An analytical approach for decoupling the orientation planner and the Cartesian position planner for the PR2 end effector

Gokul R. Subramanian
gosub@seas.upenn.edu

University of Pennsylvania

Dr. Maxim Likhachev
maximl@seas.upenn.edu

***Abstract* — This paper discusses an analytical approach for decoupling the orientation planner from the Cartesian position planner for the PR2 end effector. The main goals are improvement in performance and robustness without loss of completeness. To accomplish this, we employ basic 3 dimensional spatial geometric formulations. We show how this method has advantages over traditional planning methods. Solution using our method takes time in the order of microseconds. We believe that these results can be generalized for application to other robot systems as well, especially in the field of robotic object manipulation.**

***Keywords* — PR2 robot, Willow Garage, Motion Planning, Mobile Manipulation, End Effector, Yaw, Pitch, Roll (Euler Angles), Forward Kinematics, Inverse Kinematics, ROS, Gazebo.**

I. INTRODUCTION

Motion planning has been a central theme in robotics research for around three decades, and has existed for even longer. The main problem in motion planning could be described as having to be able to generate a vector of configuration states going from the start state to the desired end state, under pre-determined or real time constraints. Although no best solution exists in the general case, a variety of basic algorithms [1] have been developed that provide guidelines for choosing the *best* algorithm for the purpose at hand.

Mobile manipulation is one of the subfields of robotics which heavily relies upon motion planning. The problem is non-trivial because it involves interaction of a robot with its environment, which is often dynamic and difficult to predict and model accurately. This problem is partially solved by equipping the robot with highly sophisticated manipulation devices often in the shape of a human arm. The robot arm is articulated and is capable of performing complicated maneuvers. This helps to abstract the problem of implementation from motion planning for manipulation.

The PR2 (Personal Robot 2) robot has been developed at Willow Garage [2]. The PR2 is an open and robust platform designed for software developers, so as to eliminate the need to design hardware every time a new code is written. Our main focus will be on the two 7 DOF (Degree Of Freedom) arms of the robot. The PR2 robot is fully integrated with ROS (Robot Operating System) [3], which provides a convenient interface for writing code for robots. Figure 1 shows the PR2 robot.

Figure 1 - The PR2 robot, courtesy Willow Garage

This paper is organized as follows. Section II consists of the background material necessary for being able to fully appreciate the content of this paper. Section III intuitively develops the motivation for the work in this paper. Section IV, which is the heart of this paper discusses the theoretical formulation and the specific details of our suggested method for orientation planning. Section V presents some valuable experimental results in Gazebo, an inbuilt robot simulator in ROS.

II. BACKGROUND

Two papers [4] and [5] on mobile manipulation are the basis for our work. The paper on search based planning using improvised heuristic calculations, and a lattice based ARA* planner using motion primitives is a representative example of how mobile manipulation algorithms are implemented on robots. Many other algorithms exist for object manipulation but the main idea behind all algorithms is the same and is briefly outlined here.

The robot arm is the manipulator. Its end effector is often capable of at least grasping objects in its neighborhood i.e. workspace. The first step in manipulation is for the robot to compute a *smart* trajectory for its arm in order to get to the vicinity of the object being manipulated. The next step is of course the manipulation of that object. It is in the first step that motion planning is involved and wherein, different algorithms exhibit different performance characteristics such as speed of execution, computational memory requirements and optimality of returned path for the arm.

Search based algorithms involve constructing a graph of states beginning at the start state and going all the way till the goal (desired) state, and then choosing an optimal path in this graph based on the cost of state transitions. Often, these algorithms provide bounds on how suboptimal a returned path can be, but are not as fast as sample based planners. Sample based planners take samples in the planning space thus reduce the size of the constructed graph and the time consumed in evaluating a path. However, the solutions returned might be highly suboptimal in most cases. Specifically in case of robots, these paths might be too jerky and might be hard for the robot arm to navigate.

Our proposed algorithm is meant to work in conjunction with a planner such as the ones described above or any other for that matter. Our method mainly focuses on planning for orientation of the end effector. This is a very important task because in manipulating objects, the end effector is constrained to take a particular orientation as required by the disposition and shape of the object being manipulated.

Motion planning for the PR2 end effector is a computationally intense process because of the high dimensionality of the problem. There are two general paradigms that one could adopt in solving this problem. Either, the planning could be done in Cartesian space i.e. the 3D world around us or in the *joint angle space*. While planning in Cartesian space, all obstacles are considered as they are, whereas when planning in joint angle space, all obstacles are converted into their equivalent representations in the coordinate space spanned by the joint angles. Clearly for the 7 DOF PR2 robot arm, the joint angle space is 7-dimensional and hence planning in the full joint space is not necessarily the best way to compute plans. Planning in Cartesian space, on the other hand seems more intuitively obvious. The Cartesian space consists of only 6 dimensions how much ever be the dimensionality of the robot arm. Planning in the Cartesian space also allows us to think of end effector position separately from that for its orientation, which is a concept that forms the crux of this paper.

The SBPL (Search Based Planning Library) [6] is a motion planning library implemented in ROS. The package implements a generic set of motion planners developed by the co-author at the University of Pennsylvania in collaboration with Willow Garage. Our innovative solution for orientation planning for the PR2 end effector has been written and tested in conjunction with the SBPL. However, this does not limit the application of our method to any one type of planner. It must be noted that this method may not work for other robot arms which have fewer DOF at the end effector or which have a totally different joint angle description than the one for the PR2 robot arm. The following section presents the motivation for this paper.

III. MOTIVATION

The PR2 robot has two 7 DOF arms, each of which is fully actuated and capable of performing a wide variety of tasks. The 7 DOF are as follows – shoulder pan, shoulder lift, shoulder roll, forearm flex (FF), forearm roll (FR), wrist flex (WF) and wrist roll (WR). Additional DOF are present in the gripper, but those will not be of much concern in this paper.

If motion planning for manipulation were to be done purely by traditional methods discussed above, then the motion of the arm would restricted to a limited set of atomic motions called motion primitives. Each motion primitive would specify a change in the 7 joint angles, for example {0,0,5,0,5,0,0} is a motion primitive specifying a 5 unit change in shoulder roll and a 5 unit change in FR, and no change in the other joint angles. This atomization of motion is done so as to make the planning graph sparse and hence computation less intensive. This method of planning is called lattice based planning.

Other methods such as sample based planning are also very intensive on the whole because although the actual planning time is less, the returned path is highly suboptimal and often jerky, and thus requires special smoothing techniques which consume a lot of processing power. This again shows that attempting to solve the planning problem as a whole using one simple algorithm is not the best way to compute plans.

Clearly any planning procedure for all 7 DOF of the PR2 robot arm would be very uneconomical as far as time and resources are concerned. Thus, it becomes necessary to mentally break this problem into smaller problems, each of which can be solved independently. The question however, is whether such a breakdown exists. In this paper, we provide an innovative solution for implementing this breakdown, or in other words *decoupling*. Specifically, we intend to decouple the two noticeably separate sub-problems of position planning and orientation planning.

This concept of decoupling is immediately clear if we look at the way humans interact with the environment. When grasping objects or manipulating them, humans do not patiently compute plans or smoothen out jerky trajectories from the so called start configuration to the desired configuration. They solve the problem of planning in a more straightforward and natural way by means of visual servo. If we carefully observe the way humans plan for object manipulation, we can infer that they deal with the problem in two separate steps, excluding the grasping and manipulation of the object. In naturally computing a path from the present state of the arm to the desired state, priority is first given to positioning the arm and the hand correctly with respect to the object of concern. It is only after appropriate positioning of the hand that the orientation of the hand is altered.

Since robotics derives significant thrust from the methods followed by human beings, in this paper, we suggest a modification, which is inspired by the concepts explained in the previous paragraph to the standard planning algorithms for object manipulation. Our method is very attractive to implement due to its relatively less computational requirement and ease of incorporation into existing algorithms without much effort. The following section deals with the method in full mathematical rigor and explains our algorithm step by step.

IV. ALGORITHM

We assume in this entire discussion that a main planning algorithm is available, to which our algorithm can be appended. We have used the SBPL as explained in the Background section. Specifically, we have used the ARA* planner [7]. This planner has been chosen because it guarantees plans within specified time limits, which is often an important constraint in manipulation.

The problem description is as follows.

Given an (x, y, z) location and a (yaw, pitch, roll) orientation for the end effector in the global frame of the robot, find a feasible path that the robot arm can follow to reach the desired state from its present state, under given constraints such as time, obstacle avoidance and optimality bounds.

We assume that the main planner is per se capable of solving the planning problem, but it is not necessary for this requirement to be strictly met. In any case whatsoever, the planner must be able to get the robot end effector to the right (x, y, z) location in the planning phase by itself. It is at this point in the planning phase that our algorithm is designed to be invoked. Our algorithm is capable of reorienting the end effector, using purely analytical means, to any pose within the joint limits of the wrist without altering the (x, y, z) position of the end effector.

At first sight, this might sound trivial but that is not the case because of the yaw constraint at the wrist of the PR2 robot i.e. the PR2 robot does not have yaw capability at the wrist. Thus, the question of whether the end effector might able to reach all orientations without changing the (x, y, z) location does not have an obvious answer. The first part of our algorithm, however, involves the proof of the fact that this reorientation in place is possible for the PR2 end effector.

We begin with the proof. First we state the assumptions under which our proof holds. The robot end effector must have continuous roll about the end effector axis, and the forearm must have continuous roll capability about its length. Any joint limits on the wrist flex will only limit the workspace of the end effector but not the applicability of our algorithm. The wrist flex is assumed however, to be about an axis perpendicular to the forearm roll axis i.e. unless the forearm rotates about its roll axis, the wrist will flex about the same axis.

The first step in the proof is straightforward. If ***F*** denotes the frame of reference attached to the forearm frame, then any (yaw, pitch, roll) requirement in the global frame ***G*** can be converted to corresponding requirements in the frame ***F***. All further discussion will refer to the frame ***F***. If the frame attached to the end effector be labeled ***E***, then the frame ***E*** is completely specified by the (yaw, pitch, roll) in the frame ***F***. The relation between the three Euler angles and the frame ***E*** is bijective. So if a set of three successive rotations can always map to the (yaw, pitch, roll) one-one or many-one, then the three rotations may be used to orient the end effector to any desired pose, within of course the joint limits. We propose these three angles to be (forearm roll, wrist flex, wrist roll). The reason why (yaw, pitch, roll) cannot themselves be used is that there is no yaw at the wrist of the PR2 end effector and hence yaw does not map onto any joint angle directly.

The second part of the proof is even more straightforward. Here, we prove that the three angles specified above namely (forearm roll, wrist flex, wrist roll) are necessary and sufficient to reach any possible end effector pose without changing (x, y, z) of the end effector. Firstly, a forearm roll of 2π can sweep through a circle depending upon the wrist flex. Since the forearm roll is assumed continuous, all points along the circle can be reached. Further, by varying the wrist flex, different circles can be traced by the end effector as shown in figure 2.

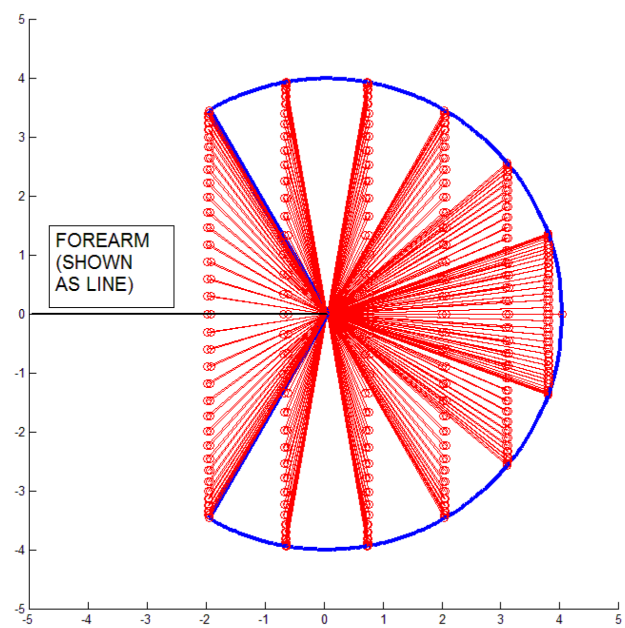

Figure 2 – Forearm roll and wrist flex, showing end effector workspace in forearm frame F

The envelope of all the circles is a sphere centered at the wrist and limited only by the wrist flex joint limit, which can be seen in figure 2. This proves sufficiency of the three angles chosen for reorientation. Proof of necessity can be reasoned out as follows. Forearm roll is necessary to take the end effector off plane (see figure 2). Wrist flex is necessary to reorient in the given plane and wrist roll in necessary to pose the end effector frame ***E*** into any roll configuration.

Thus, we have proved that a series of successive rotations namely the (forearm roll, wrist flex, wrist roll) can be effectively substituted for pure (yaw, pitch, roll) without sacrificing the capability to attain any end effector pose, in position. In case of the PR2 robot, this substitution is a requirement because as already stated, there is no yaw at the wrist. In the PR2 end effector, the wrist flex joint has limits of around 120^0 on one side and around 10^0 on the other. Thus the traversable workspace is a sphere centered at the wrist, with a radius equal to the length of the end effector, but from which a cone of half angle 60^0 is cut out because of the wrist flex joint limits. Now, since the basic idea of end effector orientation is laid down clearly, we present our algorithm for orientation planning.

Step 1:

Obtain desired (x, y, z) and (yaw, pitch, roll) for the end effector frame ***E*** in the global frame.

Step2:

Set intermediate goals starting from a *nearby* (x_temp, y_temp, z_temp) to the final (x, y, z), all with the same end effector pose.

Step 3:

Call the main planner (SBPL in our case) to plan up to the next goal position. Once *close enough* to the goal location, invoke our orientation planner which returns the three rotations described in order to bring the end effector to the desired orientation.

Step 4:

Once that path is determined collision free, execute the action i.e. let the robot arm traverse the path. Then return to step 3. If the path has unavoidable collisions, then discard the temporary goal and move onto the next goal i.e. return to step 3. If none of the goals is attainable, then the vicinity of the chosen goals is too cluttered for object manipulation. Go to step 2 and reselect goals.

The following section discusses some of the experiments conducted in Gazebo, an inbuilt simulator in ROS. These experiments confirm the potential of our method and clearly demonstrate that decoupling orientation planning from position planning has many advantages as mentioned earlier.

V. EXPERIMENTAL RESULTS

Our orientation planner is akin to a typical inverse kinematics program except that it is much faster and easier to manage. The algorithm is analytical and purely procedural involving no loop constructs or numerical methods. Thus, the runtimes are exceptionally low considering the complexity of the PR2 robot arm. Figure 3 shows the average runtime for our planner.

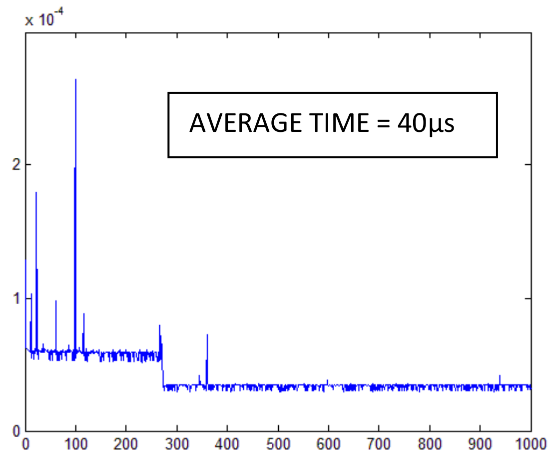

Figure 3 – Runtime, orientation planner, 1000 runs

Our orientation planner was first tested with the PR2 model in Gazebo simulated free space. The test results were satisfactory because given a desired (x, y, z) and (yaw, pitch, roll), the planner was able to achieve the goal for most cases. This was confirmed using Rviz, a visualization aid for Gazebo. Figure 4 (next page) shows an example of how the PR2 arm attained the desired pose of (yaw = 0, pitch = 0, roll = 0), the most common configuration required for grasping objects.

To test our algorithm, the setup shown in figure 5 was adopted. The planner is able to get to the vicinity of the object to be manipulated, in this case the stick placed on the table. The only step to be completed then is the actuation of the gripper at the right time and manipulation of the stick. This is proof of the fact that our algorithm works as promised.

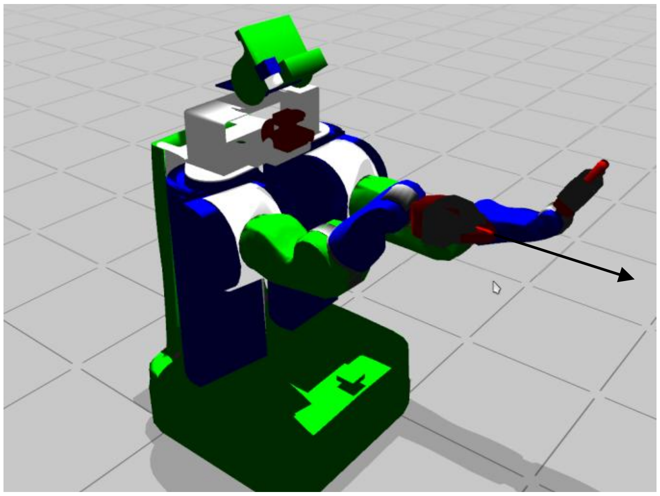

Figure 4 – PR2 in Gazebo, the zero configuration indicated by the arrow

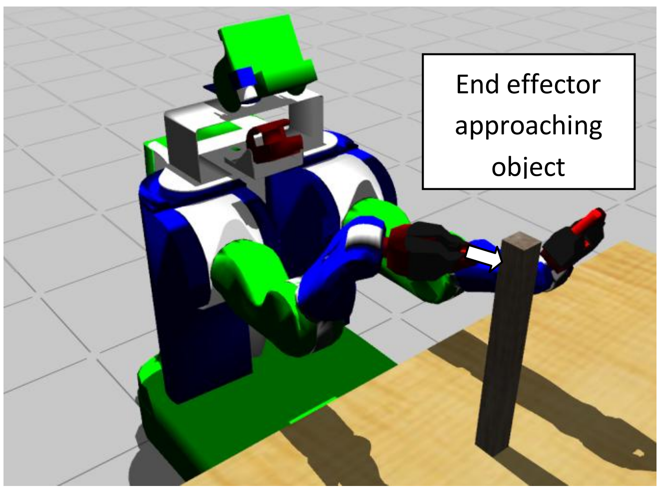

Figure 5 – PR2 in Gazebo, object manipulation

V. CONCLUSIONS

In this paper, we have discussed a novel approach to object manipulation using the PR2 robot. We started the paper by first providing some of the related background material in manipulation and motion planning, and then the motivation for this paper. We then proved both theoretically and experimentally that decoupling the orientation planner from the position planner provides better performance and is easily implemented using any existing planner. We have demonstrated all our experimental results in Gazebo, which is a robust platform for simulation and hence, we can confidently claim that the results of the simulation will carry over without much change to the real world. We hope that this paper will significantly benefit the field of object manipulation.

VI. ACKNOWLEDGEMENTS

We are very thankful to Mr. Ben Cohen, a PhD student at the Planning Lab at the University Of Pennsylvania for all his help in setting up the ROS simulations.

VII. REFERENCES

[1] Steven M. LaValle, Planning Algorithms, Cambridge University Press, 2006, available online at www.planning.cs.uiuc.edu.

[2] Willow Garage, www.willowgarage.com

[3] ROS, www.ros.org

[4] Benjamin J. Cohen, Sachin Chitta, Maxim Likhachev, Search Based Planning For Manipulation With Motion Primitives, IEEE Internation Conference On Robotics and Automation, 2010.

[5] Sachin Chitta, Benjamin J. Cohen, Maxim Likhachev, Planning For Autonomous Door Opening With a Mobile Manipulator, IEEE Internation Conference On Robotics and Automation, 2010.

[6] SBPL (Search Based Planning Library), www.ros.org/wiki/sbpl.

[7] Maxim Likhachev, Geoff Gordon, Sebastian Thrun, ARA*: Anytime A* with Provable Bounds on Sub-Optimality, In Advances in Neural Information Processing Systems (NIPS) 16. Cambridge, MA: MIT Press, 2003.