SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549 / https://doi.org/10.14445/23488549/IJECE-V10I1P101
Volume 10 Issue 1, 1-5, January 2023

Original Article

Trajectory Tracking based on Sliding Mode Controller

Fakhur-un-nisa Alias Fizza Syed[1], Shakeel Ahmed Shaikh[2], Saifullah Samo[3], Qamar un nisaKamal[4]

[1,2,3,4] *Mehran University of Engineering and Technology Jamshoro, Pakistan.*

Received: 02 December 2022 Revised: 05 January 2023 Accepted: 18 January 2023 Published: 31 January 2023

Abstract *- This paper presents trajectory tracking based on a sliding mode controller for the Puma robot manipulator. Puma is most commonly used in industries; it has great flexibility compared to other manipulators like SCARA, which decreases its precision. In order to increase the precision Sliding mode controller based on the Lyapunov stability approach is used in this work. The first motion control block for the sliding mode Controller is designed and link it with the Puma Robot manipulator in MATLAB Simulink. The Experiment Results for the effectiveness of this method are verified.*

Keywords *- Trajectory Tracking, Sliding mode, PUMA560 Robot, Trajectory generation.*

1. Introduction

Robot Manipulators have been successfully developed for a long time to achieve high precision control performance. Nevertheless, due to their higher accuracy, more accurate and effective control of robot manipulators is desirable [1-4]; in many engineering disciplines, control is still essential. There is an available state of the art in control technology, but this field still has certain gaps and unresolved issues [5]; there is still a need for robot manipulator control to make implementation more feasible and to improve control performance [6-9]; it can be solved by Designing a stable controller that provides maximum precision, robustness, fast response rate, and properties through which manipulator uncertain dynamics and disturbances can be handled.

Within the state of art, the ordinary SMC and its advancements have been explored for mechanical technology and nonlinear frameworks control for a few decades. To realize acceptable systems execution [10-12]

2. Literature Review

The PUMA robot is a six-degree-of-freedom industrial manipulator developed at MIT in the mid-1970s. It's the oldest industrial robot used in many research areas [8]. Without interfering with the primary purpose, manipulator redundancy can be exploited to accomplish secondary objectives, including avoiding join limits, avoiding obstacles, and minimizing joint torque [15-16].

Without interfering with the primary purpose, manipulators' redundancy can be exploited to accomplish secondary objectives, including avoiding joint limits [9], avoiding obstacles, and minimizing joint torque [19-20]. The end-effector's intended velocity and the IK's preset constraints (inverse kinematics) are used to construct an appropriate sequence of joint speed to drive the redundant manipulator. However, the Jacobian matrix's underdetermined ability would lead to endless IK solutions and likely singularity.

3. Robot Kinematics

Creating the Kinematic Equation for the robot is simple once the D-H coordinate system has been constructed for each link and the relevant link parameters have been discovered. The i^{th}coordinate frame and the $(i-1)^{th}$ the coordinate frame will be connected by a homogenous transformation matrix [20-22]

By applying four sequential transformations, any point in the i^{th} the coordinate system can be described in the $(i-1)^{th}$coordinate system: rotating an angle of the system can be described ɵ1 about the zi-1 axis; Translating di distance along $zi-1$ axis; rotating ai angle along axis xi.

In mathematical term, this can be stated as follow:

$$Ai^{i-1} = Rot(Zi-1,\theta).Trans(Zi-1.di).Trans(zi,ai).Rot(zi,ai)$$

So general form of link transformation become:

$$Ai^{i-1} = \begin{bmatrix} cos\theta & -sin\theta & 0 & \alpha i-1 \\ sin\theta i.cos\theta i-1 & cosi.cos\alpha i-1 & -sin\alpha i-1 & -di.sin\alpha i-1 \\ sin\theta i, sin\alpha i-1 & cos\theta i.sin\alpha i-1 & cos\alpha i-1 & di.cos\alpha i-1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

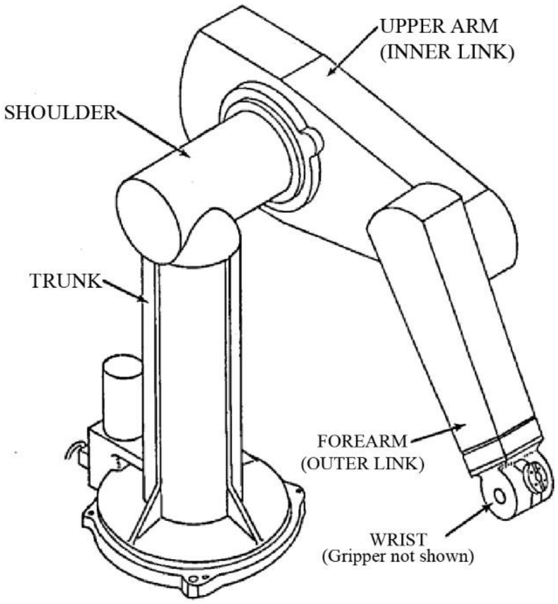

Fig. 1 D-H coordinate frame of PUMA robot [11]

Table 1. D-H parameters

I	a_{i-1}	χ_{i-i}	d_i	θ_i
1	0	0	0	q1
2	0	-90	0.2435 m	q2
3	0.4318	0	-0.0834 m	q3
4	-0.0203	90	0.4331 m	q4
5	0	-90	0	q5
6	0	90	0	q6

4. Inverse Kinematics

Unlike Forward kinematics, Inverse kinematics solves for the joint angle, given the position and the introduction of the end-effector. Inverse kinematics problems can be illuminated utilizing arithmetical, iterative or geometric approaches [22-26]. The logarithmic approach does not donate a clear indication of how to choose the correct solution from the few conceivable arm arrangements. On the other hand, the iterative strategy necessitates massive computations and may not yield the correct solution. The Geometric technique is typically complicated and necessitates a strong understanding of geometry.

We will have twelve equations with only three unknowns from the homogenous matrix that describes the position of the end-effector coordinate frame with respect to the base frame, which is more than enough to solve for the joint parameters.

We initially solved the frame 5 position (P5x, P5y, P5z) and utilized these coordinates values to calculate the first three angles of joints.

Frame 5 Position can be determined as:

$$P5 = \begin{bmatrix} Px - d_6 ax \\ py - d_6 ay \\ pz - d_6 az \end{bmatrix}$$

The joint angle θ_1 is derived as:

$$C2 = P_5 x - d_1$$

$$C3 = \sqrt{{C_1}^2 + {C_2}^2}$$

$$C4 = \sqrt{{r_3}^2 + {d_4}^2}$$

$$\theta_1 = \alpha tan_2(P_5 y, P_5 x) - \alpha tan_2(D_1, \sqrt{{D_1}^2})$$

$$D_1 = \frac{d_2}{\sqrt{P_5 y^2 + P_5 y^2}}$$

$$\theta_2 = \alpha tan_2(C_2, C_1) - \alpha tan_2(\sqrt{1 - D^2{}_2}, D_2)$$

We used the angles obtained at the first three joints to solve for the forward kinematics from frame 0 to frame 3. By taking the dot product of ${T_6}^0$ and the inverse of ${T_6}^3$, we solve the homogenous transformation matrix ${T_6}^3$ from frames 3 to 6. In general, the Dh model for a manipulator with 6 degrees of freedom is defined as

$${T_6}^0 = {T_1}^0 . {T_2}^1 . {T_3}^2 , {T_4}^3 . {T_5}^4 . {T_6}^5$$

We can simplify the model as:

$${T_6}^3 = {T_3}^0 . [{T_6}^3]^{-1}$$

$$\theta_3 = \alpha tan2\left(\sqrt{1 - {D_3}^2}, D^3\right) - 90$$

$$\theta_4 = \alpha tan2({T_6}^3(2,3), {T_6}^3(1,3))$$

$$\theta_5 = \alpha tan2\left(\sqrt{({T_6}^3(2,3), {T_6}^3(1,3))}\right)$$

$$\theta_6 = \alpha tan2({T_6}^3(3,2), {T_6}^3(3,1))$$

5. General Dynamic Model for Robot Manipulator

5.1. Dynamic Model for Robot Manipulator

The L-E equation is applied to each of the device's links to determine the torque vector delivered to the manipulator's joints. The dynamic model of the manipulator can then be represented generally as:

$$\mathrm{M(q)\acute{q}\acute{q} + V(q, \acute{q})\acute{q} + G(q) = r}$$

$\mathrm{M(q)}$ is the inertia matrix of the robot Manipulator.

$\mathrm{V(q, \acute{q})}$ is the Coriolis and Centrifugal term.

$\mathrm{G(q)}$ is the Gravity term.

Γ is the torque.

All matrices are a function of simply the manipulator position when the velocity-dependent term is expressed in a different way; in this instance, the dynamic equation is known as the configuration state space equation and has the following form:

$$M(q)\acute{\acute{q}} + B[\acute{q}\acute{q}] + C(q)[\acute{q}^2] = r$$

$B[q]$ is the Coriolis toque.

$C(q)$ is the Centrifugal torque matrix.

$[\acute{q}\acute{q}]$ Joint velocity vector product

$[\acute{q}^2]$ Vector given as $[\acute{q}1^2, \acute{q}_2{}^2, \acute{q}_3{}^2, \acute{q}_4{}^2 \dots \acute{q}n^2]$

So, the dynamic system equation of PUMA560 in the state space is formed as below:

$$\acute{\acute{q}} = M^{-1}F - C(\acute{q}\acute{q}) - G(q)$$

And derivation of the state is:

$$\acute{x}1 = x2 \quad \acute{x}2 = \acute{\acute{q}}1,$$

$$\acute{x}3 = x4 \quad \acute{x}4 = \acute{\acute{q}}2$$

$$x'5 = x6 \quad \acute{x}6 = \acute{\acute{q}}3$$

$$x7 = q4 \quad x'7 = x8\,,$$

$$x8 = \acute{x}8 \quad \acute{x}8 = \acute{\acute{q}}4$$

$$x'9 = x10 \quad \acute{x}10 = \acute{\acute{q}}5$$

$$x'11 = x12 \quad \acute{x}12 = \acute{\acute{q}}6$$

5.2. Sliding Mode Controller

Sliding mode, the known characteristic of control, is its insensitivity to model uncertainty and remarkable resilience to various disruptions. This Control legislation, however, suffers from a very serious drawback known as chattering. Recent SMC research aims to lessen chattering while studies concentrate on learning and developing optimal SMC algorithms.

Several strategies were explored input by varying the boundary layer's width with fuzzy logic to overcome the sliding mode controller chattering issue. This approach guarantees nearly flawless control and does away with chattering; another approach provides a reliable control rule for reducing chatter based on generalized sliding mode control that activates the control input's derivative rather than the input itself.

We are creating a sliding mode control Legislation; the following two procedures must be taken:

- Creating a sliding surface such that when the system is stuck on it, it exhibits the desired behavior.
- Creating a switching control rule that will move the plant state to the manifold and keep it there for the rest of the time.

$$U = U\,eq + U\,dis$$

The sliding surface is defined as:

$$S(x,t) = \left(\frac{d}{dt} + c\right)^{n-1} . e(t)$$

Error can be defined as:

$$e(t) = x(t) - xd(t)$$

$x(t)$ is the actual state.

$xd(t)$ is the desired state.

Reachability condition can be defined as:

$$\acute{V}(s) = s.\acute{s}$$

$$\begin{aligned}\acute{s}(x,t) &= \acute{\acute{e}}(t) + c.\acute{e}(t) = \acute{\acute{x}} - \acute{\acute{x}}d + c.\acute{e} \\ &= f(x) + g(x).u(t) - \acute{\acute{x}}d + c.e(t)\end{aligned}$$

The sliding manifold is defined by:

$$s.\acute{s} = s.((\hat{f}(x) + \hat{g}(x)u(t) - \acute{\acute{x}}d + c.\acute{e}(t))$$

This is the Equivalent of the continuous control law.

The discontinuous control law is:

$$udis(t) = -ksign(s)$$

6. MATLAB Simulation

A visual simulation was performed to validate the effectiveness of the safe trajectory simulation control. In the algorithm, we employed two blocks in the sliding mode controller, one for continuous and one for discontinuous control law, and a dynamic model to simulate six join PUMA robots in Simulink. The robot is represented as a block that accepts control torque as input and output joint displacements.

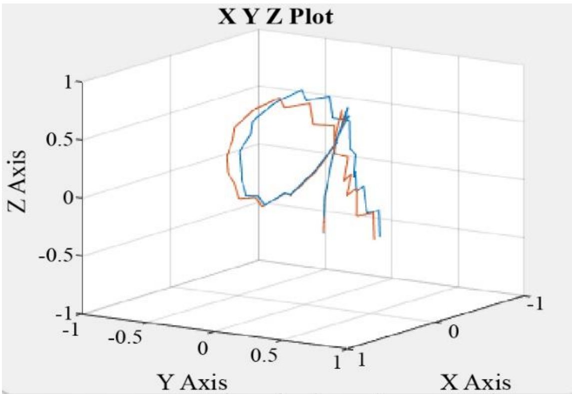

Fig. 2 End effector trajectory red line shows the Desired Path, blue shows the Actual Path

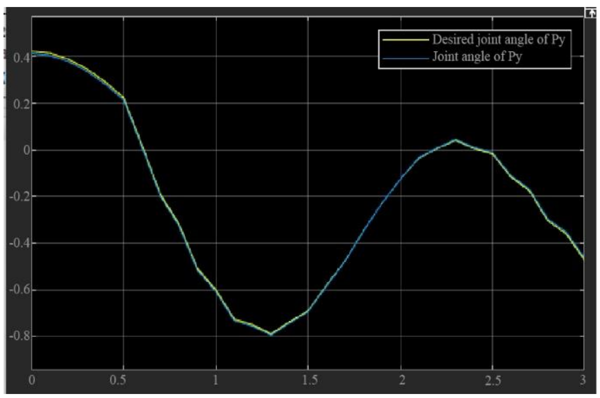

Fig. 3 Actual Verses desired joint angle trajectory of the End effector

The calculation of error makes it easier to compare the Flexibility and Efficiency of control regulations. The error can be defined as the mathematical distance between a desired point in space and its real counterpart at a given moment and its degree of divergence of the Actual path drawn by the robot from the desired path chosen by the user.

Table 2. Error in every joint

No#	Error in every joint	Overall, Error
01	0.3634	0.1768
02	0.2636	
03	0.693	
04	0.1153	
05	0.4046	
06	0.113	

7. Conclusion

Based on the trajectory tracking control approach for PUMA560, it is proposed to utilize a sliding mode control hypothesis. The framework states can be modified through a sliding mode control that converges to zero in a limited time. The sliding mode control includes a predominant concurrent characteristic. The formulated sliding mode control guarantees the stability of the system. Observing the simulation results above proves that the chattering is reduced, but the speed and direction of the system are kept up.

Acknowledgment

I would like to acknowledge my gratefulness and render my warmest thanks to my supervisor Dr. Saifullah Samo and co-supervisor, Dr. Shakeel Ahmed Shaikh who made this work possible. Their friendly guidance and expert advice have been valuable throughout all stages of work.

References

[1] Joseph Davidson, “Robotic Manipulation for Specialty Crop Harvesting: A Review of Manipulator and End-Effector Technologies,” *Global Journal of Agricultural and Allied Sciences*, vol. 2, no. 1, pp. 25–41, 2020. *Crossref*, https://doi.org/10.35251/gjaas.2020.004

[2] Bin Wei, “A Tutorial on Robust Control, Adaptive Control and Robust Adaptive Control—Application to Robotic Manipulators,” *Inventions*, vol. 4, no. 3, pp. 1-13, 2019. *Crossref*, https://doi.org/10.3390/inventions4030049

[3] Ethelbert Ezemobi et al., “Battery State of Health Estimation with Improved Generalization Using Parallel Layer Extreme Learning Machine,” *Energies*, vol. 14, no. 8, pp. 1-15, 2021. *Crossref*, https://doi.org/10.3390/en14082243

[4] Long Chen et al., “Extreme-Learning-Machine-Based Robust Integral Terminal Sliding Mode Control of Bicycle Robot,” *Control Engineering Practice*, vol. 121, 2022. *Crossref*, https://doi.org/10.1016/j.conengprac.2022.105064

[5] Yuxiang Wu et al., “Adaptive Neural Network Control of Uncertain Robotic Manipulators with External Disturbance and Time-Varying Output Constraints,” *Neurocomputing*, vol. 323, pp. 108–116, 2019. *Crossref*, https://doi.org/10.1016/j.neucom.2018.09.072S.

[6] Chouraqui, and Habiba Benzater, “A Multiobjective Genetic Algorithm Applied to Control Optimization,” *Science International*, vol. 3, no. 1, pp. 7-17, 2015.

[7] Qingyun Zhang et al., “Adaptive Sliding Mode Neural Network Control and Flexible Vibration Suppression of a Flexible Spatial Parallel Robot,” *Electronics*, vol. 10, no. 2, pp. 1-22, 2021. *Crossref*, https://doi.org/10.3390/electronics10020212

[8] Anh Tuan Vo, Mr. Truong Thanh Nguyen, and Dr. Hee-Jun Kang, “A Novel Prescribed-Performance-Tracking Control System with Finite-Time Convergence Stability for Uncertain Robotic Manipulators,” *Sensors*, vol. 22, no. 7, pp. 1-18, 2022. *Crossref*, https://doi.org/10.3390/s22072615

[9] Peiyu Wang et al., “Prescribed Performance Control with Sliding-Mode Dynamic Surface for a Glue Pump Motor Based on Extended State Observers,” *Actuators*, vol. 10, no. 11, pp. 1-28, 2021. *Crossref*, https://doi.org/10.3390/act10110282

[10] Qiang Chen, "Adaptive Robust Finite-Time Neural Control of Uncertain PMSM Servo System with Nonlinear Dead Zone," *Neural Computing and Applications*, vol. 28, pp. 3725–3736, 2017. *Crossref*, https://doi.org/10.1007/s00521-016-2260-5

[11] Ngoc Phi Nguyen et al., "Finite-Time Attitude Fault Tolerant Control of Quadcopter System via Neural Networks," *Mathematics*, vol. 8, no. 9, 2020. *Crossref*, https://doi.org/10.3390/math8091541

[12] Saeed Benjamin Niku, *Introduction to Robotics: Analysis, Control, Applications*, John Wiley & Sons: Hoboken, NJ, USA, p. 480, 2020

[13] Haoyan Zhang et al., "Adaptive Fuzzy Hierarchical Sliding Mode Control of Uncertain Under-Actuated Switched Nonlinear Systems With Actuator Faults," *International Journal of Systems Science*, vol. 52, no. 8, pp. 1499–1514, 2021. *Crossref*, https://doi.org/10.1080/00207721.2020.1831645

[14] Zhi-Min Li et al., "Quantized Static Output Feedback Fuzzy Tracking Control for Discrete-Time Nonlinear Networked Systems with Asynchronous Event-Triggered Constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3820–3831, 2021. *Crossref*, https://doi.org/10.1109/TSMC.2019.2931530

[15] Li Ma et al., "Small-Gain Technique-Based Adaptive Neural Output-Feedback Fault-Tolerant Control of Switched Nonlinear Systems with Unmodeled Dynamics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 7051–7062, 2020. *Crossref*, https://doi.org/10.1109/TSMC.2020.2964822

[16] Yulin Li et al., "Command Filter-Based Adaptive Neural Finite-Time Control for Stochastic Nonlinear Systems with Time-Varying Full-State Constraints and Asymmetric Input Saturation," *International Journal of Systems Science*, vol. 53, no. 1, pp. 199–221, 2022. *Crossref*, https://doi.org/10.1080/00207721.2021.1943562

[17] Somdavee Bhosinak, Dechrit Maneetham, and Tenzin Rabgyal, "Hybrid Fuzzy PID Controller for Intelligent Tractor Steering Control," *International Journal of Engineering Trends and Technology*, vol. 70, no. 12, pp. 359-369, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I12P235

[18] Wen-Kung Tseng, and Hou-Yu Chen, "The Study of Tracking Control for Autonomous Vehicle," *SSRG International Journal of Mechanical Engineering*, vol. 7, no. 11, pp. 57-62, 2020. *Crossref*, https://doi.org/10.14445/23488360/IJME-V7I11P108

[19] Yukai Zhu, Jianzhong Qiao, and Lei Guo, "Adaptive Sliding Mode Disturbance Observer-Based Composite Control with Prescribed Performance of Space Manipulators for Target Capturing," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 1973–1983, 2019. *Crossref*, https://doi.org/10.1109/TIE.2018.2838065

[20] Yang Liu, Xiaoping Liu, and Yuanwei Jing, "Adaptive Neural Networks Finite-Time Tracking Control for Non-Strict Feedback Systems via Prescribed Performance," *Information Sciences*, vol. 468, pp. 29–46, 2018. *Crossref*, https://doi.org/10.1016/j.ins.2018.08.029

[21] Yuanwei Jing, Yang Liu, and Shaowei Zhou, "Prescribed Performance Finite-Time Tracking Control for Uncertain Nonlinear Systems," *Journal of Systems Science and Complexity*, vol. 32, pp. 803–817, 2019. *Crossref*, https://doi.org/10.1007/s11424-018-7287-5

[22] Zhi-Gang Zhoua, "Prescribed Performance Fixed-Time Tracking Control for a Class of Second-Order Nonlinear Systems with Disturbances and Actuator Saturation," *International Journal of Control*, vol. 94, no. 1, 223–234, 2020. *Crossref*, https://doi.org/10.1080/00207179.2019.1590644

[23] Runqi Chai et al., "Six-DOF Spacecraft Optimal Trajectory Planning and Real-Time Attitude Control: A Deep Neural Network-Based Approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 5005–5013, 2020. *Crossref*, https://doi.org/10.1109/TNNLS.2019.2955400

[24] Runqi Chai et al., "Dual-Loop Tube-Based Robust Model Predictive Attitude Tracking Control for Spacecraft with System Constraints and Additive Disturbances," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4022 – 4033, 2022. *Crossref*, https://doi.org/10.1109/TIE.2021.3076729

[25] Xudong Cao, Jianjun Wang, and Wei Xiang, "Composite Adaptive Fuzzy Prescribed Performance Control of Nonlinear Systems," *Mathematical Problems in Engineering*, vol. 2020, 2020. *Crossref*, https://doi.org/10.1155/2020/2948130

[26] J. Tao, and T. Zhang, "Novel Finite-Time Adaptive Neural Control of Flexible Spacecraft with Actuator Constraints and Prescribed Attitude Tracking Performance," *Acta Astronautica*, vol. 179, pp. 646-658, 2021. *Crossref*, https://doi.org/10.1016/j.actaastro.2020.10.010

[27] Anh Tuan Vo, and Hee-Jun Kang, "An Adaptive Terminal Sliding Mode Control for Robot Manipulators with Non-Singular Terminal Sliding Surface Variables," *IEEE Access*, vol. 7, pp. 8701–8712, 2018. *Crossref*, https://doi.org/10.1109/ACCESS.2018.2886222

[28] Thanh Nguyen Truong, Anh Tuan Vo, and Hee-Jun Kang, "A Backstepping Global Fast Terminal Sliding Mode Control for Trajectory Tracking Control of Industrial Robotic Manipulators," *IEEE Access*, vol. 9, pp. 31921–31931, 2021. *Crossref*, https://doi.org/10.1109/ACCESS.2021.3060115