1 2 6 m m m , ,... and 1 2 6 l l l , ,... respectively.

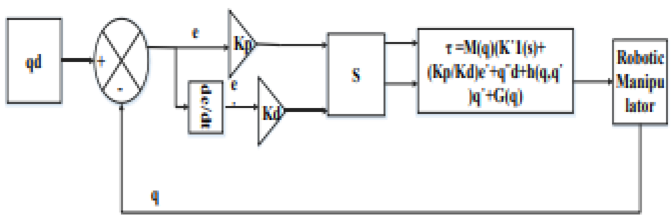
Within the context of the Cartesian coordinate system, when the terminus of the arm interacts with an external entity, both the arm and this external factor exert reciprocal forces Fe upon one another. To ensure the equilibrium and stability of the arm during this interaction, it's imperative to apply a specific driving torque . T T J Fe = to each of its joints. Here, J represents the Jacobian matrix. This framework allows the Cartesian force, which acts upon the arm's extremity, to be translated into a corresponding joint torque. When the arm engages with an external medium, the kinetic equation governing its actuator can be articulated as

$$T = M(q)\ddot{q} + h(q,\dot{q}) + G(q) - J^T F_e$$

In the realm of forward dynamics, we engage in a process where we compute a series of essential parameters within a robotic system. These parameters include joint accelerations, velocities, and positions. To perform these calculations, we use the output torques as our input data. In essence, forward dynamics provides us with a way to understand how the robot's components move and change in response to the forces and torques applied. In contrast, when we delve into the realm of inverse dynamics, our objective shifts. Here, the focus is on determining the precise torques or forces that must be applied at each joint of the robot to achieve a desired motion or trajectory. This involves a complex calculation process that takes into account various factors such as the distribution of mass within the robot and its geometric properties. In essence, inverse dynamics helps us decipher the forces needed to execute a specific motion while considering the intricacies of the robot's physical structure.

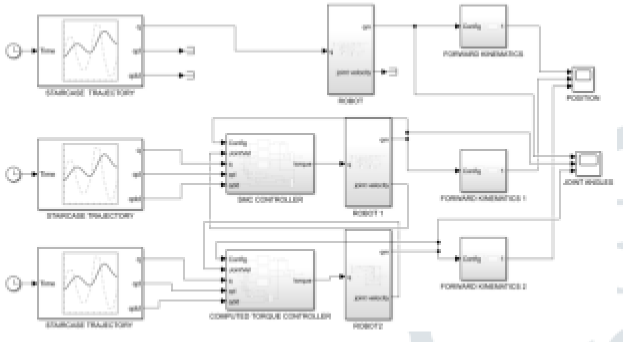


Simulation circuit of dynamic model



SMC Controller

complete simulation system for the designed robotic arm for basic robot and pd controlled robot and SMC controlled robot with staircase signal is as shown in figure.



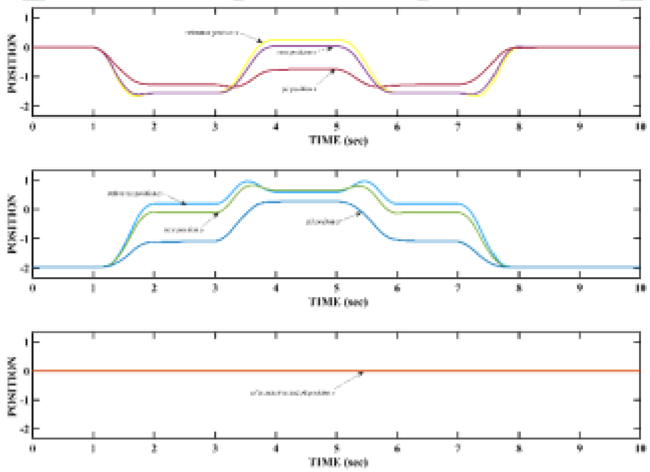
simulation circuit of designed robot with SMC and PD controllers

RESULTS

The outcomes from the simulations of the basic robot, the robot with PD control, and the robot with SMC when subjected to a staircase signal are depicted in the provided figures.

Among the visual representations, the initial three waveforms illustrate the positions of the end effector during the execution of forward kinematics.

Following these, the next six waveforms detail the angular positions of each individual link within the designed robotic arm when its dynamic operations are carried out.



x, y, z. positions of the designed robotic arm

Figure. provides a visual representation of the variations in the position of the specifically designed robotic arm in relation to the set trajectory, considering the basic, PD, and SMC controlled robotic arm configurations. From the illustration, it's evident that when comparing the performance of the robotic arm operated using the PD-computed torque controller to that of the one operated using the SMC controller, the latter exhibits superior accuracy. Specifically, the robotic arm controlled by the SMC controller adheres more closely to the intended position, manifesting a reduced margin of error. Shifting our focus to the joint angles, it becomes evident how precisely the robotic arm, under the guidance of the SMC controller, aligns with the reference trajectory. This alignment is virtually error-free, especially when compared against the performance of the PD controller. From this observation, we can infer that the SMC controller outperforms the PD controller in terms of delivering superior accuracy and outcomes, ensuring that the robotic arm closely follows the designated trajectory in figure.

CONCLUSION

To sum up, every controller possesses both strengths and shortcomings. The PD calculated torque controller provides precise trajectory control with appropriate dynamic compensation, while the sliding mode controller excels at robust tracking. The controller to choose relies on the particular needs and features of the robotic manipulator system. The MATLAB-based robotic arm that was created is functional, and for experimental purposes, this kind of MATLAB-based robotic arm is affordable and straightforward.

It is preferable to utilize these kinds of CAD models for educational and research purposes. since they function similarly to standard hardware models and allow for easy, lowcost error correction during testing. then we may create the hardware model utilizing 2D printing tools. When compared to the PD Computed Torque Controller, the suggested controller (SMC) performs better. In future Fuzzy controllers and sliding mode controllers will be integrated to create a hybrid fuzzy sliding mode controller in order to enhance outcomes in subsequent efforts. This controller will advance to the position with less time and fewer mistakes.

REFERENCES

[1]. Juan Liu, Quang Luo “modelling and simulation of robotic arm for industrial applications” 2017,11th international conference human machine systems and cybernetics IEEE. [2]. Dinh Tho long, to van binth, roan van hoa, le van anh, Nguyen van taon “robotic arm simulation by using MATLAB and robotics toolbox for industry application” international journal of electronics and communication engineering 2016. [3] Iqbal, Jamshed, R. Ul Islam, and Hamza Khan, "Modelling and analysis of a 6 DOF robotic arm manipulator", Canadian Journal on Electrical and Electronics Engineering 3, no. 6, pp. 300-306, 2012. [4] Topaz, Metin, and Serdar Kucuk, "Dynamics simulation toolbox for industrial robot manipulators," Computer Applications in Engineering Education 18, no. 2, pp. 319- 330, 2010. [5] Mohammed Abu Qassem, I.M. Abuhadrous, H. Elaydi, "Modelling and Simulation of 5 DOF educational robot arm", In Conference: Advanced Computer Control (ICACC), 2010 2nd International Conference on, Volume. 5 April 2010. [6] Wang, Zhi-Xing, Wen-Xin Fan, Bao-Cheng Zhang, and Yuan-Yuan Shi, "Kinematical analysis and simulation of an industrial robot based on Matlab," Jidian Gongcheng/Mechanical & Electrical Engineering Magazine 29, no. 1, pp. 33-37, 2012 [7] Safiuddin Che Suhaimin, Nur Liyana Azmi, Md Mozasser Rahman, and Hazlina Md Yusof, “Analysis of Point-to-Point Robotic Arm Control using PID controller” 2017 7th International Conference on Mechatronics Engineering (ICOM), Jan. 2017. [8] Maria Philip and Lal Priya P S, “Pick and Place Operation of a Robotic Arm Using Multirate Event Triggered Sliding Mode Control” IEEE Second International Conference on Control, Measurement and Instrumentation (CMI), India, 2014. [9]. Charles Fallaha , Maarouf Saad, Jawhar Ghommam , and Yassine Kal ,“Sliding Mode Control With Model-Based Switching Functions Applied on a 7-DOF Exoskeleton Arm”, IEEE/ASME Trans. Mechatronics, vol.26, no.1, pp.539–550, Feb. 2015. [10]. Bdereddin Abdul Samad , Mahmoud Mohamed and Fatih Anayi, “Enhanced the Control Strategy of a Triple Link Robotic System (Robogymnast)”,vol.11, pp. 31997 – 32005, ISSN: 21693536,Mar.2016. [11]. Madiha Farman, Muneera Al-Shaibah, Zoha Aoraiath, Firas Jarrar, “Design of a Three Degrees of Freedom Robotic Arm”, International Journal of Computer Applications, vol.179, no.37, Apr. 2017 . [12] Kang, Hee-Jun, and Young-Shick Ro, "Robot manipulator modelling in Matlab-SimMechanics with PD control and online gravity compensation," In International Forum on Strategic Technology 2010, pp. 446-449. IEEE, 2017.