Reducing the Error of Manipulator Jacobian in the Control System

MOHAMMAD MAHDI FATEH & HASAN FARHANGFARD
Department of Electrical and Robotic Engineering
Shahrood University of Technology, Shahrood, Iran

Abstract: In this paper, the uncertainty of Jacobian matrix and its effects in the control system are considered. A new control approach is then provided for reducing error in the case of trajectory tracking in the task space. This control approach is based on using feedback linearization. The joint angles, position, velocity and acceleration for the end-effector are required to feedback. Although, the case study in this paper is a 2-link manipulator, but it can be generalized for multi degrees of freedom robots. The control system is simulated and considered for canceling the error.

Key-Words: manipulator Jacobian, uncertainty, feedback linearization, tracking, canceling error.

1 Introduction

Robot dynamic equations are highly nonlinear. For the reason of parametric uncertainties and the un-modeled dynamic, it is difficult to derive the exact system model. Using feedback linearization approach and then applying a linear controller such as the PD and PID controller are effective for set point control, despite the nonlinearity and uncertainty of the robot dynamics [1-2].

End-effector is controlled in the task space while the actuators operate in the joint space. Therefore, it is required to transfer the joint space to task space using the Jacobian matrix [3]. The Jacobian error produces output error in the control system. In the robot control it is very often to use the Jacobian matrix. However, the exact Jacobian matrix is not available.

Because of the production limits, industrial product parameters will be slightly different with the design. On the other hand, considering the different conditions in load, geometry and robot component dimensions, Jacobian matrix involves the errors. A PD control law based on the approximate Jacobian has been provided for set point control of robots with uncertainties in Jacobian matrix [4,5,6]. Required conditions for the bound of the estimated Jacobian matrix and stability conditions with feedback gains are presented. Despite the fact that the Jacobian is not certain, it is required to provide the asymptotic stability for the system [7].

Control laws have the best operation by using the exact model of the system. However, the exact model of robot is not available. Moreover, a robot performs different tasks, so its model may vary. For example: when a robot picks up several tools with different dimensions, or gripping point, therefore the overall kinematics and dynamic of robot changes.

In more control methods, the Jacobian matrix is used and the robot control law is defined in the joint space. Singularity is one of the most important problems which must be considered for the robot control. In the case of singularity, the determinant of Jacobian matrix becomes zero and as a result the system will be out of control. Therefore, a robot can track a trajectory where the Jacobian matrix is non-singular.

In studying uncertainty, it is propounded that Human beings act intelligently while they don't have an accurate knowledge of the environment. For example, with the help of our eyes, we are able to pick up a new tool or object and manipulate it skillfully to accomplish a task.

In addition, Humans are able to learn and resolute their uncertainties by using previous experiences. For example, after using an unknown tool for a few times, we seem to have a better knowledge of it and we are able manipulate it more skillfully. Therefore it is not require the exact knowledge of kinematics and Jacobian matrix. In order to overcome dynamic and kinematics uncertainties, we should apply suitable control laws.

2 Jacobian error

Manipulator Jacobian or Jacobian matrix is an important factor for robot control and analysis. Jacobian matrix is used for planning smooth trajectory, determining the singular cases, and transformation from joint space to work space.

Parametric error in system will cause Jacobian error, and then it causes velocity error in the Cartesian space. Jacobian matrix is derived using forward kinematic equations as follows.

$$\mathbf{T} = \text{Fkin}(\mathbf{q}) \quad (1)$$

where, $\mathbf{q} \in \mathbf{R}^{\mathbf{n}}$ is joint variable vector, Fkin is the forward kinematic function, T is a transformation matrix defined as $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$, where $\mathbf{R}$ is a rotation matrix to show the direction of the end-effector and d is the position of the end-effector.
Jacobian matrix is defined as:

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathbf{v}} \\ \mathbf{J}_{\mathbf{w}} \end{bmatrix} \dot{\mathbf{q}} \quad (2)$$

$\mathbf{v}$ and $\boldsymbol{\omega}$ are the linear and angular velocity of the end-effector, respectively, $\mathbf{J}_{\mathbf{v}}$ and $\mathbf{J}_{\mathbf{w}}$ are Jacobian matrix related to linear and angular velocity, respectively. In this paper, we consider to track the end-effector position in the work space, so $\mathbf{J}_{\mathbf{v}}$ the Jacobian matrix is simply denoted by J and the end effector velocity $\mathbf{v}$ is denoted by $\dot{\mathbf{X}}$.
If we show Jacobian error as $\boldsymbol{\delta}\mathbf{J}$ and the approximate Jacobian matrix as $\hat{\mathbf{J}}$, then we have:

$$\mathbf{J} = \hat{\mathbf{J}} + \boldsymbol{\delta}\mathbf{J} \quad (3)$$

Substituting Equation (3) into Equation (2), we have:

$$\begin{aligned} \dot{\mathbf{X}} &= (\hat{\mathbf{J}} + \boldsymbol{\delta}\mathbf{J})\dot{\mathbf{q}} \\ &= \hat{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\dot{\mathbf{q}} \end{aligned} \quad (4)$$

As a result, approximate velocity is obtained as:

$$\dot{\hat{\mathbf{X}}} = \hat{\mathbf{J}}\dot{\mathbf{q}} \quad (5)$$

Therefore, the velocity error will become as:

$$\boldsymbol{\delta}\dot{\mathbf{X}} = \boldsymbol{\delta}\mathbf{J}\dot{\mathbf{q}} \quad (6)$$

Then, the position error produced by Jacobian error is derived as:

$$\boldsymbol{\delta}\mathbf{X} = \int \boldsymbol{\delta}\mathbf{J}\dot{\mathbf{q}}\, \mathrm{dt} \quad (7)$$

3 Designing the controller

Robot dynamic equation can be expressed in joints space as [3].

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (8)$$

where $\mathbf{q}$ is the joint variable vector, $\boldsymbol{\tau}$ is the applied joint torque to the robot, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{G}(\mathbf{q})$ is the gravitational force vector, $\mathbf{F}(\dot{\mathbf{q}})$ is friction force vector, $\mathbf{V}(\mathbf{q},\dot{\mathbf{q}})$ include the coriolis and centripetal terms.
A control law is defined using the feedback linearization approach according to Equation (8).

$$\mathbf{M}\mathbf{p} + \mathbf{h} = \boldsymbol{\tau} \quad (9)$$

where $\mathbf{h}$ is $\mathbf{h} = \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}})$
Applying the above control law will result in a linear system as follows.

$$\mathbf{p} = \ddot{\mathbf{q}} \quad (10)$$

Derivative of $\dot{\mathbf{X}}$ in Equation (4), will be:

$$\ddot{\mathbf{X}} = \hat{\mathbf{J}}\,\ddot{\mathbf{q}} + \dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \boldsymbol{\delta}\dot{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\ddot{\mathbf{q}}$$
(11)

And, then:

$$\ddot{\mathbf{q}} = \hat{\mathbf{J}}^{-1}[\ddot{\mathbf{X}} - (\dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \boldsymbol{\delta}\dot{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\ddot{\mathbf{q}})] \quad (12)$$

Thus, from Equations (10) and (12), the control input is derived as:

$$\mathbf{p} = \hat{\mathbf{J}}^{-1}[\ddot{\mathbf{X}} - (\dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \boldsymbol{\delta}\dot{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\ddot{\mathbf{q}})] \quad (13)$$

To control robot in task space for tracking a trajectory, we can apply the following control low in Cartesian space.

$$\ddot{\mathbf{X}} = \ddot{\mathbf{X}}_\mathbf{d} + \mathbf{K}_\mathbf{v}(\dot{\mathbf{X}}_\mathbf{d} - \dot{\mathbf{X}}) + \mathbf{K}_\mathbf{p}(\mathbf{X}_\mathbf{d} - \mathbf{X}) \quad (14)$$

Let $\mathbf{e} = \mathbf{X}_\mathbf{d} - \mathbf{X}$, where $\mathbf{e}$ is the position error. Therefore we have:

$$\ddot{\mathbf{e}} + \mathbf{K}_\mathbf{v}\dot{\mathbf{e}} + \mathbf{K}_\mathbf{p}\mathbf{e} = \mathbf{0} \quad (15)$$

As $t \to \infty$, the position error approaches zero asymptotically. Substituting Equation (14) into Equation (13) and then substituting equation (13) into Equation (9) will result in a control low as:

$$\begin{aligned}\boldsymbol{\tau} = \mathbf{M}\,\hat{\mathbf{J}}^{-1}[\,\ddot{\mathbf{X}}_\mathbf{d} + \mathbf{K}_\mathbf{v}(\dot{\mathbf{X}}_\mathbf{d} - \dot{\mathbf{X}}) + \mathbf{K}_\mathbf{p}(\mathbf{X}_\mathbf{d} - \mathbf{X}) \\ -(\dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \boldsymbol{\delta}\dot{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\ddot{\mathbf{q}})] + \mathbf{h}\end{aligned} \quad (16)$$

From equation (11), we have:

$$\hat{\mathbf{J}}\,\ddot{\mathbf{q}} - \ddot{\mathbf{X}} = -(\dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} + \boldsymbol{\delta}\dot{\mathbf{J}}\dot{\mathbf{q}} + \boldsymbol{\delta}\mathbf{J}\ddot{\mathbf{q}}) \quad (17)$$

As a result of substituting Equation (17) into Equation (16):

$$\boldsymbol{\tau} = \mathbf{M}\,\hat{\mathbf{J}}^{-1}[\,\ddot{\mathbf{e}} + \mathbf{K}_\mathbf{v}\dot{\mathbf{e}} + \mathbf{K}_\mathbf{p}\mathbf{e} + \hat{\mathbf{J}}\,\ddot{\mathbf{q}}] + \mathbf{h} \quad (18)$$

The control law is arranged as:

$$\boldsymbol{\tau} = \mathbf{M}\,[\hat{\mathbf{J}}^{-1}(\,\ddot{\mathbf{e}} + \mathbf{K}_\mathbf{v}\dot{\mathbf{e}} + \mathbf{K}_\mathbf{p}\mathbf{e}) + \ddot{\mathbf{q}}] + \mathbf{h} \quad (19)$$

In order to make the control low given by Equation (19), the variables named as $\dot{\mathbf{q}}$, $\mathbf{X}$, $\dot{\mathbf{X}}$, $\ddot{\mathbf{X}}$ should be feedback in control system. The control system is shown in Fig.1.

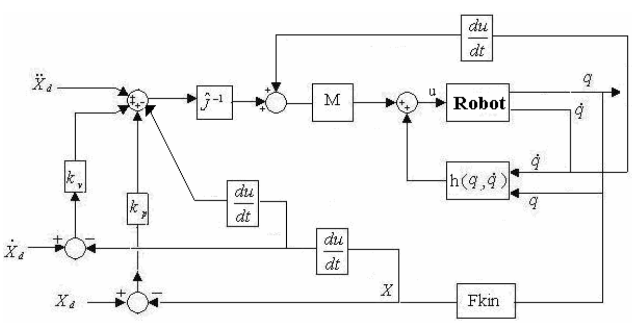

Fig.1 block diagram of the control system.

In this paper, although the case study is a 2-link manipulator shown in Fig.2, but it can be generalized for multi degrees of freedom robots.

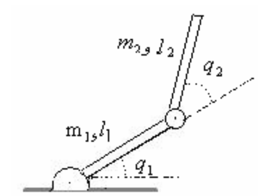

Fig.2 the 2-link manipulator.

Forward kinematic Equations are given by:

$$\begin{aligned} x_1 &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ x_2 &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{aligned} \quad (20)$$

Where $\mathbf{X} = [x_1 \quad x_2]^T$, l_1 and l_2 are the link lengths, m_1 and m_2 are the link masses, q_1 and q_2 are the joint variables, respectively. And, the Jacobian matrix of 2-link manipulator is then derived as:

$$\mathbf{J} = \begin{bmatrix} -\mathrm{l}_1 \sin \mathrm{q}_1 - \mathrm{l}_2 \sin(\mathrm{q}_1 + \mathrm{q}_2) & -\mathrm{l}_2 \sin(\mathrm{q}_1 + \mathrm{q}_2) \\ \mathrm{l}_1 \cos \mathrm{q}_1 + \mathrm{l}_2 \cos(\mathrm{q}_1 + \mathrm{q}_2) & \mathrm{l}_2 \cos(\mathrm{q}_1 + \mathrm{q}_2) \end{bmatrix} \quad (21)$$

In this simulation, m_1 is 2kg and m_2 is 1kg, l_1 is 1m and l_2 is 5m.

4 Simulation

If it is assumed that $\boldsymbol{\delta}\mathbf{J} = k\mathbf{J}$, $0 \langle k \langle 1$ and $\hat{\mathbf{J}} = (1-k)\mathbf{J}$. Substituting $\hat{\mathbf{J}}$ into Equation (16), the control law is written as follows:

$$\begin{aligned}\tau = \mathrm{M}\,\hat{\mathrm{J}}^{-1}[\,\ddot{\mathrm{X}}_\mathrm{d} + \mathrm{k}_\mathrm{v}(\dot{\mathrm{X}}_\mathrm{d} - \dot{\mathrm{X}}) + \mathrm{k}_\mathrm{p}(\mathrm{X}_\mathrm{d} - \mathrm{X}) - \\ ((1+\mathrm{k})\dot{\hat{\mathrm{J}}}\dot{\mathrm{q}} + \mathrm{k}\mathrm{J}\,\ddot{\mathrm{q}})] + \mathrm{h}\end{aligned} \quad (22)$$

The simulation results are obtained for different values of k. The reference signal is shown in Fig.3.

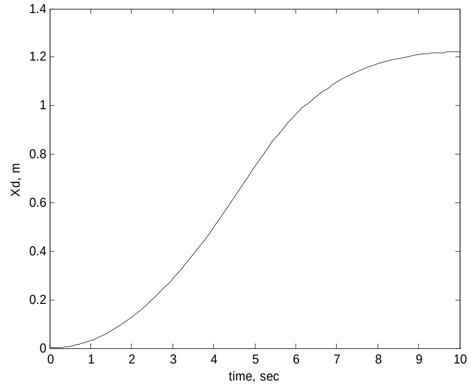

Fig.3 reference signal.

Choosing $k_v = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, $k_p = \begin{bmatrix} 7 & 0 \\ 0 & 7 \end{bmatrix}$, k=0.6, tracking error in task space is presented in Fig. 4. The error has an overshoot about 0.21 at time 0.5 sec. and it approaches zero after 1sec.

Now, we increase k. So, by selecting k=0.8, $k_v = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, $k_p = \begin{bmatrix} 7 & 0 \\ 0 & 7 \end{bmatrix}$, the tracking error is presented in Fig.5. In comparison with Fig.4, the overshoot of error increases to 0.3m at time 0.4 sec and it reaches to zero after 1sec. Anyway, the position error approaches zero.

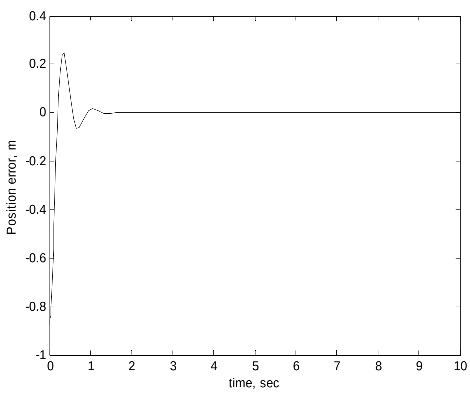

Fig.4 The position error.

To consider the effect of k_p on the control system, we increase it. The tracking error is shown in Fig.6 by given $k_v = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, $k_p = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$, k=0.6. In comparison with Fig.5, the position error approaches zero with no overshoot. The control system operates well to provide a desired output.

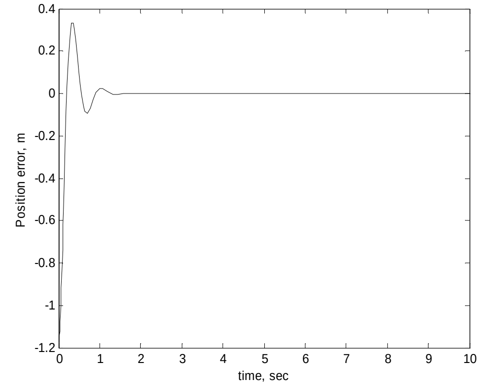

Fig.5 The position error.

Now, we consider k_v. Tracking error is presented in Fig.7 where k=0.6, $k_v = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}$, $k_p = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$. The error has an overshoot that is not much. The error approaches zero well.

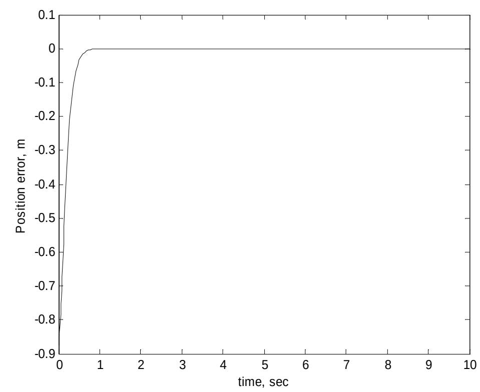

Fig.6 The position error.

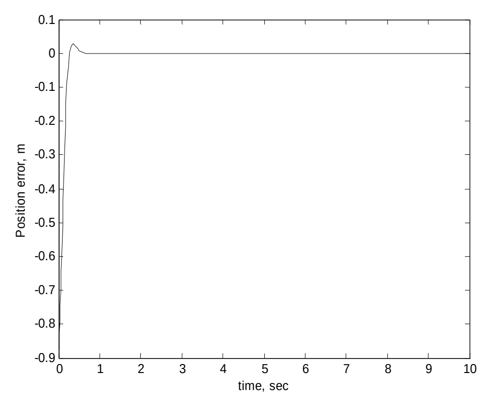

Fig.7 The position error.

Finally in the case of k=0.6, $k_v = \begin{bmatrix} 120 & 0 \\ 0 & 120 \end{bmatrix}$, $k_p = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$, the tracking error shown in Fig.8. It has no overshoot and approaches zero well.

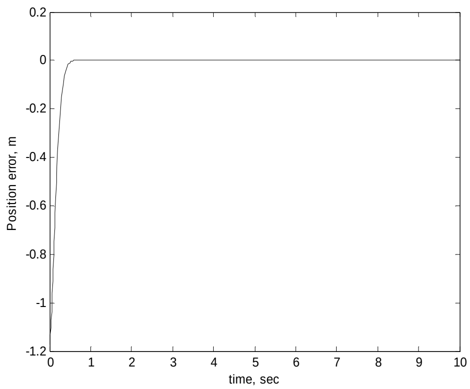

Fig.8 The position error.

5 Conclusion

The Jacobian error causes the error for tracking a desired trajectory in task space. A control law has been developed to cancel the Jacobian error using feedback linearization. Simulation results have been provided and the tracking error has been considered to approach zero. To implement the control law, the acceleration of the joint angles, and also the position and velocity acceleration of the end-effector in the task space are required to feedback.

References:

[1] Caccavale F., Chiaverini S. Siciliano B., Second-Order Kinematic Control of Robot Manipulators with Jacobin Damped Least-Squares Inverse: Theory and Experiments 188 *IEEE/ASME TRANS. ON MECHATRONICS*, Vol. 2, No. 3, SEP. 1997, pp. 188-194.

[2] Cheah C. C, Kawamura S, Arimoto K. Lee , PID Control of Robotic Manipulator with Uncertain Jacobin Matrix, *Conference on Robotic & Automation, Seoul, Korea,* May 21-26 , 2001,pp. 494-499.

[3] Spong M. W., Vidyasagar M., *Robot Dynamics And Control*, John Wily And Sons , 1989.

[4] Cheah H. Y., Han S., Kawamura S., Arimoto , Grasping and Position Control for Multi-fingered Robot Hands with Uncertain Jacobin Matrices, *Proceedings of the American Control Conference Arlington*, V A June 25-27, 2001, pp. 2403-2408.

[5] Yazarel H, C. C. Cheah, Adaptive Visual Servoing of Robots with Uncertain Gravity Regressor and Jacobin Matrices, *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, Vol. 46, No. 8, AUGUST 2001, pp. 652-657 .

[6] Cheah C. C., Kawamura S., Arimoto S., and Lee K., H Tuning for Task-Space Feedback Control of Robot with Uncertain Jacobin Matrix, *TRANS. ON AUTOMATIC CONTROL*, Vol. 46, No. 8, AUGUST 2001, pp. 1313-1318.

[7] Cheah C. C, Slotine Liu J.J.E., Approximate Jacobin Adaptive Control for Robot Manipulators , *Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orleans*. LA April 2004, pp.3075-3080.