Path Planning for Quadrotor using Artificial Potential Field Approach to follow Dynamic Target and avoid Static Obstacles

1st Lokesh Bansal
Robert Bosch Center for Cyber Physical Systems
Indian Institute of Science, Bangalore, India
lokeshbansal@iisc.ac.in

2nd Sangeet M
Division of Electrical, Electronics and Computer Sciences
Indian Institute of Science, Bangalore, India
sangeetm@iisc.ac.in

3rd Vaishnav K V
Division of Electrical, Electronics and Computer Sciences
Indian Institute of Science, Bangalore, India
vaishnavk@iisc.ac.in

***Abstract*— The potential field algorithm which was introduced by Khatib in the 1980s is well-known in path planning for robots. The algorithm is very simple yet provides real-time path planning and is effective to avoid robot's collision with obstacles and hence it is a reactive path planning algorithm. It can provide the path in a changing environment. The purpose of the paper is to implement an artificial potential path planner in V-REP.VREP is a 3D robot simulation software with an integrated development environment that allows us to model, edit, program, and simulate any robot or robotic system. We have used MATLAB to do the computational aspects for path planning and VREP is used as a visualization platform. In addition to this, we have applied this algorithm to a six DOF drone whose modeling is done in MATLAB. The simulation result we got indicates the fact that the artificial potential field algorithm is a good choice if we used it as a local planner than a global planner. In many cases, the robot got trapped in local minima, which prevents the robot from reaching the target position. Hence It can be concluded that the artificial potential field algorithm needs to be modified to pass all of the local minima problems.**

Keywords—Quadrotor, path planning, Artificial Potential Field, PD Controller, Dynamic Target Trajectory Introduction

I. Introduction

Path planning is an important primitive for autonomous mobile robots and various algorithms are available to solve the same. Path planning can be defined as finding a path from the current location of the mobile robot to its target location in such a way that it avoids all obstacles that come in optimal or near-optimal paths. It can be viewed as an optimization problem, we have a set of paths from which we need to choose the best one. How one defines the term best here is what helps us to compare different algorithms. We had concentrated on the artificial potential method. The Artificial Potential field approach is a path planning algorithm for moving the robot from the initial to the goal point with the help of potential function. The Artificial Potential field algorithm has been examined by several researchers for various applications such as mobile robots [1] [2] [3], wheelchair [4], underwater vehicles [5] [6], humanoid robots [7], walking robots [8]. It is not a computationally expensive algorithm [9]. The artificial potential field approach comes under reactive planning algorithms [10], which means it can react to the changing environment. In summary, it provides simple and effective motion planners for practical purposes [11] [12] [13] [14].

The paper will be written in the following structure. The first part is the introduction. Next, the second part is about a brief overview of different types of algorithms for Path planning. The third part will discuss the Artificial Potential Field Algorithm. The fourth part contains six DOF models of a quadrotor with Proportional plus Derivative controller implementation to follow the desired trajectory. The fifth part contains the simulation results, where a quadrotor follows a dynamic target while avoiding static obstacles. The last part is the conclusion and future work.

II. Path Planning Algorithms

Path planning is the controller of the robot-motion that determines the path that the robot should follow to reach the destination from the start. The whole environment is divided into the robot, obstacles, and the goal. In path planning, the robot has to implement an algorithm that gives the direction on reaching the destination simultaneously, avoiding obstacles.

Path Planning broadly divides into 2

- Global
- Local

Global path planning usually uses an already known environment where the robot is working, and they are generally more effective in a static environment. Local path planning is closer to real-time applications. They work efficiently in an unknown environment for the robot, including dynamic environments. The robot collects information in real-time and rewrites its control laws to adapt to the changing surroundings, and importantly reach the goal.

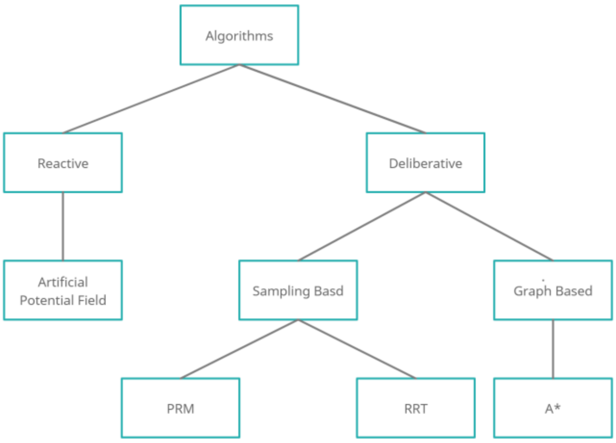

Fig 1. Types of path planners

A. A* Algorithm

It uses a distance with the cost heuristic function of the node to determine the order in which the search visits nodes in the tree. The distance-plus-cost heuristic is a sum of two functions:

i) The path-cost function, cost of starting node to present node

ii) An admissible "heuristic estimate": distance from the present node to the goal

B. RRT Algorithm

Points are randomly generated. The new point is connected to the closest available node. A couple of checks are needed to be carried out for each vertex.

- The vertex must lie outside of an obstacle.
- The connection to the neighbor should not contain any obstacles.

The stopping condition will be generating a valid node in the goal region.

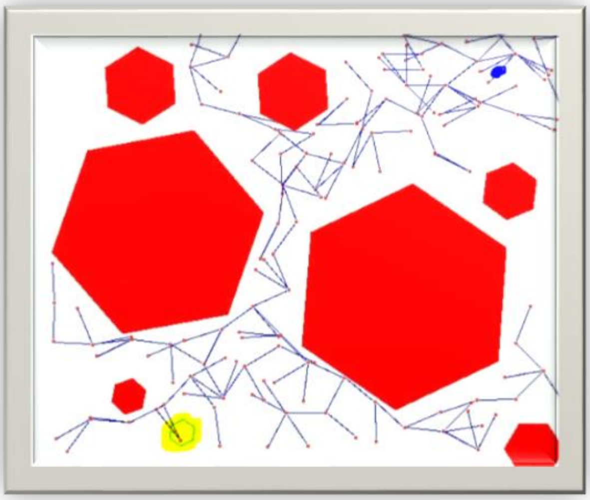

Fig 2. Node creation in RRT

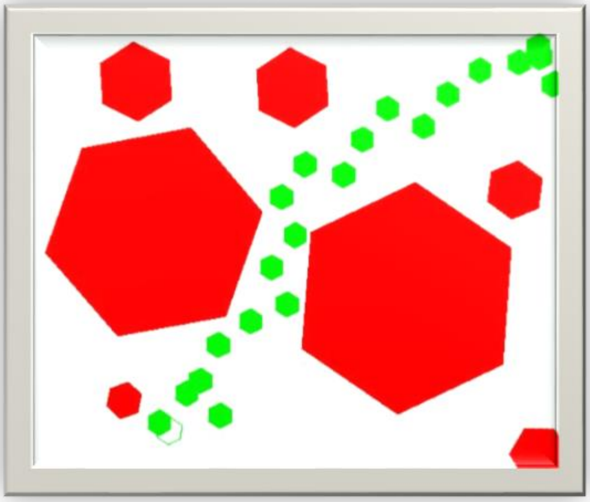

Fig 3. The optimal path to the destination

C. PRM Algorithm

It is very much similar to RRT in execution. PRM generates a limited number of random points within a given area. After the nodes are created, clusters of nodes as connected components are created. Clustering is done on nodes within a given radius. The new node is added to the cluster if the condition is satisfied. A node can, and should, belong to multiple clusters. The characteristics of the PRM algorithm depend on different parameters. And these parameters have to be specified early on, depending on the problem, to get optimal output from the algorithm.

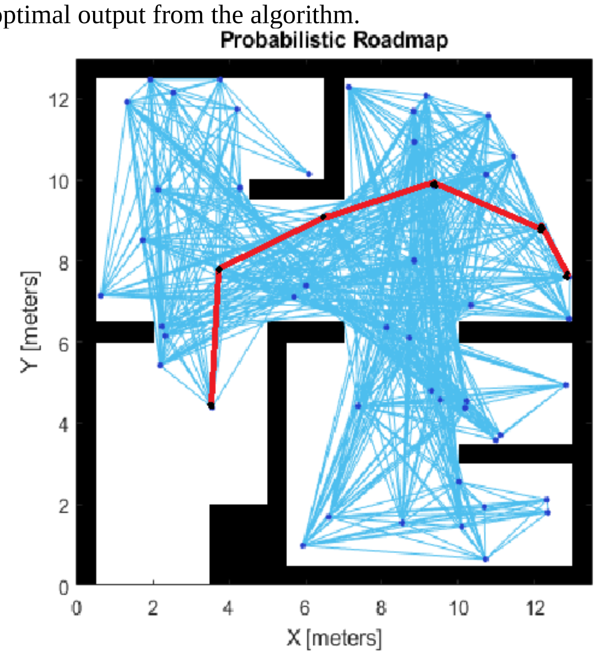

Fig 4. Map created in PRM

D. Comparison of the methods

Algorithm	*PROS*	*CONS*
APF	-Works for a dynamic environment -easy computations	-local minima
A*	-global maximum -suitable for a static environment	-computationally complex -time-consuming
RRT	-less complex -probabilistically compete	-not optimal -practically not complete
PRM	-no local minimum -small calculation	-parameter-dependent -random solution

III. Artificial Potential Field

The artificial potential field approach is a reactive path planning strategy. When the surrounding is changing with respect to time other active path planning algorithms will fail. This reactivity to the changing environment is one of the defining features of the artificial potential fields. This approach will make the path planner handle dynamic obstacles, dynamic targets. The robot is considered as a negatively charged object and the goal is a high positive charged object. We associate each point in the plane with a scalar number called the potential at a particular point, denoted by U(x,y).considering this potential as the energy, we need a scalar distribution in such a way that will make the robot reach the low energy configuration. At every point, the robot will choose the best possible direction which ensures the maximum decrease in the function value.

A. *Optimization perspective*

The AFP approach is very similar to the gradient descent algorithm. Assuming the potential function is differentiable At each point we choose a descent direction. The step size can be chosen if required. The artificial potential field is analogous to the flow of rainwater. Ideally, we want it to reach the ocean (here is the goal) and at each point, water flows along the direction of the maximum decreasing slope. Sometimes water gets trapped without reaching the ocean, this problem is analogous to the local minima problem which is discussed later in the report. So clearly Artificial potential field is a nature-inspired algorithm.

B. *Attractive potential function*

We can employ conical potential function to model attractive potential function. In the conical potential function, the potential increases as the distance grow hence a constant pulling force towards the goal irrespective of the distance. Note that the derivative of the function is not defined at the goal, Hence we should introduce another smooth function to rectify this problem, hence we choose quadratic potential function when the robot is near to the goal. In total, we have a piecewise function and suitable arrangements are done to ensure continuity at the transition point.

$$U_{au}(q) = \begin{cases} \frac{1}{2}\xi d^2\left(q, q_{\text{goal}}\right) & \text{if } d\left(q, q_{\text{gool}}\right) \le d^*_{\text{goal}} \\ d^*_{\text{goal}}\,\xi d\left(q, q_{\text{goal}}\right) - \frac{1}{2}\xi\left(d^*_{\text{goal}}\right)^2 & \text{if } d\left(q, q_{\text{goal}}\right) > d^*_{\text{goal}} \end{cases}$$

C. *Repulsive potential function*

If the robot is far away, there shouldn't be any interaction, and as the robot becomes closer to the obstacle repulsion must happen. The model should be able to capture the above behavior. Ideally, the obstacle should be modeled as an infinite potential well but we chose inverse quadratic potential function. Every obstacle has a specific limited region that has a repulsive field so that when the robot comes in that region, it will be repelled from that obstacle, Here it is denoted by the symbol Q*.

$$U_{rep_i}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{D_i(q)} - \frac{1}{Q^*}\right)^2 & \text{if} \quad D_i(q) \le Q^* \\ 0 & \text{if} \quad D_i(q) > Q^* \end{cases}$$

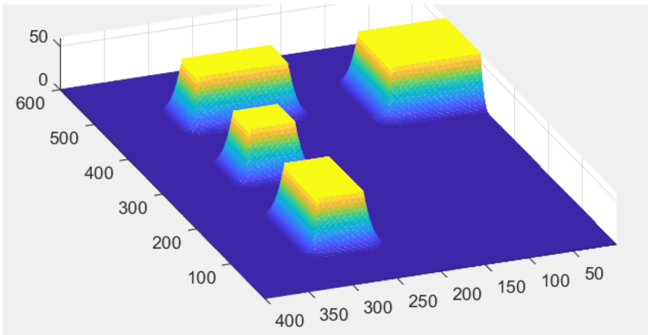

Fig 5. Repulsive potential

In summary, we have a goal that pulls the robot towards itself, governed by the attractive potential function. There can be multiple obstacles and each of their respective repulsive potential function is added to get the net repulsive potential function. We add the repulsive potential function and the attractive potential function to yield the total potential function.

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

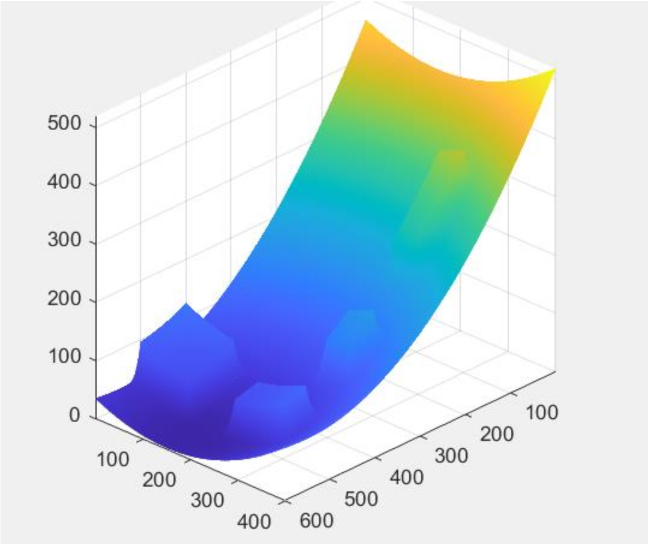

Fig 6. Total potential

D. *Local minima problem*

The robot should reach the coordinate where the total potential function attains the global minimum value, But the artificial potential field approach can only guarantee to reach a point where the gradient vanishes. If the potential functions are nicely chosen then this approach will reach a local minimum, It may or may not be a global minimum. In short, we want a global minimum but we may get trapped in one of the local minima. This is one of the disadvantages of the artificial potential field approach. A method to get rid of this problem is to find local minimums explicitly, but this is a computationally expensive task. We can't afford that since the artificial potential field is a reactive path planning strategy.

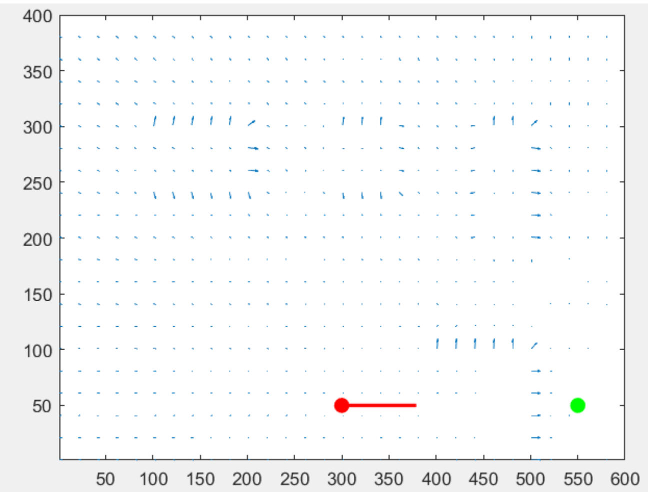

Fig 7. Local minima

IV. Modeling

The total project can be modularized into 3 major parts, which are individually modeled and then merged using interface programs. This binding program acts as an intermodular communication channel and helps in instruction and data exchange between them.

A. *Artificial Potential Field (APF)*

Often continuous distance computation is not practical hence discretization of distance is done. We divide the 2-D space into grid space and associate a boolean value to each index. True if there is an obstacle is present False if there is no obstacle. For each of these obstacles, there is a repulsive potential function associated with them now add them up. But when it comes to the implementation part this can easily be done using bwdist function available in MATLAB. Other than summing up the individual functions, using the bwdist function we do this in a single step and generate the repulse potential matrix. Similarly, we can get the repulsive potential matrix. But this time we use a simplified model for the easiness of implementation. This is important because on each iteration we need to recompute these matrices. We add these matrices to get the total potential matrix. We use the function gradient which is available in MATLAB, which will compute the discrete equivalent of gradient from the total potential matrix. Using this obtained direction compute the next point and now repeat the process until the gradient becomes zero. This is how the artificial potential field path planner is implemented in this project.

B. *Quadrotor Model and PD Controller[15]*

Quadrotor: In the model, the body reference frame B (b_1, b_2, b_3) is attached with the quadrotor with origin at the Centre of Mass (C) of the quadrotor, and the inertial frame A (a_1, a_2, a_3) is taken at a reference point with O as origin. b1 is considered in forward direction, b2 in lateral direction, and b3 in vertical upward direction. The four rotors are attached at a distance L from C along positive and negative b1 and b2.

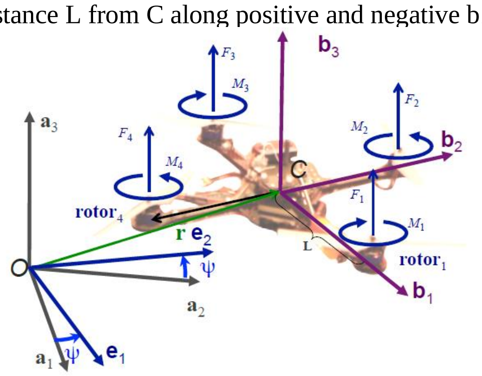

Fig 8. Quadrotor model with the body fixed and the inertial frames

Forces and Moments: Considering angular velocity of the rotors to be ω_i then Forces and Moments can be expressed as

$$F_i = k_F \omega_i^2$$

Where $k_F \approx 6.11 \times 10^{-8} \frac{\text{N}}{\text{rpm}^2}$

$$M_i = k_M \omega_i^2$$

Where $k_M \approx 1.5 \times 10^{-9} \frac{\text{Nm}}{\text{rpm}^2}$

Equation of Motion of Quadrotor:

Considering the Euler angles: ϕ, θ, ψ roll, pitch, and yaw respectively and taking their derivatives to get roll rate, pitch rate, and yaw rate. Now, to get the angular velocities in the body frame expressed as p, q, r along b_1, b_2, b_3 respectively,

$$\mathcal{A}_{\omega_{\mathcal{B}}} = p\boldsymbol{b}_1 + q\boldsymbol{b}_2 + r\boldsymbol{b}_3$$

p, q, r can be obtained using,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Further, the rotors will generate forces and moments that will be considered as inputs (u_1, u_2) to the system. Where u_1 and u_2 can be expressed as,

$$u_1 = \Sigma_{i=1}^{4} F_i$$

Where F1, F2, F3, and F4 are forces generated by the four rotors. Using Newton's Law, governing equations of motion of center of mass (C) for translation can be expressed as,

$$m\ddot{\boldsymbol{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

Where r is the position of C with respect to the origin in the inertial frame and R is rotation matrix:

$$A[R]_{\mathcal{B}} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

And,

$$u_2 = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

Where L is the distance of arm of the quadrotor and $\gamma = \frac{k_M}{k_F}$

Where $k_F \approx 6.11 \times 10^{-8} \frac{\text{N}}{\text{rpm}^2}$

Where $k_M \approx 1.5 \times 10^{-9} \frac{\text{Nm}}{\text{rpm}^2}$

Using Newton's Law, governing equations of motion of center of mass (C) for rotation can be expressed as,

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

PD Controller

To implement a proportional plus derivative (PD) controller, linearize the state-space model of governing equations of quadrotor around by taking hover state as equilibrium conditions,

$r = r_0, \theta = \phi = 0, \psi = \psi_0, \dot{r} = 0$, and $\phi = \dot{\theta} = \psi = 0, c\phi \approx 1, c\theta \approx 1, s\phi \approx \phi, s\theta \approx \theta, F_{i,0} = \frac{mg}{4}$

.

Linearized equation of motions can be expressed as,

$$\ddot{r}_1 = g(\Delta\theta \cos\,\psi_0 + \Delta\phi \sin\,\psi_0)$$
$$\ddot{r}_2 = g(\Delta\theta \sin\,\psi_0 - \Delta\phi \cos\,\psi_0)$$
$$\ddot{r}_3 = \frac{1}{m} u_1 - g$$
$$\dot{p} = \frac{u_{2,x}}{I_{xx}} = \frac{L}{I_{xx}}(F_2 - F_4)$$
$$\dot{q} = \frac{u_{2,y}}{I_{yy}} = \frac{L}{I_{yy}}(F_3 - F_1)$$
$$\dot{r} = \frac{u_{2,z}}{I_{zz}} = \frac{\gamma}{I_{zz}}(F_1 - F_2 + F_3 - F_4)$$

Where $I_{xx} = I_{yy}$ by assuming the quadrotor to be symmetric.

PD Controller is used to make the quadrotor to follow the desired trajectory by giving inputs u_1 and u_2 that will minimize the error between current position-orientation parameters and desired position-orientation parameters.

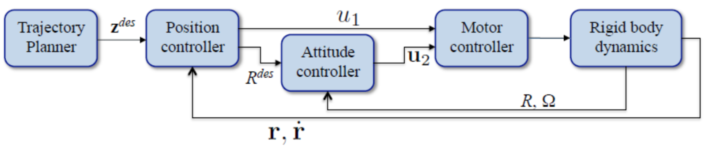

Fig 9. The Position and Attitude Controller Loops

The objective is to follow the desired trajectory, the desired trajectory is obtained by joining a set of waypoints obtained from the artificial potential field algorithm. This set of waypoints are joined in such a manner that a smooth trajectory with minimum jerk is obtained, this is the minimization of jerk problem is solved by the method of Calculus of Variation. To follow the desired trajectory the PD controller will give inputs u_1 and u_2. PD Controller will work in two sections: Position Controller and Attitude Controller.

Position Controller: To obtain the input u_1 PD controller minimizes the error by working with a linearized set of equations for translation.

$$u_1 = mg + m\ddot{r}_{3,\text{des}} = mg - m\left(k_{d,3}\dot{r}_3 + k_{p,3}(r_3 - r_{3,0})\right)$$

Attitude Controller: To obtain the input u_2 PD controller minimizes the error by working with a linearized set of equations for rotation.

$$\boldsymbol{u}_2 = \begin{bmatrix} k_{p,\phi}(\phi_{\text{des}} - \phi) + k_{d,\phi}(p_{\text{des}} - p) \\ k_{p,\theta}(\theta_{\text{des}} - \theta) + k_{d,\theta}(q_{\text{des}} - q) \\ k_{p,\psi}(\psi_{\text{des}} - \psi) + k_{d,\psi}(r_{\text{des}} - r) \end{bmatrix}$$

Now, to follow the desired trajectory, take a vector $z_{\text{des}} = \begin{bmatrix} r_T(t) \\ \psi_T(t) \end{bmatrix}$, where $r_T(t)$ is provided by the artificial potential field function and $\psi_T(t)$ can be taken as constant ψ_0.

Now, to drive the PD controller, position and velocity error functions are needed,

$$e_p = \left((r_T - r)\cdot \hat{n}\right)\hat{n} + \left((r_T - r)\cdot \hat{b}\right)\hat{b}$$

$$e_v = \dot{r}_T - \dot{r}$$

Where r_T is nearest position on desired trajectory from the current position r, $\hat{b}$ is unit binormal vector such that $\hat{b} = \hat{t}\times\hat{n}$, where $\hat{t}$ and $\hat{n}$ are unit tangential and unit normal vectors to the trajectory. Now command acceleration is obtained by following relation

$$(\ddot{r}_T - \ddot{r}_{des}) + k_d e_v + k_p e_p = 0$$

Further, to obtain ϕ_{des} and θ_{des} position control algorithm can be used with following equations,

$$\phi_{des} = \frac{1}{g}\left(\ddot{r}_{1,des}\sin\psi_0 - \ddot{r}_{2,des}\cos\psi_0\right)$$

$$\theta_{des} = \frac{1}{g}\left(\ddot{r}_{1,des}\cos\psi_0 + \ddot{r}_{2,des}\sin\psi_0\right)$$

The desired roll and pitch rates are taken as zero,

$$p_{des} = 0$$
$$q_{des} = 0$$

C. V-REP

Virtual Robot Experimentation Platform (V-REP) is a robotic simulator. that is used for the experimentation in this work. It is open-source software and it has a direct link with MATLAB. Its script can be written as a MATLAB script. It can be linked to MATLAB as a remote API.

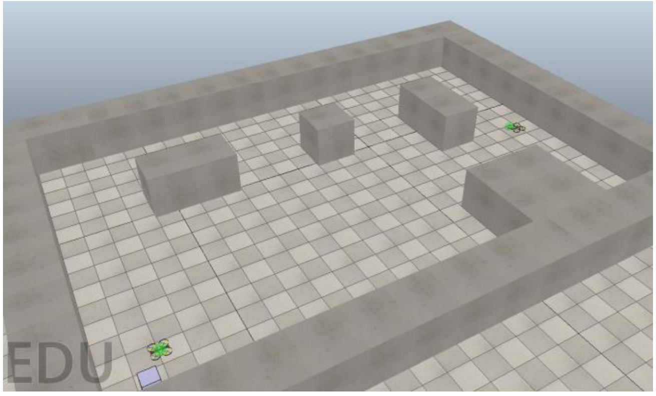

Fig 10. Drone in the environment (V-REP)

The Quadcopter model in V-REP is used to simulate and observe how the copter is working with the APF code implemented in MATLAB. The calculation parts, which are the steps to find the trajectory, are done in MATLAB, reducing the computations in V-REP and the respective instruction to the drone is passed on to V-REP using API functions

V. RESULT

The program can be divided into two. MATLAB implementation, which is the brain of the model and V-REP for modeling and executing the algorithms.

In V-REP, the environment for the simulation of the drone was set up. A cube is placed at (1,1) to indicate the starting position of the source. There are obstacles of different sizes. The source robot which uses the APF method to traverse is a drone. The target robot is mobile and can continuously move in real-time. The source drone will use the APF function to get the set of points to traverse to reach the target, and this step is executed in each iteration, which will enable it to fetch the information about the varying position of the target.

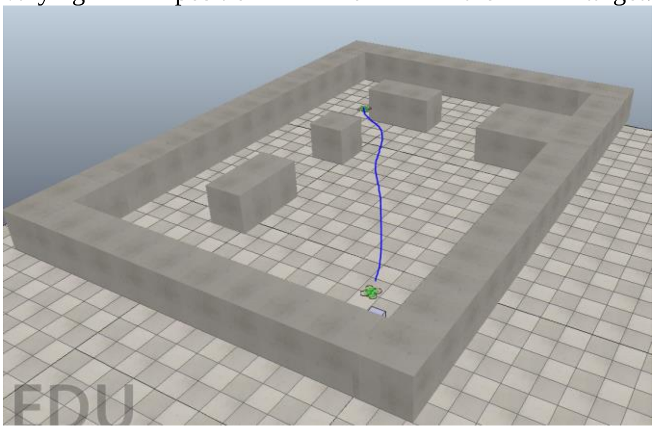

Fig 11. Drone implementation in V-REP

MATLAB scripts are of 2 parts. In the first part, the APF algorithm, which runs as an abstract method, was implemented, in which the environment skeleton is hardcoded. The environment is scaled to 50 times as it executes on integer values. Thus the communication between MATLAB and V-REP contains a scaling of 50. The APF function accepts the present and final target coordinates of the drone and the target, respectively, then calculates and returns the next step, which is to be taken by the source drone in order to reach the destination. The algorithm ensures the next jump it proposed is free from obstacles and also avoids any possibility of collision of the drone with the obstacles. The execution time of the APF function is under 1 second. At certain times the algorithm can fall into local minima, but as it is part of the APF algorithm, it means the method is properly executing.

In the second part, the control program executed the same APF function and displayed the trajectory followed by the drone from the start of the program.

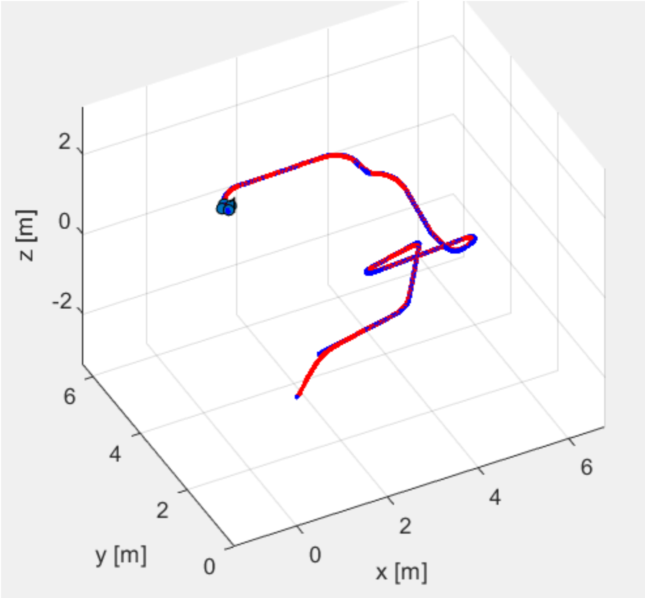

Fig 12. a) Trajectory followed by the drone

The drone moves to the next jump that the program fetches from the APF function in every iteration. The steps are executed iteratively until the distance between the source drone and the destination point is lesser than a fixed threshold.

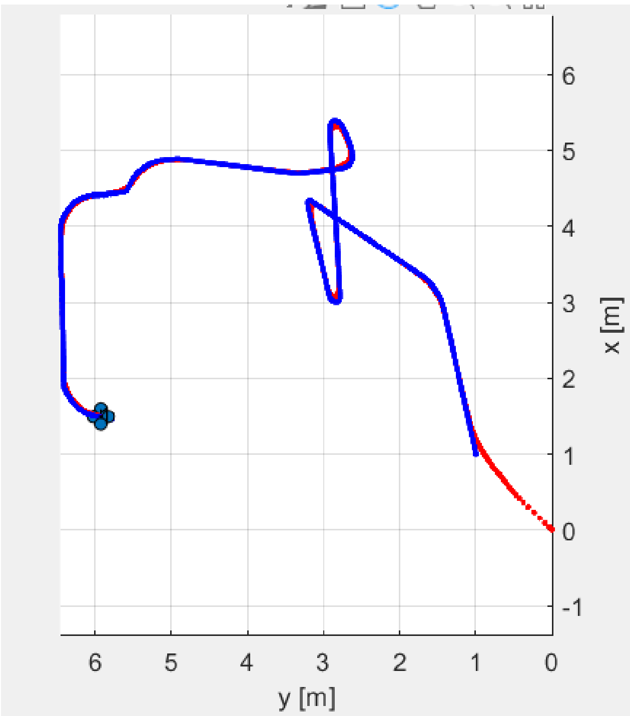

Fig 12. b) Trajectory followed by the drone

VI. Conclusion

Artificial Potential Field (APF) algorithm provides good results to follow a dynamic target and avoid static obstacles. The major problem observed is that the quadrotor gets trapped in local minima when attraction and repulsion forces are balanced. Because of the local minima problem, the APF approach can not be deployed to generate a global trajectory. Since it is very good in dynamic environments, it can be deployed as a local path planner responsible for avoiding static and dynamic obstacles while following and dynamic target.

VII. Future Work

A hybrid algorithm can be developed consisting of a global and a local path planner, where the APF algorithm can be integrated with deliberative techniques like A*, RRT etc. Further, the algorithm can be deployed for the autonomous landing of a quadrotor on a moving platform. A control strategy can be developed to land the drone on a moving platform while the drone would follow the moving landing platform using the APF algorithm.

VIII. Acknowledgment

We would like to thank Prof. Shishir N.Y. Kolathaya for giving us this opportunity to work on this problem of Path Planning for a Quadrotor and giving us the much-needed advice and suggestions whenever we were stuck in the project, without which this project would not have been possible

.

IX. References

[1] Triharminto, H. H., Wahyunggoro, O., Adji, T. B., & Cahyadi, A. I.,"An Integrated Artificial Potential Field Path Planning with Kinematic Control for Nonholonomic Mobile Robot," International Journal on Advanced Science, Engineering and Information Technology, vol. 6,no. 4, pp. 410-418, 2016.

[2] [2] Triharminto, H. H., Wahyunggoro, O., Adji, T. B., Cahyadi, A., &Ardiyanto, I., "Local Information using Stereo Camera in ArtificialPotential Field based Path Planning," IAENG International Journal of Computer Science, vol. 44, no. 3, 2017.

[3] [3] O. Montiel, U. Orozco-Rosas, and R. Sep´ulveda, "Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles," Expert Syst. Appl., vol. 42, no. 12, pp. 5177-5191, Jul. 2015.

[4] [4] Onyango, S. O., Hamam, Y., Djouani, K., Daachi, B., & Steyn, N., "A driving behaviour model of electrical wheelchair users," Computational intelligence and neuroscience, 2016.

[5] [5] G. Yun, W. Zhiqiang, G. Feixiang, Y. Bo, and J. Xiaopeng, "Dynamic Path Planning for Underwater Vehicles Based on Modified Artificial Potential Field Method," in 2013 Fourth International Conference on Digital Manufacturing & Automation, 2013, pp. 518-521.

[6] [6] S. F. J., R. A. F., V. A. Sebasti´an, and A. G. G., "Artificial potential fields for the obstacles avoidance system of an AUV using a mechanical scanning sonar," in 2016 3rd IEEE/OES South American International Symposium on Oceanic Engineering (SAISOE), 2016, pp. 1-6.

[7] [7] Q. Zhong, J. Zhao, and C. Tong, "Tracking for humanoid robot based on Kinect," in 2014 International Conference on Mechatronics and Control(ICMC), 2014, pp. 1191–1194.

[8] [8] H. Igarashi and M. Kakikura, "Path and posture planning for walking robots by artificial potential field method," in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04.2004, 2004, pp. 2165-2170 Vol.3.

[9] [9] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh "A review: On path planning strategies for navigation of mobile robot," Defence Technology, vol. 15, no. 4, pp. 582–606, Aug. 2019

[10] [10] S. Wang, T. Zhao, and W. Li, "Mobile Robot Path Planning Based on Improved Artificial Potential Field Method," in 2018 IEEE International Conference of Intelligent Robotic and Control

[11] Engineering, IRCE 2018, 2018, pp. 257–262.

[12] [11] X. Lin, Z. Q. Wang, and X. Y. Chen, “Path Planning with Improved Artificial Potential Field Method Based on Decision Tree,” in 27th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2020 - Proceedings, 2020.

[13] [12] Y. Liu and Y. Zhao, “A virtual-waypoint based artificial potential field method for UAV path planning,” in CGNCC 2016 - 2016 IEE Chinese Guidance, Navigation and Control Conference, 2017, pp. 949–953.

[14] [13] E. Wu, Y. Sun, J. Huang, C. Zhang, and Z. Li, “Multi UAV Cluster Control Method Based on Virtual Core in Improved Artificial Potential Field,” IEEE Access, vol. 8, pp. 131647–131661, 2020.

[15] [14] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, “Obstacle Avoidance Path Planning Design for Autonomous Driving Vehicles Based on an Improved Artificial Potential Field Algorithm,” Energies 2019, Vol. 12, Page 2342, vol. 12, no. 12, p. 2342, Jun. 2019.

[16] [15] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP Multiple Micro-UAV Testbed," IEEE Robotics and Automation Magazine, 2010