A Study on 7-DOF Manipulator Control by Using MATLAB Robotics Toolbox

Karam Dad Kallu
Pusan National University

Min Cheol Lee
Pusan National University

A Study on 7-DOF Manipulator Control by Using MATLAB Robotics Toolbox

Bao YuLong, Zhou YuDong, Wang Jie, MH Khan, Karam Dad , Abbasi Saad and Min Cheol Lee*
Department of Mechanical Engineering, Pusan National University, Busan, South Korea
(Tel : +82-10-510-3081; +82-10-3861-2688; E-mail: sshowbo07@pusan.ac.kr; mclee@pusan.ac.kr)

***Abstract*—This paper proposes an efficient method to implement control simulation for a 7-DOF manipulator by using Robotics Toolbox. Due to non-linear and coupling characteristics possessed by the dynamics of a manipulator, conventional simulation method that simplifies robotic dynamics as a linear SISO system can result in inaccurate simulation and cannot validate the performance of a devised controller. Thus, in this study, Robotics Toolbox programmed in MATLAB is utilized for obtaining the actual dynamic model of a 7-DOF manipulator, sliding mode control(smc) is used to be a designated control method for the robotics arm mentioned above. Lastly, numerical simulation results are given to verify validity of the method.**

I. INTRODUCTION

For the control simulation of a robotic arm, most studies focus on the implementation of a single axis. In other words, the control method is simulated based on the dynamics of one axis, ignoring the coupling effect from other axes and the time-variant of parameters.To improve performance and correctness of control simulation of the multi-axis system, this paper uses MATLAB robotics toolbox to simulate control of 7-DOF manipulator based on sliding model control. The Robotics Toolbox is a software package that allows a MATLAB user to readily create and manipulate datatypes fundamental to robotics such as homogeneous transformations, quaternions and trajectories. Functions provided, for arbitrary serial-link manipulators, include forward and inverse kinematics, Jacobians, and forward and inverse dynamics [1].

II. METHOD

The toolbox provides a serial-link manipulator object based on the D-H algorithm, so it is quite handy to implement kinematics [2] of the manipulator. Firstly, after acquiring the D-H parameters, the Link function provided by the toolbox can be used to build a link object that contains all information of a single axis, then the Serial-Link function combines all link objects to create a 7-DOF robotics arm object which is the basis for the subsequent simulation. In this case, the D-H parameters are given in table 1.

Table 1. 7-Link D-H parameter table

Joint	θ (°)	d (m)	a (m)	α (°)
1	θ_1^*	0.124	0	90
2	θ_2^*	0	0	-90
3	θ_3^*	0.155	0	90
4	θ_4^*	0	0	-90
5	θ_5^*	0.160	0	90
6	θ_6^*	0	0	-90
7	θ_7^*	0.150	0	0

*:variable joint angle

In the toolbox, several functions are given to solving inverse kinematics [3] problem of arbitrary serial-link manipulators. Briefly, the algorithms it used can be divided into two classes, the Jacobian iteration method and the optimization toolbox method using the libraries that provide solvers to nonlinear optimization problems and developed by MATLAB. In this study, the Jacobian method [4] ,[5] is utilized for getting the inverse kinematics result of trajectory points. Thereby, the correspondence between the joint space and the operation space is obtained,

$$atj=\begin{bmatrix} 0.9999 & -0.0122 & -0.0010 & -0.0002598 \\ 0.0123 & 0.9965 & 0.0832 & 0.02118 \\ 0 & -0.0833 & 0.9965 & 0.5869 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj2=\begin{bmatrix} 0.8914 & 0.4531 & -0.0072 & -0.09885 \\ 0.4532 & -0.8913 & 0.0142 & 0.1944 \\ 0 & -0.0159 & -0.9999 & 0.09896 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$atj3=\begin{bmatrix} 0.6540 & 0.7331 & -0.1863 & -0.2111 \\ 0.7565 & -0.6339 & 0.1611 & 0.1825 \\ 0 & -0.2463 & -0.9692 & 0.09903 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj4=\begin{bmatrix} -0.7154 & 0.6905 & -0.1068 & -0.1953 \\ 0.6987 & 0.7070 & -0.1093 & -0.1999 \\ 0 & -0.1528 & -0.9883 & 0.09988 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$atj5=\begin{bmatrix} -0.8576 & 0.5142 & -0.0049 & -0.1113 \\ 0.5142 & 0.8576 & -0.0081 & -0.1856 \\ 0 & -0.0095 & -1.0000 & 0.09901 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj6=\begin{bmatrix} 0.9010 & 0.4318 & -0.0402 & -0.09894 \\ 0.4337 & -0.8972 & 0.0835 & 0.2055 \\ 0 & -0.0927 & -0.9957 & 0.1073 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, atj is position and orientation of end-effector in operation space.

The next step is to decide some trajectory points or middle points where the robot should definitely arrive. Because the toolbox has the advantage of easily visualizing the state of the robot, some certain points including orientation and position information of the end-effector can be got directly from the rendered graph. The chosen points are shown as homogeneous transformation matrices in atj and the inverse kinematics results are in table 2.

Table 2. 7 Joint space table

Joint	θ1	θ2	θ3	θ4	θ5	θ6	θ7
Position1	0	0	0	0	0	0	0
Position2	0.46088	0.37699	0	1.31	0	1.4451	0
Position3	0.81681	0.56549	0	1.0681	0	1.2566	0
Position4	2.36	0.69115	0	0.848	0	1.4451	0
Position5	2.66	0.37699	0	1.31	0	1.4451	0

when these points are determined, the path between the two points should be explicit [6] in this experiment, A quintic polynomial is used with default zero boundary conditions for velocity and acceleration with 0.01s sampling time.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (1)$$

A 5-order polynomial has 6 coefficients so he can satisfy the six constraints given by the expression
six constraints:

$$\theta_0 = a_0 \quad \theta_f = a_0 + a_1 t_f + a_2 t_f{}^2 + a_3 t_f{}^3 + a_4 t_f{}^4 + a_5 t_f{}^5 \qquad (2)$$

This research was funded by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education(1345253125)and Technology Innovation Program(10073147, Development of Robot Manipulation Technology by Using Artificial Intelligence) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).and conducted under 「the Competency Development Program for Industry Specialists」 of the Korean Ministry of Trade, Industry and Energy operated by Korea Institute for Advancement of Technology (KIAT). (No. P0008473, The development of high skilled and innovative manpower to lead the Innovation based on Robot)

Where, θ_0, θ_f Constraints between the starting point and the final point position,

$$\dot{\theta}_0 = a_1 \qquad \dot{\theta}_f = a_1 + 2a_2 t_f + 3a_3 t_f^{\ 2} + 4a_4 t_f^{\ 3} + 5a_5 t_f^{\ 4}$$

$$\dot{\theta}_0 = 0 \qquad \dot{\theta}_f = 0 \tag{3}$$

Where, $\dot{\theta}_0$ and $\dot{\theta}_f$ Constraints between the starting point and the final point velocity, default zero boundary conditions,

$$\ddot{\theta}_0 = 2a_2 \qquad \ddot{\theta}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^{\ 2} + 20a_5 t_f^{\ 3}$$

$$\ddot{\theta}_0 = 0 \qquad \ddot{\theta}_f = 0 \tag{4}$$

Where, $\ddot{\theta}_0$ and $\ddot{\theta}_f$ Constraints between the starting point and the final point accelerated velocity, default zero boundary conditions,
These constraints determine the solution of six equations and six unknown linear equations,

$$a_0 = \theta_0 \quad (5) \qquad a_1 = \dot{\theta}_0 \quad (6) \qquad a_2 = \frac{\ddot{\theta}_0}{2} \quad (7)$$

$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)\,t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)\,t_f^{\ 2}}{2t_f^{\ 3}} \tag{8}$$

$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)\,t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)\,t_f^{\ 2}}{2t_f^{\ 4}} \tag{9}$$

$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)\,t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)\,t_f^{\ 2}}{2t_f^{\ 5}} \tag{10}$$

Where, $a_0, a_1, a_2, a_3, a_4, a_5, a_6$,Eq.(5)-Eq.(10) are solutions of 6 constraints and put them in Eq.(1) can find the fifth-order polynomial parameter of any joint angular position to the desired end position.

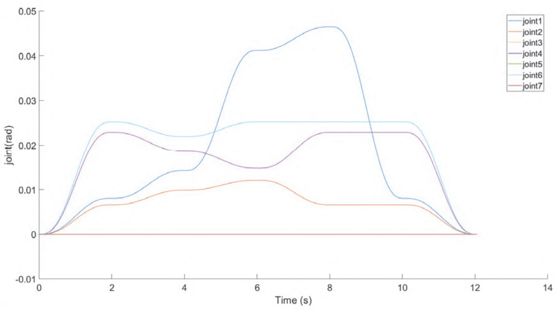

Fig.1. Designed Joint Variable

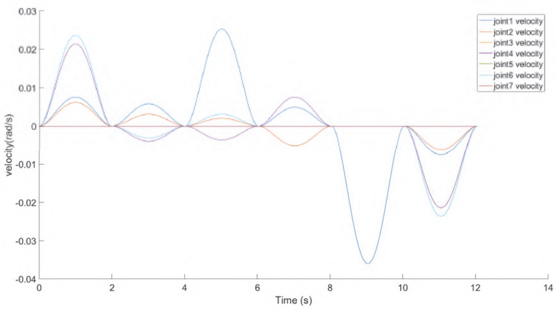

Fig.2. Designed Velocity Variable

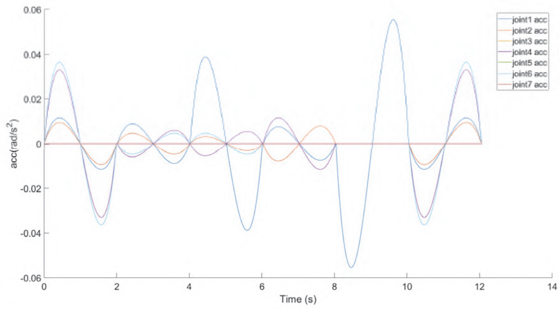

Fig.3. Designed Acceleration Variable

By solving the equation to obtain the coefficients, numerous transition points between middle points can be calculated with 0.01s sampling time. As shown in fig.1, fig.2 and fig.3, the designed joint angle, velocity and acceleration have the characteristic of smoothness and continuity, and the whole trajectory is shown in fig.4.

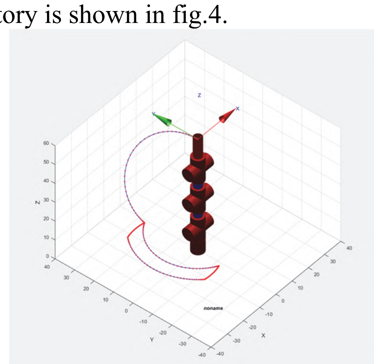
Fig.4. Trajectory of Simulation

Newton Euler recursive method [7],[8] is applied to implement forward, and inverse dynamics. The presumed dynamics parameters needed by the method are presented in table 3.

Table 3. Dynamics parameters table

Axis	1	2	3	4	5	6	7
Mass m_i/kg	3	3	2.5	2.5	2	2	2
Centroid ${}^i_i c/m$	[0 0 0]	[0 0 0.124]	[0 0 0]	[0 0 0.075]	[0 0 0]	[0 0 0.075]	[0 0 0.075]
Rotation inertia ${}^i_{c_i}I/kg\cdot m^2$							
I_{xx}	0.5	0.5	0.5	0.5	0.4	0.3	0.3
I_{yy}	0.5	0.5	0.5	0.5	0.4	0.3	0.3
I_{zz}	0.5	0.5	0.5	0.5	0.4	0.3	0.3

At this point, the preliminary work for control simulation has been completed.

III. Control Method

Manipulator control simulation based on MATLAB s-function and Robotics Toolbox. 30% uncertainties of parameters and a sinusoidal torque disturbance whose amplitude depends on 10% of the maximum control torque of a particular axis are imposed on the system.

$$\tau_d = A_j sin(2\pi t) \quad (11) \qquad A_j = 0.1\tau_{j,max} \quad (12)$$

$$m_u = m_0 - 0.3m_0 \quad (13) \qquad I_u = I_0 + 0.3I_0 \quad (14)$$

$$c_u = c_0 + 0.3c_0 \qquad (15)$$

Where Eq.(11) ,(12) is 10% of the maximum control torque of the each link disturbance. Where Eq.(13) (14) (15) ,I， m , c ,are uncertain dynamic parameters used to calculate control input τ.

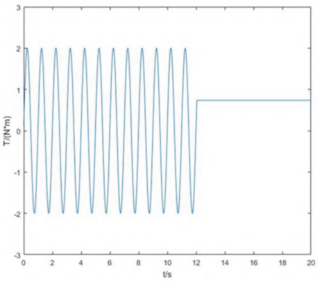

Fig.5.Disturbance Torque for Link-2

It is of great significance to select a control method according to a system. The primary targets in designing control system are stability, robustness and small tracking error.

calculated torque control(ctc)is a method for calculating the dynamic equation of the robot and controlling it with the PD controller [9],[10] Due to the increased calculation results of the robot dynamics equation, nonlinear and interference factors such as Coriolis, centrifugal force and robotic gravity are compensated. The dynamic model of the robot is as follows,

$$\mathrm{D}(\mathrm{q})\ddot{q} + C(q,\dot{q}) + g(q) = \tau \qquad (16)$$

The control input of the PD controller can be expressed as:

$$\tau=\widehat{\mathrm{D}}\left(\ddot{\mathrm{q}}_{\mathrm{d}}+\mathrm{K}_{\mathrm{v}}(\dot{\mathrm{q}}_{\mathrm{d}}-\dot{\mathrm{q}})+\mathrm{K}_{\mathrm{p}}(\mathrm{q}_{\mathrm{d}}-\mathrm{q})\right)+\hat{C}(\mathrm{q},\dot{\mathrm{q}})+\widehat{\mathrm{G}}(\mathrm{q}) \qquad (17)$$

Where, in Eq.(17) added the calculation results of the robot dynamics equation, compensating for such things as Coriolis, centrifugal force and robot gravity. $\dot{q}_d-\dot{q}$ is velocity error vector. q_d-q is position error vector.

Sliding mode controller is a powerful nonlinear robust controller to handle uncertainties of a mechanical system. [11],[12] This controller is used to control highly nonlinear systems, Chattering phenomenon and nonlinear equivalent dynamic formula in uncertain dynamic parameters are two main shortcomings of the pure sliding mode controller. The main reason for choosing this controller is its wide range of acceptable control performance and can solves two of the most important challenging topics in control which are stability and robustness,

$$s = ce + \dot{e} \quad (\mathrm{e} = \theta - \theta_d, \dot{e} = \dot{\theta} - \dot{\theta}_d) \qquad (18)$$

Design system sliding surface. e is position error vector.$\dot{e}$ is velocity error vector,

$$\dot{s} = -Ksat(s) \qquad (19)$$

The constant k represents the rate at which the motion point of the system approaches the switching plane s=0. The smaller k is, the lower the speed will be; the larger k is, the faster speed will be when the moving point reaches the switching surface, resulting in the larger chattering,

$$\tau = -M(\theta)(Ksat(s) + c\dot{e}) + M(\theta)\ddot{\theta}_d + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \qquad (20)$$

Where, M is the robot arm n by n inertia matrix，C is n by 1 centrifugal force and Coriolis force vector，G is n by 1 Grav ity vector,

According to the Lyapunov stability theory [13] to ensure that the system is stable and the error will converge to zero in finite time,

$$\dot{s}s < 0 \qquad (21)$$

Approaching law satisfies sliding arrival conditions K is large enough to overcome the disturbance.

IV. ANALYSIS SIMULATION

Table4.7-Link ctc

Axis	1	2	3	4	5	6	7
K_p	100	120	100	120	80	80	80
K_v	20	30	25	30	25	25	25

Table 5.7-Link smc

Axis	1	2	3	4	5	6	7
K	4	12	4	12	0.5	1	0.5
c	40	35	40	35	40	30	40

Trajectory errors for axis-6 by using two methods are shown in Figs.6, 7, respectively. The simulation results show that there are almost no steady-state error in the situation we set. But the switching control in sliding mode control solves the error caused by the model uncertainties and nonlinear terms to the greater extent than computed torque method. Since the SMC controller itself has a robust structure to noise disturbance and parametric uncertainties.

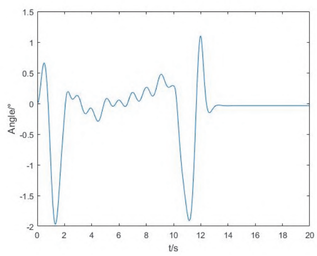

Fig.6. Trajectory error of axis-6 by ctc

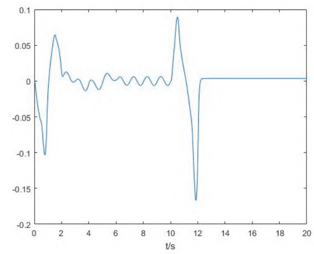

Fig.7. Trajectory error of axis-6 by smc

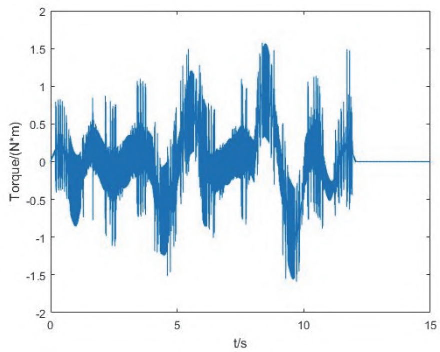

Fig.8. Extreme situation For Axis-7

Since the system continuously switches on the sliding surface,the high-frequency control signal is input into the system, which easily causes the robotic arm to shake, exciting the structure resonance, and causing system instability. When k is increased until it is 10 times than former one, a typical SMC input can be observed as shown in Fig.8.

V. CONCLUSION

In this study, kinematics and dynamics model of the 7-DOF manipulator are established by using MATLAB Robotics Toolbox, greatly simplifying the workload of control simulation. The experiment result shows that the method is convenient and effective, providing a fast simulation way for a 7-DOF robot arm or even the ones with any degree of freedom. Due to the fast computer operation and short simulation time, this method contributes to accelerating the optimized design process of the controller. Moreover, the whole simulation process is offline, which also improves the security of the control experiment. In the future, various system identification methods can be used to obtain more accurate dynamics parameters. Besides, the friction model can also be introduced and imported into the robot toolbox to obtain a more accurate robotic arm model. By combining the method of this paper, the simulation results closer to the reality ones can be obtained.

REFERENCES

[1] P. I. Corke, "A robotics toolbox for MATLAB," in IEEE Robotics & Automation Magazine, vol. 3, no. 1, pp. 24-32, March 1996. doi: 10.1109/100.486658

[2] J. M. Hollerbach, Optimum kinematic design for a seven degree of freedom manipulator, Tech. rep., MITArticial Intellegence Laboratory, Cambridge (1985).

[3] R. Manseur, K.L. Doty: A robot manipulator with 16real inverse kinematic solutions, Int. J. Robot. Res.8(5), 75–-79 (1989)

[4] R.P. Paul, B.E. Shimano, G. Mayer: Kinematic control equations for simple manipulators, IEEE Trans. Syst. Man Cybern. SMC-11(6), 339–-455(1981)

[5] William A. Wolvich, “ROBOTICS: Basic Analysis and Design”, CBS College Publishing, 1987

[6] C.S.Lin and P.R. Chang, “Joint Trajectory of Mechanical Manipulators for Cartesian Path Approximation, ” IEEE Transactions on Systems,Man,and Cybernetics,Vol.SMC-13,1983

[7] K.W. Buffinton: Kane ’ 's Method in Robotics. In:Robotics and Automation Handbook, ed. by T.R. Kurfess (CRC, Boca Raton, FL 2005), 6-1 to 6-31

[8] T.R. Kane, D.A. Levinson: The Use of Kane’ 's Dynamical Equations in Robotics, Int. J. Robot. Res. 2(3),3–-21 (1983)

[9] J.J. Slotine, W. Li: On the adaptive control of robot manipulators, Int. J. Robot. Res. 6(3), 49–-59 (1987)

[10] F.L. Lewis, C.T. Abdallah, D.M. Dawson: Control of Robot Manipulators (Macmillan, New York 1993)

[11] C. S. Chen andW. L. Chen, “"Robust adaptive sliding-mode control using fuzzy modeling for an inverted-pendulum system,” " IEEE Trans. on Industrial Electronics, vol. 45, no.2, pp. 297-306, 1998. [click]

[12] Jinkun Liu, “Sliding Mode Control Using MATLAB”, pp. 1-29

[13] A.M Lyapunov,” On the General Problem of Stability of Motion,”