IJCTA, 9(33), 2016, pp. 71-78

Comparative Study of Potential Field and Sampling Algorithms for Manipulator Obstacle Avoidance

Alekh V* Rahul ES and Rao R. Bhavani*****

Abstract : Industrial robots should be implemented with proper path planning to generate a trajectory which ensures obstacle avoidance and collision free movement. Selection of a path planning techniques need to be done only after detailed studies. This paper deals with the study and implementation of a path planning technique using potential field method for a 2R robot in a given workspace with obstacles. The results are compared with Open Motion Planning Library (OMPL)which uses sampling based path planning algorithms in a similar context. It is found that in higher dimensionality spaces sampling algorithms are more useful and potential field methods are handy in real time applications.

Keywords : Path planning, potential field method, sampling algorithms, ROS, OMPL, manipulator.

1. INTRODUCTION

The development in the field of automation and flexible manufacturing systems has resulted in the increased use of industrial mobile robots and manipulators. Industrial robot application includes painting, welding, pick and place, palletizing, packaging, product inspection, testing and assembly line productions. Various industrial tasks can be accomplished with precision, speed and endurance using these robots.

Industrial robotic systems operate in real environment, which in most of the cases consists of obstacles that can be static or dynamic. In order to complete the given task successfully, the robot manipulator must find the objects and avoid obstacles, create a collision free path and follow the trajectory. The user aims at achieving optimal solutions, taking in account of time, energy and other factors, satisfying the applied constraints at the same time.

Human safety in robot's work environment can be factored by ensuring collision free task completion for the robot. Such algorithms can make the robot intelligent in the presence of humans or other obstacles. Safety and efficiency are the most highlighted factors while defining and establishing a shared workspace[1].

Various methodologies have been developed and implemented for the path planning of mobile robots. Collision avoidance schemes must be extended from end-effector level to link level in case of manipulators [2]. For manipulators, the problem becomes complex due to the structure. Understanding the strengths and weaknesses of existing algorithms forms a key factor in the implementation phase. Hybrid algorithms are sometimes formulated incorporating the best features and switching the algorithms

* Amrita School of Engineering, Amritapuri, alekhvelayudhan@gmail.com.

** Amrita Vishwa Vidyapeetham. rahules@am.amrita.edu.

*** Amrita University, Kerala, India, bavani@amrita.edu.

according to the scenario. In this paper, a comparison is done between potential field method and sampling algorithms through study and implementation of the two approaches. For this purpose, the potential field method is implemented on a 2R serial robot and simulation results are obtained in MATLAB. Sampling based algorithms are implemented by using OMPL(Open Motion Planning Library) in MoveIt! to achieve path planning in the presence of obstacles and visualized in Rviz plugin of ROS.

The paper proceeds as: Section 2, gives an idea about the related work in the domain for the two approaches. Section 3, provides an insight to the methodologies that are to be compared. Section 4 lays down the simulation results. Section 5 discusses the results of the work. Section 6 gives the conclusion and proposes future work.

2. RELATED WORK

Path planning has been an important field of research for the last few decades. Research is being carried out on soft computing techniques to control and perform the path planning for manipulators and mobile robots [3].Various heuristic based algorithms have been implemented and compared with deterministic methods [4].

Potential field based obstacle avoidance for manipulators and mobile robots was introduced by Khatib [5]. Even though the method is widely used, it suffers limitations such as local minima problems. Advanced methods were later developed to reduce the short comings of the potential field method [6]. The method was extended form 2-D to 3-Dspace. Later work also includes strategies that avoid obstacle for the whole manipulator [7]. Similar work was done [8] for solving the local minima problem and evaluating the technique. The problem becomes complex when the degrees of freedom increases. A collision free algorithm was developed for high DOF manipulator by selecting the appropriate configuration for the respective potential [9]. A global path planning strategy is used and followed by local planning for the entire robot. The general idea of potential field method was implemented using various numerical techniques [10] in which graph based search for local minima, forms the criteria.

Frameworks are necessary to write software for robots. An overview of ROS, detailing the basic architecture, concepts and components was given in [11]. The basic principles and best practices to reduce the complexity of using the framework is a major concern. Case studies were done on MoveIt! [12], for improving usability.

3. METHODOLOGY OVERVIEW

The problem of path planning for industrial manipulators has been approached in two different ways in this paper. The first approach is by applying the potential field method to achieve collision free motion for the manipulator. The second method uses ROS with default motion planners from OMPL. The obstacle is assumed to be static and its position is known.

3.1. Potential Field Method

Potential field method is one of the classical methods used in the field of path planning mainly in real time applications. The method is formulated in such a way that the robot under consideration is under the influence of a potential field. The total potential is always the sum of attractive and repulsive potential. The negative gradient of the field yields forces, one of which is attractive and the other repulsive.

$$\mathrm{F}(q) = -\nabla \mathrm{U}_a(q) - \nabla \mathrm{U}_r(q) \tag{1}$$

where $\mathrm{U}(q)$ represents the potential field applied, $\mathrm{F}(q)$ is the resultant force,

$\mathrm{U}_a(q)$ defines the attractive field and $\mathrm{U}_r(q)$ gives the repulsive field.

The attractive force pulls the robot towards the goal and repulsive force repels the robot away from the obstacle. The attractive field is a combination of conic and quadratic potentials where the quadratic potential is a continuous function at every point including the origin.

$$\mathrm{U}_a(q) = \begin{cases} \frac{1}{2}k_1\rho^2(q), & \rho(q) \le d \\ dk_1\rho(q) - \frac{1}{2}k_1 d^2(q), & \rho(q) > d \end{cases} \tag{2}$$

A parameter k_1 is used to tune the attractive potential field, $\rho(q)$ is the Euclidean distance between q and q_{final}, where q is the set of all points from start point to final point in the path, $d = \rho(q)$ is the boundary that separates the two potentials.

The repulsive force field is responsible for preventing the robot from colliding with the obstacles. Repulsion takes place in such a way that infinite force is exerted on the robot when it gets near to obstacles. Also, when the robot is at a considerable amount of distance from the obstacle, the repulsive force acting is taken as zero. A boundary should be defined to establish the region subjected to the force originating from the obstacle.

$$\mathrm{U}_r(q) = \begin{cases} \frac{1}{2}k_2\left(\frac{1}{\rho_s(q)} - \frac{1}{\rho_0}\right)\rho_s^2 & \rho_s(q) \le \rho_0 \\ 0, & \rho_s(q) > \rho_0 \end{cases} \tag{3}$$

where k_2 is the tuning parameter for the repulsive field, ρ_0 sets the distance that determines the effect of obstacle, ρ_s gives the shortest distance between current robot position and obstacle, which needs to be updated at times.

A gradient descent approach is followed, considering the scenario as an optimization problem. The planning starts from the initial configuration, the direction of negative value of the gradient is identified and a small step is taken. This results in attaining a new defined configuration. The operation is repeated so that the robot finds the final configuration. The step size selected should be small so that robot-obstacle collision is avoided and larger value increase the time for computation. For making the implementation less difficult, often configuration potential field is reduced to workspace potential. The force developed in the workspace is to be mapped for the robot's motion. For manipulator, the method is to applied to links by selecting a set of control points. Torques are induced on joints as the robot arm is subjected to the force field. The required mapping is governed by the relationship between applied force to the robot arm and induced torque on the robot joints is[13]:

$$\mathrm{J}^{\mathrm{T}} = \tau \tag{4}$$

where τ is the vector for joint torques, F is the force vector and J is the Jacobian for mapping. The robot configuration is at singularity, when inverse of J does not exist. Implementation and simulation are discussed in section 4.

3.2. Robot Operating System (ROS)

ROS is a framework to write software for robots and provides various user functions. It contains algorithms, libraries and tools for different types of robotic applications. One can access these packages and modify it accordingly for their specific application. ROS being multilingual, the user can write the software in any language provided that, a client library should be written for the particular language.

MoveIt! of ROS allows the user to develop applications involving mainly robot manipulation and motion planning. Robot designs can be tested and validated using the workspace analysis tool. The primary node made available by MoveIt! is named move_group. One can use MoveIt! through move_group_interface in C++, Python or GUI based Rviz plugin.

OMPL (Open Motion Planning Library) uses sampling based algorithms for motion planning. Sampling based algorithms works on the basic idea that the configuration space is sampled and the configurations associated with free space are remembered. A roadmap is then created which connects the configurations lying in free space. Another set of sampling algorithms exists, where incremental search methods are used instead of building maps. The planners can be of multi-query or single query solvers. When several number of queries related to same system are to be addressed, multi-query methods are used and single-query methods solve specific queries. Pre-processing phase is required for multi-query methods where it takes most of the computational time. Single-query methods are much faster but outputs are less accurate for further use. These planners can be effectively applied when the degrees of freedom are higher.

OMPL can be incorporated with other systems to get additional functionalities such as state validation, collision checking. The motion planners in the library are set as default planners while using MoveIt!. In OMPL, state space denotes the configurations on the generated path. Workspace is used to represent a space where the robot performs the assigned task. The planners ensure a collision free path by linking the elements of free state space. The information related to the space where the path plan is to be done is contained in the Space Information part of the library. The user must create objects that are required for the planning procedure. The class named SimpleSetup can be used for making the procedure easier. It also makes the components of the setup available for further modifications. Planners in the library can be switched as per the user choice. New planners can also be added to the library. OMPL provides the capability to process the path output. The function, simplifySolution is used to simplify and make the path smoother. It works by removing unwanted states and vertices from the obtained path . MoveIt! along with OMPL allows the user to develop and test new algorithms.

Algorithms developed in MATLAB can be used in ROS through a toolkit, Robotics Systems Tool box. The interface allows the user to test robotic applications in ROS supported simulators. It also enables the user to create ROS nodes that can be used for communication in a ROS network.

4. SIMULATION

Simulations are carried out in two different platforms: MATLAB for potential field method and ROS Indigo for OMPL. Potential field method is implemented in MATLAB and simulation results are obtained. Fig.1, shows the field generated where the obstacle boundary is modeled as peaks and the goal position as valley. A 2R robot is subjected for the method with its end-effector considered as a point. The end-effector is under the influence of the attractive and repulsive forces depending upon its proximity to the goal and obstacle respectively. The method yields a collision free path connecting start point and the final point. Intermediate points are selected in the resultant path and cubic spline trajectory is generated between consecutive intermediate points. Fig.2, shows the path generated for 2R robot by implementation of potential field path planning.

Intermediate points are selected in the resultant path and cubic spline trajectory is generated between consecutive intermediate points. The polynomials are solved satisfying the boundary conditions for each segmented trajectory. Fig.3, shows the plots for position, velocity and acceleration after trajectory generation using cubic splines.

Rviz plugin of ROS is used to visualize the motion planning with OMPL. ABB IRB 2400 is selected for the simulation test which is of 6 DOF. The motion planner is applied on the links 3 and 4.The URDF (Unified Robot Description Format) of the robot is taken as input to the MoveIt! Setup Assistant where it is configured. Rviz motion planning interface is then launched with the configured ABB IRB 2400 SRDF (Semantic Robot Description Format). The user can configure the plugin as per the requirement.

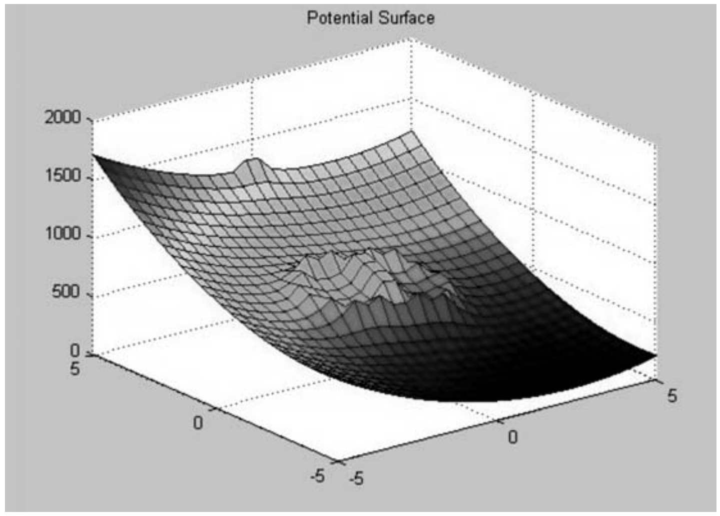

Figure 1: Potential field generated

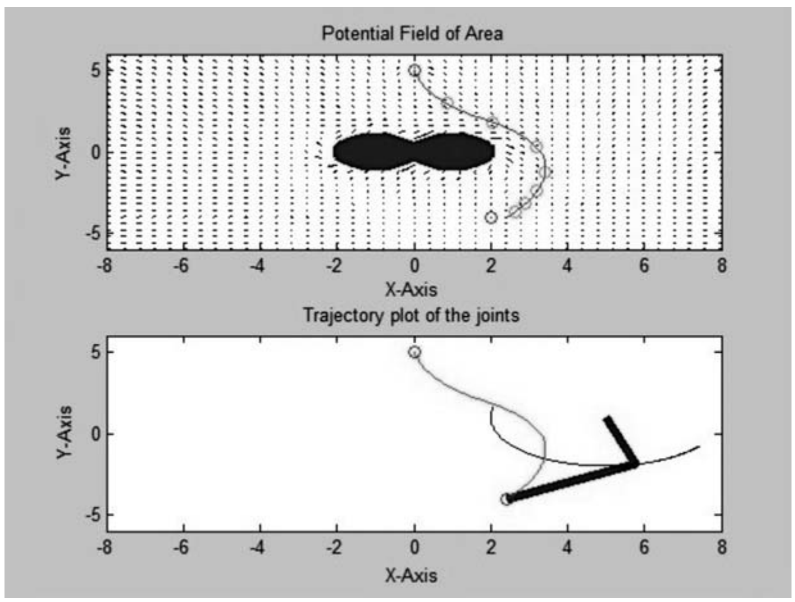

Figure 2: Collision free path generated by implementing potential field method

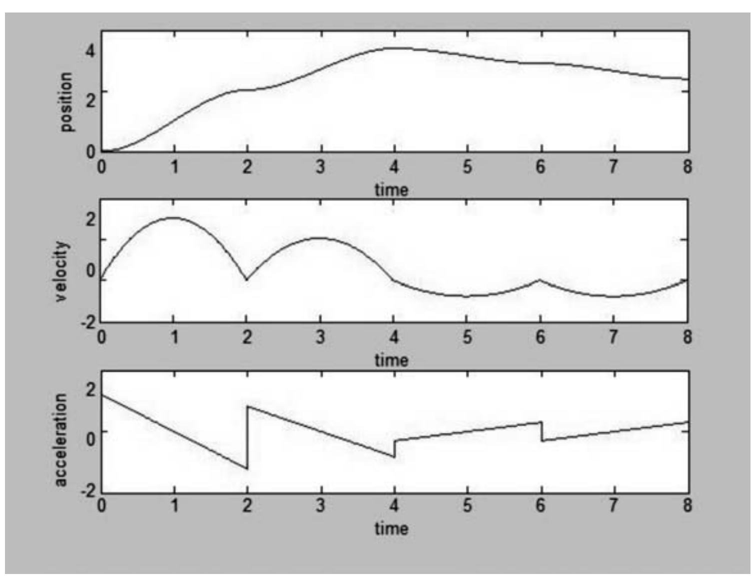

Figure 3: Position, velocity and acceleration plots after trajectory generation through via points

Interactive markers are provide in the visualizer through which the user can interact with the robot. The goal state and the end state for the robot can also set by using these markers.

Next step is the addition of known objects to the environment which can be done through the scene objects. Scene object is created using text file and imported in the visualizer. Any change in planning environment must be updated and the current planning scene should be published. Motion planning is done by setting the planning library as OMPL. The start state and the goal state are given using interactive markers to get a collision-free motion of the manipulator completing the trajectory. Fig. 4, shows the GUI for motion planning in ROS where Rviz is used as the plugin.

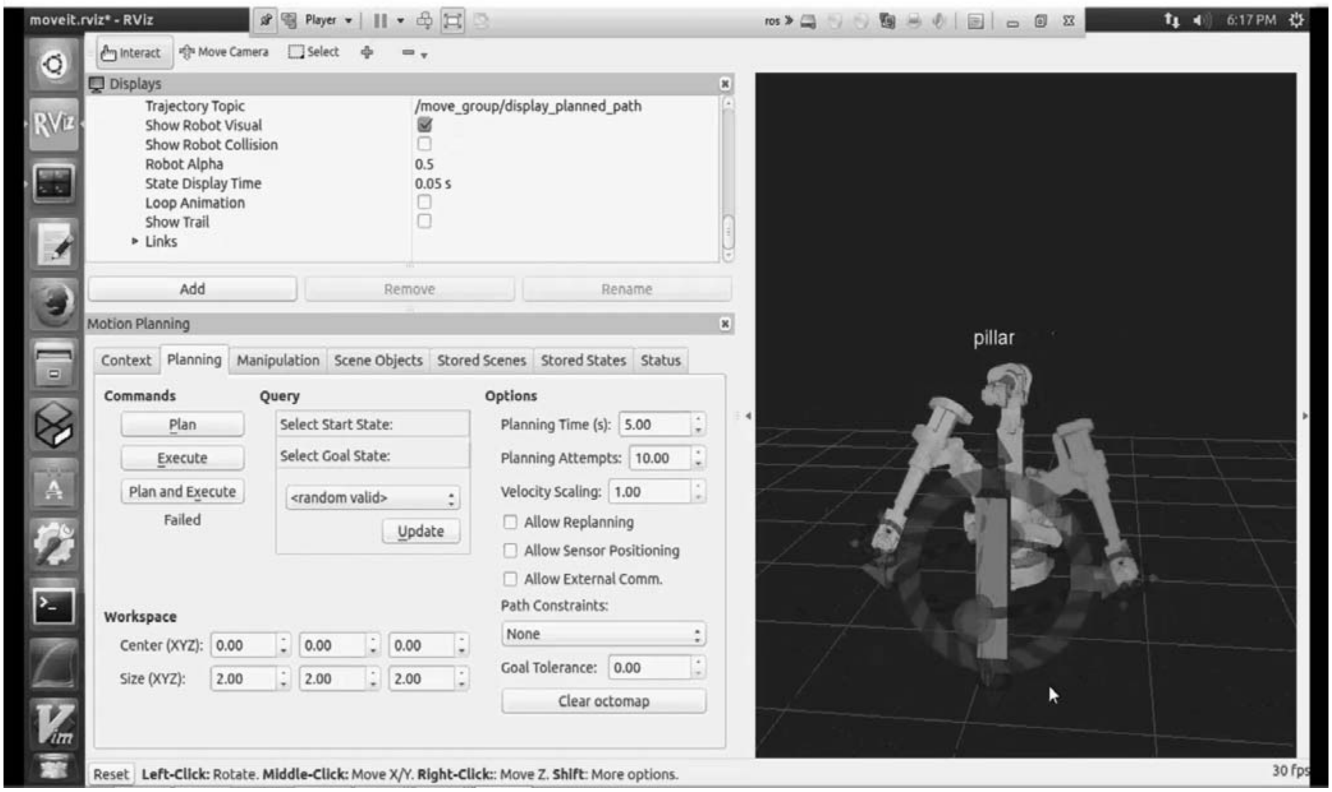

Figure 4: Rviz plugin used for motion planning in ROS

5. RESULTS AND DISCUSSION

The real time performance of potential field method is commendable which gives an upper hand over high level planning. The method was found to be failing in case of local minima points. Another problem related to the algorithm is unstable vibration of the robot. There exist chances that the robot will not reach the target, instead gets stuck in local minima especially when the obstacles are in a cluttered manner and are of concave shaped. Since the robot is always under the constant influence of attractive as well as repulsive forces, it is found to be subjected to unstable vibrations. The planner was implemented with start point, final point and obstacle location as inputs. From the simulation results, it is observed that path planning using potential field method took 3.12 seconds without obstacle and 2.34 seconds with obstacle.

Sampling based algorithms finds the connectivity between the collision-free areas of configuration space. The sampling based algorithms in OMPL fail to identify whether there exists a solution for given set of constraints. In such situations, the user has to set runtime for the planner. Approximate solutions are given as output when the planner finds no solution within the defined runtime. These solutions can be approved for less dimensional state spaces, but not acceptable for higher cases.

Online path planning can also be done with OMPL but encountered difficulties. The planner has to be re-initialized at times, where there is change in planning scene. OMPL sets all the states as valid while solving the problem, which is not recommended. The user must therefore, define the criteria that makes the state valid. For simulation, start point, final point, obstacle and its location used were same as that in potential field implementation. Using OMPL, path planning was achieved in 1.17 seconds with planning scene having no obstacle whereas in the presence of obstacle, the process took 1.32 seconds.

6. CONCLUSIONS AND FUTURE WORK

A study on the potential field path planning for obstacle avoidance was done. The method was implemented in MATLAB for 2R robot where the output is a collision free path. The path thus generated was segmented with via points and formulated using cubic splines.

Obstacle avoidance with trajectory following was simulated in the Rviz plugin of ROS. Motion planning was carried on two links of the ABB IRB 2400 URDF in MoveIt! using OMPL as the planning library. Collision checking is completed with a lighter version of FCL (Free Collision Library) provided with OMPL. The process require continuous publishing of the planning scene to update the simulation environment. The manipulator follows a collision free path but the simulator shows uncertain path patterns when the obstacle is too close to the manipulator. The results of both the methodologies are studied and compared.

We would like to extend the study on path planning strategy for robots with higher degrees of freedom, by implementing the potential field method through ROS. Velocity control of the manipulator in the presence of obstacle is also considered for future work.

7. ACKNOWLEDGMENT

We are deeply indebted to Mata Amritanandamayi Devi, Chancellor of Amrita University and world renowned humanitarian whose humanitarian work around the world has been a source of inspiration and strength to us. We would like to express our gratitude to Akshay Nagarajan, for providing technical and moral support, our mentor Gayathri Manikutty, for supporting and motivating us throughout the process, Prof. K. Kurien Issac and Asst. Prof. S.Vishnu Rajendran for their valuable feedbacks at various stages of the work.

8. REFERENCES

1. Hayes B, Scassellati B. **Challenges in shared-environment human-robot collaboration *learning***. 2013; 8,9.
2. Choset HM. Principles of robot motion: theory, algorithms and implementation. MIT press, 2005.
3. Watanabe K, Izumi K. A survey of robotic control systems constructed by using evolutionary computations *Systems, Man, and Cybernetics, 1999.IEEE SMC'99 Conference Proceedings*, 1999, 758–763.
4. Ferguson D, Likhachev M, Stentz A. A guide to heuristic-based path planning. *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS*), 2005, 9–18.
5. Khatib O. **Real-time obstacle avoidance for manipulators and mobile robots**. ***The international journal of robotics research***. 1986; 5 (1),90–98.
6. Koren Y, Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation. *Robotics and Automation Proceedings, IEEE International Conference*, 1991, 1398–1404.
7. Liu S, Zhang Q, Zhou D**. Obstacle avoidance path planning of space manipulator based on improved artificial potential field method**. ***Journal of The Institution of Engineers (India): Series C***. 2014; 95(1), 31–39.

8. Tang L, Dian S, Gu G, Zhou K, Wang S, Feng X. **A novel potential field method for obstacle avoidance and path planning of mobile robot**. ***Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference***, IEEE, 2010; 9, 633–637.

9. Ranjbar B, Mahmoodi J, Karbasi H,Dashti G, Omidvar A. **Robot manipulator path planning based on intelligent multi-resolution potential field**. ***International Journal of u-and e-Service, Science and Technology***. 2015; 8(1), 11–26.

10. Barraquand J, Langlois B, Latombe JC. **Numerical potential field techniques for robot path planning**. ***Systems, Man and Cybernetics, IEEE Transactions.*** 1992; 22 (2), 224–241.

11. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, N g A Y. **Ros: an open-source robot operating system**. ***ICRA workshop on open source software***. 2009;3 (3.2),5.

12. Coleman D, Sucan I, Chitta S, Correll N. **Reducing the barrier to entry of complex robotic software: a moveit! Case study**. ***Journal of Software Engineering for Robotics***. 2014.

13. Jazar RN. Theory of applied robotics: kinematics, dynamics, and control. Springer Science & Business Media, 2010.