Collision Avoidance Using Potential Field

University of Southern Denmark

shkum20@student.sdu.dk

shkum20@student.sdu.dk | 490532

Abstract

This report covers learning by demonstration of obstacle avoidance. Main idea is that adding external forces to the joint points of the robotic arm. If any obstacle (object) is close enough to the robot, the algorithm adds an external linear force and this force causes an acceleration in the same direction. The algorithm generates and converts this directional external force into joint torques. To detect obstacles and collisions in the robot environment. A simulation has been set up with a 7 DOF KUKA IIWA to test link collision avoidance using artificial potential field. In the environment a robot has a set trajectory with obstacle in the environment. Multiple tests have been performed on implementation to ensure satisfactory functional.

Contents

1 Introduction

Programming by demonstration can be useful in generalizing complex tasks and allow non-experts to quickly and effectively teach certain behaviors that can then be replicated by a robotic arm.However, changes in the robot environment can occur and therefore the robot would need to adapt to this new environment.

Here the collision avoidance of the redundant robot is placed in the enviornment with the obsatcle to avoid it. This report uses dynamic movement primitives (DMP) to generalize demonstrations made by a user, showing the desired positions of a robot end-effector. This obstacle avoidance works on the end-effector of the robot that the DMP is generated for.

Furthermore, modifying the angular velocity of the joints in the robot can achieve link avoidance. Simple implementations has been made, one for the link avoidance because it requires 7 degrees of freedom (DOF).A simulation has been made to test the functionality of the link avoidance.

2 Obstacle Avoidance

DMP's are good at generalizing the movement of a demonstration it has been trained on, but a problem occurs when obstacles are in the path of the DMP's trajectory. The formulation of a DMP in equation does not take into account if it has to avoid obstacles. Since the DMP is generated for the end-effector of a robot and this is in operational space it can easily be accounted for by adding an additional term (ϕ) to equation. See equation.

$$\tau \dot{z} = K(g - y) - Dz + f(x) + \Phi \tag{1}$$

ϕ acts as a force pushing the DMP away from the obstacles and can depend on a number of parameters such as velocity, position, etc. Two methods there investigated that both build on dynamic potential fields (DPF). The first is the work by where they use a DPF with a negative gradient. The second is the work by where they use the DPF with multiple distances to the obstacle. Section and will describe the two methods and section will explain which method was chosen.

These methods only take into account the avoidance for the DMP which is at the end-effector. This does not guarantee that the links of the robot do not collide with the obstacles. Since the links are in joints space it is not as straightforward as for the operational space end-effector.

2.1 Artificial Potential Field

Artificial potential field algorithm was proposed by O.khatib is a local path planning algorithm.It's basically creating a APF in the work space. It has a very basic concept to create an attractive forces to move towards the goal and having a repulsive force to move away from the obstacle in the workspace. The potential field methods treats the robot as point under the influence of artificial potential field. A minimum in the space act as an attractive force on the robot and obstacle act as a peak or repulsive force. APF smoothly guides robot towards the goal and hence avoiding obstacle.

Motion of manipulator in presence of potential field can be similar to the motion of particle as in presence of positive and negative field. It can be seen as manipulator as positive charge ; goal as a negative charge and obstacle as a positive charge. It can be like one repels and opposite one attracts.Gradient can be transition from the positive charge towards the negative.And the combination of this attractive and repulsive nature drives manipulator in the environment towards the goal while avoiding obstacle.

In the simplest case if robot is a point therefore robot orientation (θ) is neglected therefore the potential field is only 2-Dimensional. Let's consider differential potential field function as $U(q)$ and relative artificial force $F(q)$ acting on position = (x,y)

$$F(q) = -\nabla U(q) \tag{2}$$

where$\nabla U(q)$ denotes gradient vector of U at position q.

The artificial potential field where the robot moves is a scalar function U(q). $R^2 \rightarrow R$ generated by the superposition of attractive and repulsive potentials

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{3}$$

In the potential field method when robot navigate in the environment the attractive one is zero when robot approaches towards the goal and tends to increase when move away from the goal and for the case of repulsive potential it increases as move close to obstacle and tends to decrease it moves away from it. Forces which drive the robot towards the negative gradient of artificial potential are as follow :

$$F(q) = F_{att}(q) + F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \tag{4}$$

Forces (4) is a vector point in each direction of q, locally decreasing U.

2.2 Attractive forces

Attractive potential it's a standard parabola that grows quadratic-ally with the distance from the goal. This force can be divided into size components (Fa,mag) and direction components. Where X-axis direction can be expressed as a cosine and Y-axis direction as a sine, the x-axis direction and the y-axis direction can be expressed for attractive potential as :

$$U_{att}(q) = \frac{1}{2} K_{att} d_{goal}^2(q) \tag{5}$$

Here $d_{goal}(q) = ||q - q_{goal}||$ euclidean distance of robot to the goal. Gradient is vector field proportional to the difference given as :

$$\nabla U_{att}(q) = K_{att}(q - q_{goal}) \tag{6}$$

Therefore the force taken on the base of approach towards the negative gradient is :

$$F_{att}(q) = -\nabla U_{att}(q) = -K_{att}(q - q_{goal}) \tag{7}$$

This (7) moves the manipulator towards the goal and decreases when it reaches the goal.So it's force tends to zero as robot reaches the goal also the manipulator velocity. Generally the velocity is set minimal to avoid any overshot while motion.

2.3 Repulsive Forces

These forces drive the robot away from the obstacle. It grows as the manipulator approaches the obstacle and decreases as it moves away from the obstacle.It's define as sum of all the effect of repulsive forces i.e :

$$U_{rep}(q) = \sum_i U_{repi}(q) \tag{8}$$

Generally the influence of obstacle is limited in the environment space in it's vicinity up to a given distance. If the obstacle is far from robot it's less likely to repel.Moreover the magnitude of manipulator increases as it approaches the obstacle. Therefore the possible repulsive potential generated by obstacle i.e :

$$\mathbf{U}_{rep}(q) = \begin{cases} \frac{1}{2} K_{obsti}(\frac{1}{d_{obsti}(q)} - \frac{1}{d_0})^2 & d_{obsti}(q) \leq d_0 \\ \mathbf{0} & d_{obsti}(q) > d_0 \end{cases}, \tag{9}$$

Here $d_{obsti}(q)$ is minimal distance from q to obstacle i and K_{obsti} is constant and d_0 is threshold.

And negative gradient of repulsive potential is given by :

$$F_{repi} = -U_{repi}(q) \tag{10}$$

Therefore, the size of the repulsive force can be set arbitrarily by the user, but divide the repulsive force gain by do to make the size of the repulsive force smaller when the distance from the obstacle is far, and to increase the size of the repulsive force when the distance to the obstacle is close. Finally, since the robot must move in the opposite direction to the direction in which the obstacle is located.

2.4 Null Space

The equation that relates the end-effector velocity with the joint velocity through the Jacobian is:

$$\dot{x}_e = J_e \dot{q} \tag{11}$$

Where $\dot{\boldsymbol{x}}_e$ is the end-effector velocity and $\boldsymbol{J}_e$ is the Jacobian of the end-effector. Since the robot is a redundant manipulator the inverse of the Jacobian can not be found, instead the pseudoinverse is used to get a least-squares solution:

$$\dot{q} = J_e^+ \dot{x}_e \tag{12}$$

Since the robot is a redundant manipulator this also means that there are more solutions to 12.

$$\dot{q} = J_e^+ \dot{x}_e + \left(I - J_e^+ J_e\right) \mathcal{E} \tag{13}$$

The null space movement term in 13,

$$\left(I - J_e^+ J_e\right) \mathcal{E} \tag{14}$$

Will map the arbitrary velocity, $\boldsymbol{\mathcal{E}}$, into the null space of $\boldsymbol{J}$. This means that we can now find a joint velocity corresponding to a Cartesian velocity and add an arbitrary joint movement that does not affect the Cartesian velocity.

2.5 Limitation of Artificial Potential filed

1. There are certain limitation of the artificial potential algorithm. Getting stuck in local minima for example a c shape obstacle can lead to sometime cancellation of forces and stuck in local minima.
2. Sometime the sum of repulsive forces is greater then attractive making difficult to pass through a passage. Can have a unstable path for obstacle avoidance if there are multiple obstacle and can lead to different repulsive force.
3. The second extension of the repulsive potential does not consider the obstacles that will not closely affect the robot velocity. For example, it is irrelevant to consider the repulsive force generated by an obstacle in the back of the robot, when it is moving forward.

3 Implementation

In this implementation, KUKA IIWA robot arm is chosen as the testing platform. The configuration space is the degree-of-freedom joints q R, and the task-space is the end-effector position $x \varepsilon R^3$.The state is the position of the configuration space x = [q] and the action is the joint velocity, u = q . The trajectory is controlled through an inverse kinematic controller using velocity Control method.

3.1 Simulation

In this section we choose the redundant manipulator for the link avoidance. Redundant robot are known for the abundant joint actuator. Based on fact redundant robot has more degree of freedom. Which leads to more complexity of the controller design and complexity in design of robot inverse kinematics. To solve the kinematics of the Redundant robot the pseudo inverse of the jacobian to calculate minimum joint velocity for desired end effector velocity.The chosen robot is a 7-DOF Kuka IIwa which will be simulated in Pybullet. Simulation environment setup of the manipulator and a spherical obstacle

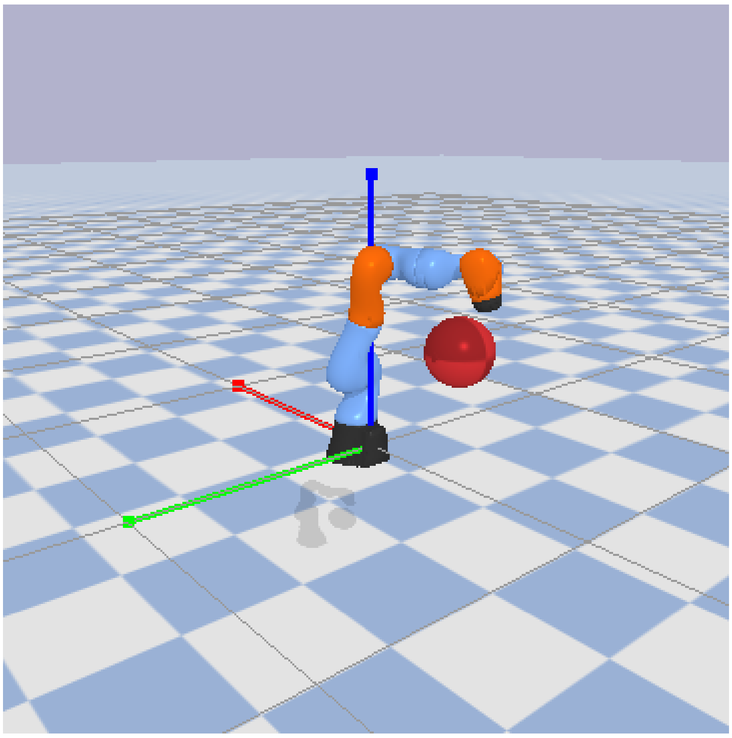


Figure 1: Pybullet environment with robot and obstacle

When performing link collision avoidance an additional repellent is needed to avoid the links of the robot colliding with an obstacle. But instead of incorporating this into the transformation system it is done through inverse kinematics making the end-effector follow the trajectory generated in previous sections while avoiding link collisions. Flow is given as below :

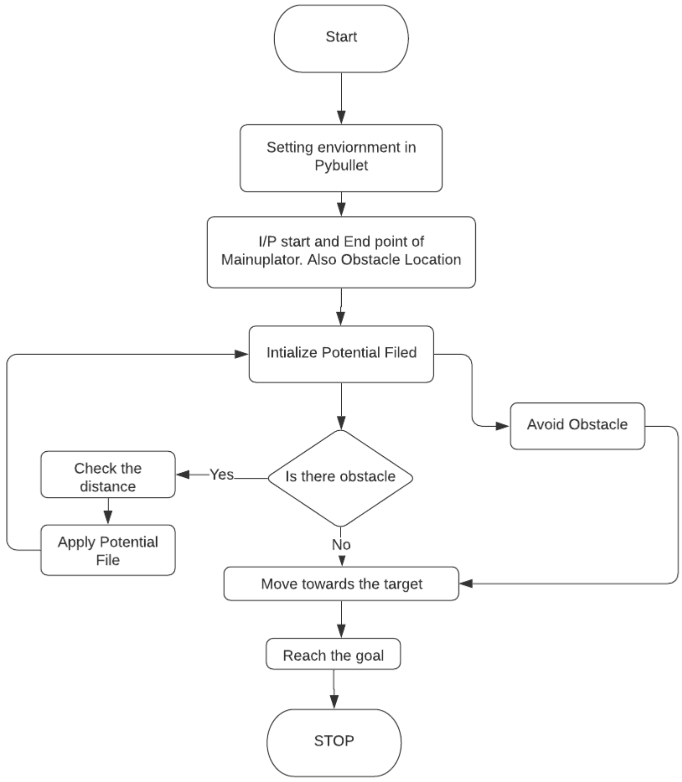


Figure 2: Pipeline or Flowchart of simulation

3.1.1 Null space for redundant manipulator

For 7 DOF all joint axis are perpendicular to the manipulator frame. Where we define Frame 0 as as reference base frame fixed.The end effector velocity is related to the joint velocity through the Jacobian :

$$\dot{x} = J(q)\dot{q} \tag{15}$$

Where $x^{\cdot}{}_{e}$ is the end-effector velocity and Je is the Jacobian of the end-effector. Since the robot is a redundant manipulator the inverse of the Jacobian can not be found, instead

$$\dot{q} = J^{+} x^{\cdot}{}_{e} \tag{16}$$

robot is a redundant manipulator this also means that there are more solutions to it, this can be illustrated by introducing a null space movement:

$$\dot{q} = J^{+} x^{\cdot}{}_{e} + (I - J^{+}J)\varepsilon \tag{17}$$

The null space movement term in (17)

$$(I - J^{+}J)\varepsilon \tag{18}$$

Will map the arbitrary velocity, ε, into the null space This means that we can now find a joint velocity corresponding to a Cartesian velocity and add an arbitrary joint movement that does not affect the Cartesian velocity.

3.1.2 Null Space with obstacle avoidance

The foundation of the utilization of null space movement to avoid obstacles builds on 13, where the null space movement is adapted to avoid an obstacle

To incorporate the obstacle avoidance in the null space movement, the point on the robot $\boldsymbol{x}_0$ closest to the obstacle is considered,

$$\dot{\boldsymbol{x}}_0 = \boldsymbol{J}_0 \dot{\boldsymbol{q}} \tag{19}$$

Where $\boldsymbol{J}_0$ is the Jacobian from the base up to $\boldsymbol{x}_0$. By imposing a constraint on the velocity of the robot closest to the obstacle, $\dot{\boldsymbol{x}}_0$, using a potential field it is possible to make this point, $\boldsymbol{x}_0$, move away from the obstacle. A potential field $U(\boldsymbol{x})$ is now considered :

$$U(\boldsymbol{x}) = \begin{cases} \frac{\eta}{2}\left(\frac{1}{p(\boldsymbol{x})} - \frac{1}{p_0}\right)^2, & \text{if } p(\boldsymbol{x}) < p_0 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The potential field in 20 repels the point $\boldsymbol{x}$ away from the obstacle if it is within a threshold distance p_0. The term $p(\boldsymbol{x})$ is the distance from $\boldsymbol{x}$ to the obstacle. The repulsive effect is implemented by,

$$\dot{\boldsymbol{x}}_0 = -\gamma_L \nabla U(\boldsymbol{x}_0), \tag{21}$$

$$\nabla_{\boldsymbol{x}} U(\boldsymbol{x}) = \eta \left(\frac{1}{p(\boldsymbol{x})} - \frac{1}{p_0}\right)\left(\frac{-1}{p(\boldsymbol{x})^2}\right) \nabla_{\boldsymbol{x}} p(\boldsymbol{x}) \tag{22}$$

Where γ_L is a constant that has to be tuned. The derivative of the potential field is only as stated above when $p(\boldsymbol{x}) < p_0$.

By substituting Equation 17 into Equation 19 the following is obtained:

$$\dot{\boldsymbol{x}}_0 = \boldsymbol{J}_0 \left(\boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e + \left(\boldsymbol{I} - \boldsymbol{J}_e^+ \boldsymbol{J}_e\right) \boldsymbol{\mathcal{E}}\right) \tag{23}$$

Since all entities in the above is known except the arbitrary velocity $\boldsymbol{\mathcal{E}}$, it can be isolated and the null-space velocity is obtained

$$\boldsymbol{\mathcal{E}} = \left(\boldsymbol{J}_0 (I - \boldsymbol{J}_e^+ \boldsymbol{J}_e)\right)^+ (\dot{\boldsymbol{x}}_0 - \boldsymbol{J}_0 \boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e) \tag{24}$$

The expression for the null-space velocity, $\boldsymbol{\mathcal{E}}$, now can be written as :

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e + \left(\boldsymbol{I} - \boldsymbol{J}_e^+ \boldsymbol{J}_e\right) \left(\boldsymbol{J}_0 (I - \boldsymbol{J}_e^+ \boldsymbol{J}_e)\right)^+ (\dot{\boldsymbol{x}}_0 - \boldsymbol{J}_0 \boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e) \tag{25}$$

According to 25 can be simplified to the following,

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e + \left(\boldsymbol{J}_0(I - \boldsymbol{J}_e^+ \boldsymbol{J}_e)\right)^+ (\dot{\boldsymbol{x}}_0 - \boldsymbol{J}_0 \boldsymbol{J}_e^+ \dot{\boldsymbol{x}}_e) \tag{26}$$

Everything is now known and therefore the link collision avoidance can now be implemented.

3.1.3 Implementation of method

To ensure that a solution exists, the start position and the target position of the end-effector are generated in the task space level, but with a few bounds for two reasons:

1. To ensure that bounds on the position of the end-effector on the robot arm are not crossed.
2. To ensure that Robot joints do not lead to singularity or cause its own links to collide.

Throughout the project, the value on x-axis for the end-effector start and target position were fixed to x_x = 0.4. The x_y and x_z values of the end-effector position for the start and target positions were randomly selected within its bounds while testing the algorithm on multiple experiments. Specifically, the submitted code randomly generates start and target positions for the end-effector.Obstacle avoidance is an important task in the field of Robotics. Many approaches have been identified to avoid obstacles. However, every approach in literature since that period has some advantages and disadvantages, which make different algorithms applicable for different obstacle avoidance situations and environments. One such obstacle avoidance algorithm is the Potential field algorithm. Below are results from the simulation:

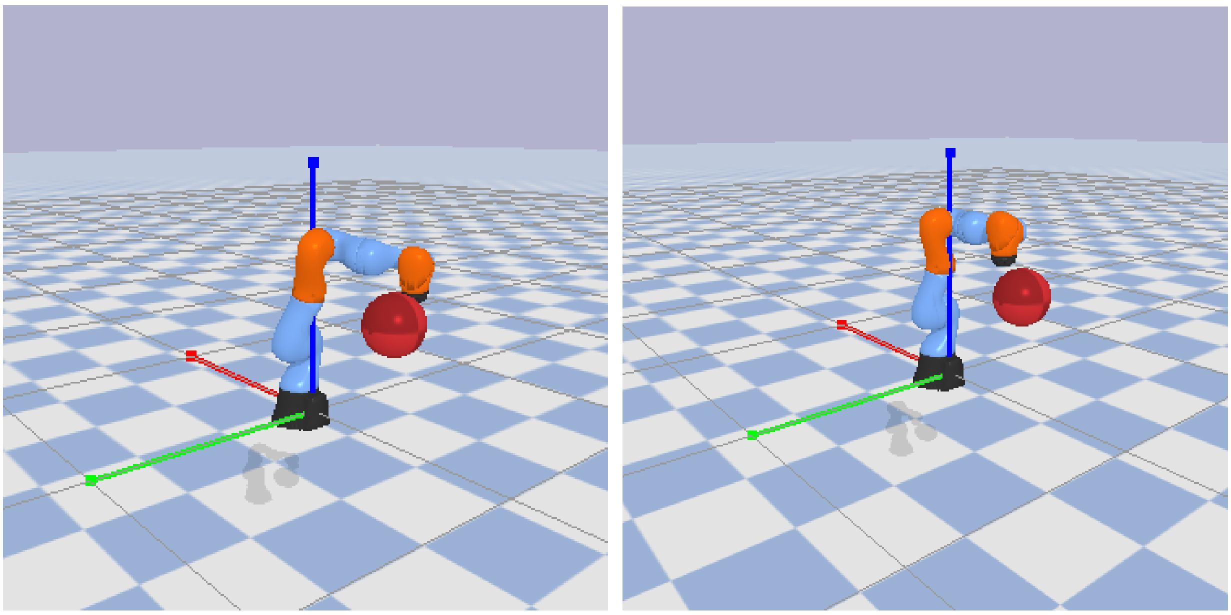
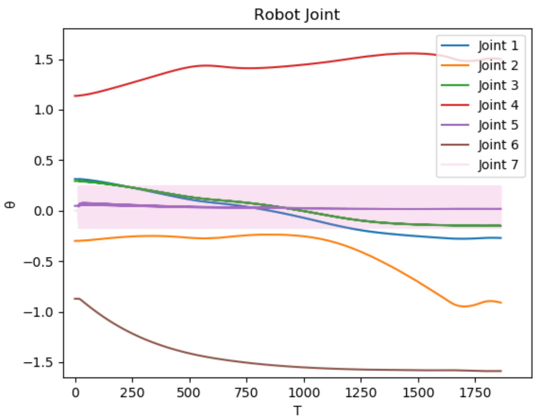


Figure 3 : (a) On the left is when the potential field is 0 and robot goes into collision (b) on the right hand side the potential field is set minimum 0.2 to avoid obstacle robot converges towards the goal

4 Results

The results shown below are for the when the link is in collision with the obstacle and when it's avoiding the obstacle. The various experiment were designed where the parameters such as potential field has been changed to check the different observation which compiling the results.



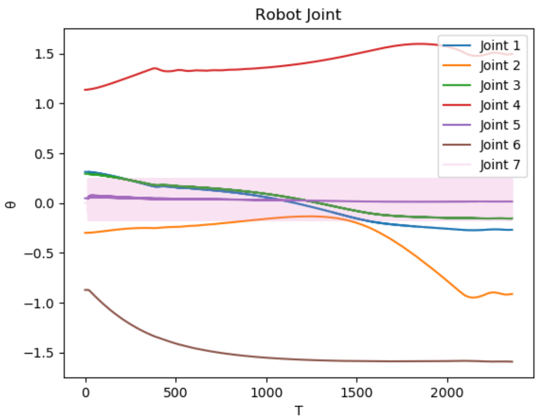
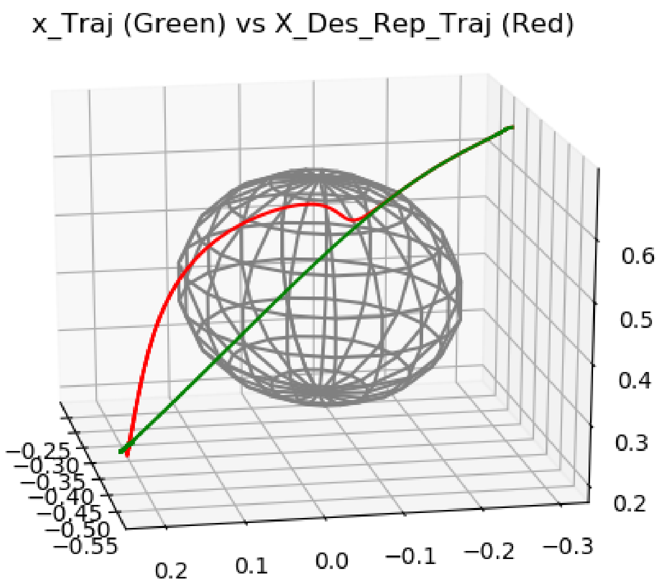


Figure 4 : Joint rotation and time step of motion of robot

In [Figure: 4] it can been seen when the robot the traj of Joint 2 when in collision on left hand side and when avoiding obstacle on right hand side it changes because of the potential filed acting. In the first case of obstacle collision the potential field value of $\rho = 0$ and on the right hand side value of $\rho = 0.2$ for achieving link collison avoidance.



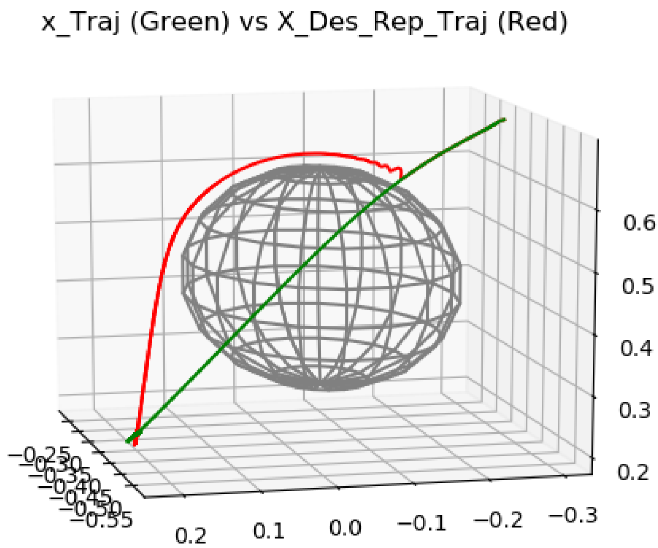


Figure 5 : 3D plot of obstacle avoidance

In [Figure: 5] is the trajectory of end effector when performing the collision or obstacle avoidance. In this the green is the trajectory from the start to goal position and the red is trajectory in presence of potential field . In the above 3D plot it can be seen on the left hand side it's observer in case of if the potential field is 0 then it's in collision with obstacle on the right hand side when the potential field is 0.2 the end effector avoids the obstacle and reach the goal position.

There is the distance to the target with respect to a given point on the arm, if the minimum distance is increase between obstacle and manipulator then output results are :

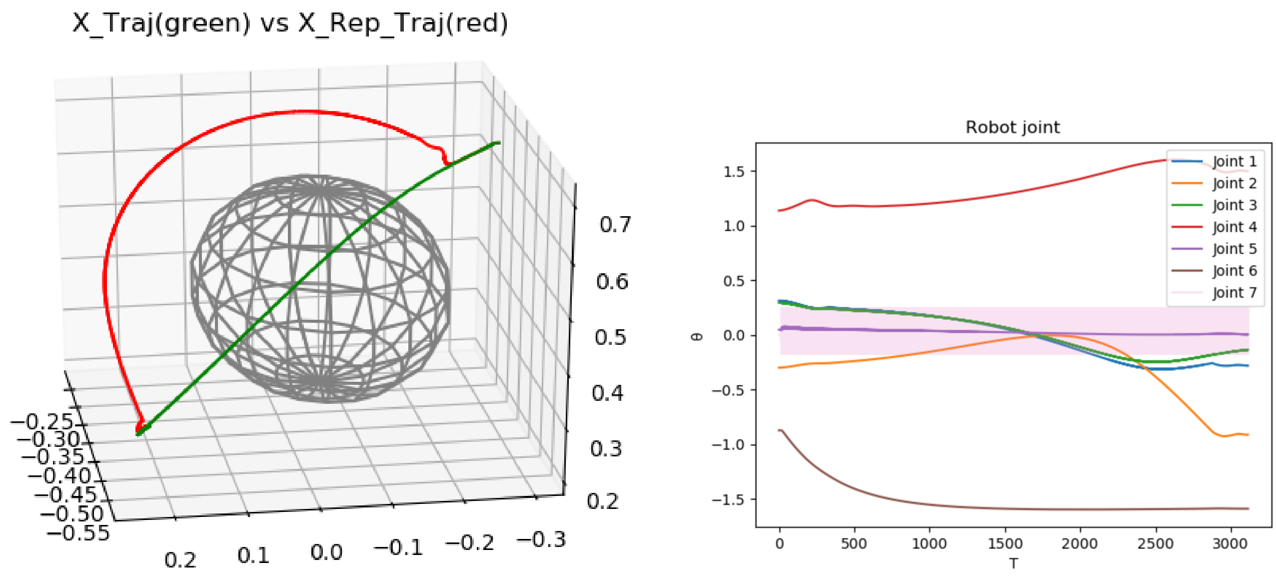


Figure 6 : Plots when minimum distance is increase

5 Conclusion

In conclusion the link collision avoidance using a 7-Dof robot has been implemented in pybullet. It can be played around by changing various parameters such as obstacle position and potential field also.This is clearly visible from every experiment run with Repulsive Potential Algorithm, as in the position when avoiding the obstacle.It's observe the link is able to avoid the collision with obstacle.

References

[1] Khatib, Oussama. *Real-time obstacle avoidance for manipulators and mobile robots." Autonomous robot vehicles. Springer,* . (New York) 1986. 396-404.

[2] Full body obstacle avoidance,
`https://studywolf.wordpress.com/2016/11/24/full-body-obstacle-collision-avoidance/`

[3] Koren, Y., Borenstein,. *Potential field methods and their inherent limitations for mobile robot navigation* . (In ICRA) Vol. 2, pp. 1398-1404.

[4] Z. Z. Yu, J. H. Yan, J. Zhao, Z.F. Chen and Y.H. Zhu,. *Mobile robot path planning based on improved artificial potential field method,*. (Journal of Harbin Institute of Technology,) vol. 43, no. 1, pp. 50–55.

[5] Maria Isabel Ribeiro,. *Obstacle Avoidance with VHFS,*.
`http://users.isr.ist.utl.pt/ mir/pub/ObstacleAvoidance.pdf`

[6] Giuseppe Carbone, Gentiane Venture and Yusuke Sugahara,. *Motion Planning and Control of Redundant Manipulators for Dynamical Obstacle Avoidance*.
`https://www.mdpi.com/2075-1702/9/6/121/htm`

Implementation

- The implementation on the physical robot and the one in simulation can be found at GitHub `https://github.com/r0b0shubham96/Link_Collison_APF`