Obstacle Avoidance of a 7-DOF Robotic Manipulator

Name : Youngkyoung Kong
Major : Aerospace Enginnering
Student ID : 2018-13561

1. Introduction

The research of obstacle avoidance of a redundant robotic manipulator is conducted actively. There are various methodologys designing collision-free path of a manipulator including inverse kinematic solution method[1], RRT algorithm[2][3], and artificial potential field[4][5], and so on. Recently, rapid developments in artificial intelligence make it possible to path-plan obstacle avoidance trajectory[6],[7].

In this project, artificial potential field method is used to make collision-free path of the manipulator.

2. Backgrounds

- Jacobian

Let $\boldsymbol{x}$ and $\boldsymbol{q}$ represent the position and the orientation of the manipulator and the joint variable vector at given configuration, respectively. Then, we can express the relationship between x and q using unknown nonlinear function.

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}) \quad (1)$$

If there are n number of joints and m number of representations, then (1) represents as follows.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_m(q) \end{bmatrix} \quad (2)$$

When the joint undertook a set of infinitesimally small displacements $\delta \boldsymbol{q}$, there will be a set of infinitesimally small change of the position and orientation $\delta \boldsymbol{x}$, corresponding to it. Then we can derive the relationship between $\delta \boldsymbol{x}$ and $\delta \boldsymbol{q}$ also, with direct differentiation.

$$\begin{gathered} \delta x_1 = \frac{\partial f_1}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_1}{\partial q_n}\delta q_n \\ \vdots \\ \delta x_m = \frac{\partial f_m}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_m}{\partial q_n}\delta q_n \quad (3) \end{gathered}$$

The above equation (3) can be written in vector form as follows.

$$\delta \boldsymbol{x} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \delta \boldsymbol{q} \quad (4)$$

The matrix in the above relationship is called the Jacobain matrix $J(\boldsymbol{q})$.

$$\delta \boldsymbol{x} = J(\boldsymbol{q})\delta \boldsymbol{q} \ \text{ or } \ \dot{\boldsymbol{x}} = J(\boldsymbol{q})\dot{\boldsymbol{q}} \quad (5)$$

$J(\boldsymbol{q})$ relates small displacements in small joint angle changes. Therefore, we can obtain a relationship between the velocities of the mechanism in joint and Catesian space.

- **Joint/Task Space Control**

Like our problem, consider a 7-DOF robotic manipulator. There are 7 joint angle variables, and the space defined by $\boldsymbol{q} = [q_1, q_2, \cdots, q_7]$ is called the joint space of the manipulator. And consider the end-effector of the manipulator.

Because the end-effector can move toward the x, y, z coordinates and rotate around the x, y, z coordinates, it has 6-DOF. So, we can represent the motion of the end-effector using 6 variables(3 position variables, 3 orientation variables), and the space defined by $\boldsymbol{x} = [x_1, x_2, \cdots, x_6]$ is called the task space or operational space of the manipulator.

When we control the mainpulator of the robot, there are two types of controller: joint space controller and task space controller. If we use joint space controller, specific joint angles are to be controlled as planned. Joint space control sets up control gains for each joint. Therefore, we cannot adjust the end-effector behavior directly. However, if we have desired joint angles, we do not have to calculate joint angles that make desired Cartesian coordinate. We just apply the torque input directly to the robot. And Joint space control typically uses high gain control to reject disturbance and to track the desired trajectory precisely. If we are to design the joint space controller, we have to consider joint space dynamics as follows.

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau \quad (6)$$

$M(q)$ is mass matrix, $V(q,\dot{q})$ is centrifugal or Coriolis effect term and $G(q)$ represents gravity effect. We should design appropriate input τ to accomplish our control goal.

On the other side, if we use task space controller, we may use it to achieve a certain task, for example, position, orientation or center of mass of a certain link. We control the robot directly at the task space by commanding control force for the task. However, we have to calculate the joint angles to command corresponding task which we want to achieve. And Jacobian is used to calculate it.

$$\delta q = J^{-1}\delta\boldsymbol{x} \quad (7)$$

Equation (7) is called instantaneous inverse kinematics and it uses the inverse of Jacobian. If the desired trajectory of end-effector $\boldsymbol{x}_d$ is given, $\delta\boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{x}$. And use the equation (7), we can derive δq. Then, $q_d = q + \delta q$. Commanding q_d to the motor drive in each time step, we can control the manipulator in the task space. We can use pseudo-inverse of Jacobian matrix when it is not invertible. There is another approach which uses force-torque relationship. If we apply principle of virtual work(virtual work is zero at static equilibrium, $\delta W=0$) to the articulated rigid body systems,

$$\tau^T \delta q + (-F)^T \delta x = 0 \quad (8)$$

and put equation (7) to above we can derive a force-torque relationship.

$$\tau = J^T F \quad (9)$$

From equation (9), we can derive operational space dynamics as follows. Starting from joint space dynamics (6),

$$M\ddot{q} + V + G = \tau \quad (6)$$

multiply by M^{-1} (M : positive definite matrix).

$$\ddot{q} + M^{-1}V + M^{-1}G = M^{-1}\tau \quad (10)$$

using equation (5),

$$\dot{x} = J\dot{q} \;\Rightarrow\; \ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (11)$$

put (11) and (9) into (10), then,

$$\ddot{x} - \dot{J}\dot{q} + JM^{-1}V + JM^{-1}G = JM^{-1}\tau$$
$$\ddot{x} - \dot{J}\dot{q} + JM^{-1}V + JM^{-1}G = JM^{-1}J^T F \quad (12)$$

If we define $\Lambda := JM^{-1}J^T$,

$$\ddot{x} - \dot{J}\dot{q} + JM^{-1}V + JM^{-1}G = \Lambda^{-1}F \quad (13)$$

If Λ is a nonsingular matrix,

$$\Lambda\ddot{x} + \Lambda(-\dot{J}\dot{q} + JM^{-1}V) + \Lambda JM^{-1}G = F$$
$$\Lambda(q)\ddot{x} + \mu(q, \dot{q}) + p(q) = F \quad (14)$$

Equation (14) is a operational space dynamics where x is end-effector position and orientation, Λ is a end-effector kinetic energy matrix, μ is end-effector centrifugal and Coriolis forces, p is end-effector gravity forces and F is end-effector generalized forces. We should design appropriate input F to accomplish our desired end-effector's position and orientation.

- Null Space Control

If the dimension of the joint space and the dimension of the task space are the same, we call the system is non-redundant. And the dimension of the joint space is larger than the dimension of the task space, we call the system is redundant. For non-redundant system, the joint angle that achieves certain position and orientation of the end-effector is unique. However, for redundant system, there are infinite number of solution of the joint angle that execute the same task.

Adopting the concept of null space to inverse kinematics, we can express equation (7) as follows.

$$\delta q = J^{\dagger}\delta x + N\delta q$$
$$\delta q = J^{\dagger} J\delta q + N\delta q$$
$$N = I - J^{\dagger} J \quad (15)$$

N means null space projection matrix, and it do not create acceleration in task space.

$$\therefore \delta q = J^{\dagger}\delta x + (I - J^{\dagger} J)\delta q_0 \quad (16)$$

In equation (16), the first term represents the particular solution and the second term represents the homogeneous solution and null space motion.

In terms of dynamics, we can express torque using null space also.

$$\tau = J^T F + \tau_{nullspace} \quad (17)$$

Given τ there is an unique F, and by using the relationship between equation (6) and (14),

$$F = \Lambda J M^{-1}\tau = \bar{J}^T\tau \quad (18)$$

where $\bar{J} := M^{-1}J^T\Lambda$ is called dynamically consistent generalized inverse. Therefore, the joint space dynamics of the redundant robot can be expressed as follows.

$$\tau = J^T F + (I - J^T\bar{J}^T)\tau_0 \quad (19)$$

And F is composed using the task space dynamics.

3. Problem Description and Solution

The robot model is a Franka Emika Panda manipulator which has 7 degrees of freedom. And consider a small rigid sphere. The manipulator should reach the goal position without colliding the ball while it moves. The initial configuration of the manipulator is $q_{init} = [0, -60^\circ, 0, -90^\circ, 0, 30^\circ, 0]^T$ and the goal position of the end-effector is $x_d = [0.3, -0.012, 0.52]^T$. The end-effector should move to the goal position maintaining the initial orientation. The position of the obstacle is $x_{ob} = [0.15, -0.012, 0.65]^T$.

To avoid an obstacle, we can use artificial potential field method. If we set repulsive field to

$$U_{ob} = \frac{1}{2}k_{ob}\left(\frac{1}{d(x)} - \frac{1}{d_0}\right)^2 \quad (20)$$

then the corresponding repulsive force becomes

$$F_{rep}{}^* = -\nabla U_{ob} \quad (21)$$

where $d(x)$ is the distance between end-effector and the obstacle, and d_0 is the distance of influence. d_0 is set by 0.15. Also, attractive force is set to be

$$F_{att}{}^* = k_v(\dot{x}_d - \dot{x}) \quad (22)$$

The problem is, artificial potential field method is selected to avoid the obstacle in Cartesian space(i.e. task space) but the input should be the form of torque. Therefore, we have to convert task space dynamics to joint space dynamics using force-torque relationship. Using equation (19), the controller is selected as follows.

$$\tau = J^T F + (I - J^T \bar{J}^T)\tau_0 + G \quad (23)$$

And if we neglect end-effector centrifugal and Coriolis forces, F can be represented as

$$F = \Lambda F_0^* \text{ where } F_0^* = \left[(F_{att}^* + F_{rep}^*)^T M^{*T} \right]^T \quad (24)$$

The end-effector gravity force is contained in gravity compensation term G. M^* represents the orientation control and it can be expressed as

$$M^* = -k_p \delta\Phi - k_v \omega \quad (25)$$

where $\delta\Phi$ means the difference between initial orientation(roll, pitch, yaw angle vector) and current orientation and ω means the difference between initial angular velocity and current angular velocity.

And we set

$$\tau_0 = M[k_p(q_{init} - q) - k_v \dot{q}] \quad (26)$$

Assume that there is a velocity limitation $\dot{x}_{\max} = 0.3$. Not to violate this constraint, there is a method called velocity saturation which prevent the manipulator to excess the velocity limitation.

$$\dot{x}_d = \begin{cases} \dfrac{k_p}{k_v}(x_d - x) & \left| \dfrac{k_p}{k_v}(x_d - x) \right| < \left| \dot{x}_{\max} \right| \\ \left| \dfrac{\dot{x}_{\max}}{x_d - x} \right| (x_d - x) & \left| \dfrac{k_p}{k_v}(x_d - x) \right| \geq \left| \dot{x}_{\max} \right| \end{cases} \quad (27)$$

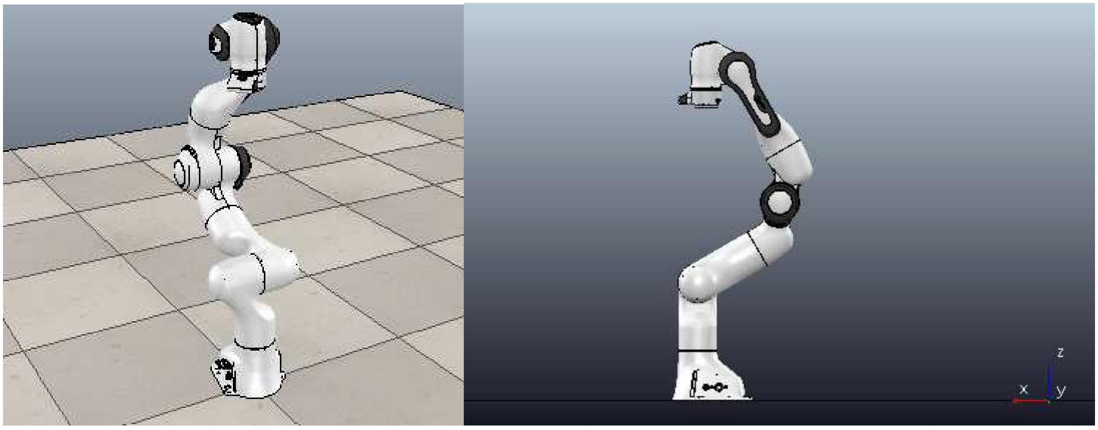

Figure 1. Initial Configuration of the Manipulator

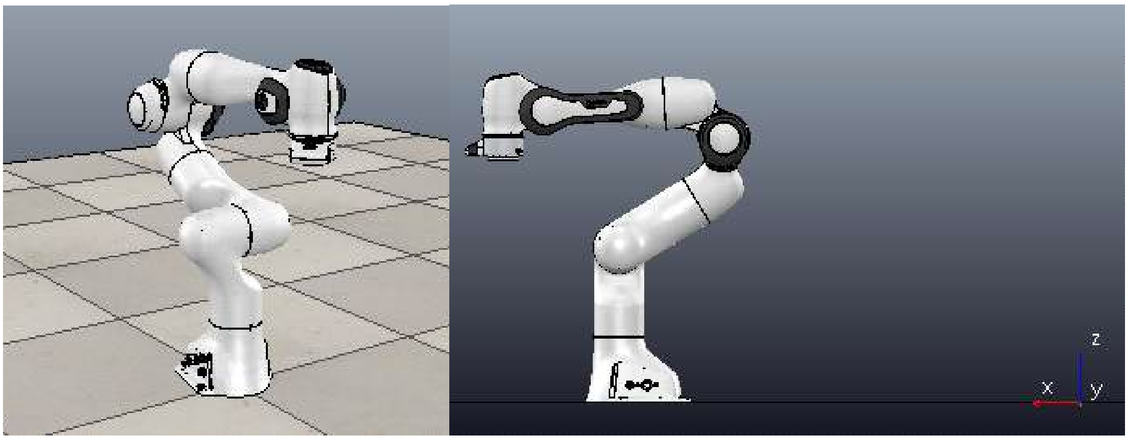

Figure 2. Goal Configuration of the Manipulator

4. Simulation Results

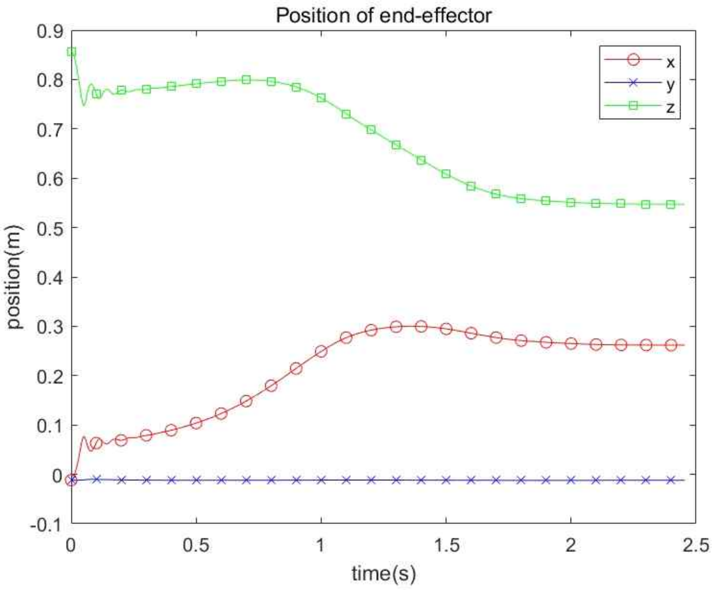

Figure 3. Position of the End-effector

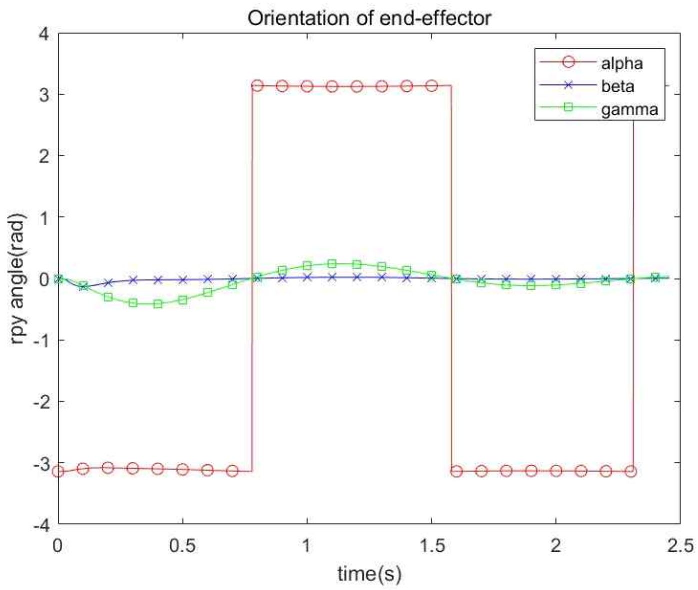

Figure 4. Roll, Pitch, Yaw Angle of the End-effector

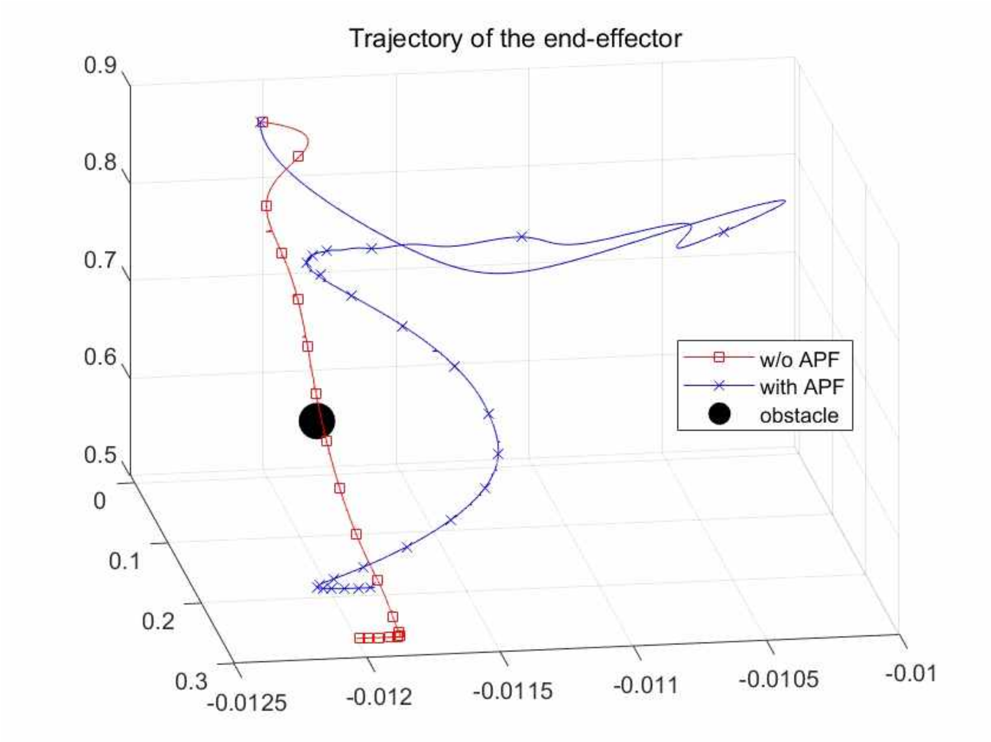

Figure 5. Comparison Between the Trajectory of the End-effector with/without Artificial Potential Field

With the designed controller, the end-effector of the manipulator could reach the desired goal position without colliding the obstacle. In Figure 3, we can check the position of the end-effector reached about $x_d = [0.3, -0.012, 0.52]^T$. And in Figure 4, we can also check that the orientation of the end-effector does not changed much compared to the initial orientation. It seems that the roll angle changed but we could say that it did not change because pi radian and -pi radian is the same. Most importantly, the manipulator avoids the obstacle and we can check the trajectory of the end-effector through Figure 5. We can see that the robot trajectory made a detour around the obstacle. If there did not exist artificial potential field, the manipulator would collide with the obstacle during its movement. However, motion planning with APF, we could control the robot safely from the rigid obstacle.

5. Conclusion

In this project, I designed the controller with artificial potential field method to avoid the fixed obstacle. I chose 7-DOF Franka Emika Panda robot to solve my problem. Assuming that the fixed rigid sphere obstacle is on the way of the movement of the end-effector, repulsive artificial potential field was made on the Catesian position of the obstacle. Designing joint space controller with null space control and converting it into task space force, I could command directly to each joint angle while the goal is related with the task space. Finally, I could check the trajectory of the end-effector made a detour around the obstacle, i.e. it did not collide with the obstacle while the controller without aritificial potential field cannot avoid the obstacle.

Reference

[1] Ge J,Sun F,Liu C.(2016) RRT-GD: An efficient rapidly-exploring random tree approach with goal directionality for redundant manipulator path planning.In: IEEE International Conference on Robotics and Biomimetics. Qing Dao, China. 1983-1988. https://doi.org/10.1109/ROBIO.2016.7866620

[2] Adiyatov O,Varol H A. (2017) A novel RRT*-based algorithm for motion planning in dynamic environments. In: IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan. 1416-1421.https://doi.org/10.1109/ICMA.2017.8016024

[3] Coelho F, Faria Pinto M, Carvalho de Souza J, et al. (2018) Direct-DRRT*: A RRTImprovement Proposal. In: 13th APCA International Conference on Automatic Control and Soft Computing. Ponta Delgada, Portugal.154-158.https://doi.org/10.1109/CONTROLO.2018.8514261

[4] Xie L, Liu S. (2018) Dynamic obstacle-avoiding motion planning for manipulator based on improved artificial potential filed. Control Theory and Applications, 35: 1239-1249. https://doi.org/10.7641/CTA.2018.70187

[5] Huang Y, Ding H, Zhang Y, et al. (2020) A Motion Planning and Tracking Framework for Autonomous Vehicles Based on Artificial Potential Field Elaborated Resistance Network Approach. IEEE Transactions on Industrial Electronics, 67: 1376-1386. https://doi.org/10.1109/TIE.2019.2898599

[6] Xia G, Han Z, Zhao B, et al.(2020) Local Path Planning for Unmanned Surface Vehicle Collision Avoidance Based on Modified Quantum

Particle Swarm Optimization. Complexity, 2020:1-15.https://doi.org/10.
1155/2020/3095426
[7] Liu J, Yang J, Liu H, et al.(2017) An improved ant colony algorithm for robot path planning. Soft Computing, 21:5829-5839.
https://doi.org/10.1007/s00500-016-2161-7
[8] Khatib, Oussama. "Advanced robotic manipulation." Rev. Lit. Arts Am 12 (2004): 9753.