Motion Planning of Robotic Systems for Applications in Nuclear Facilities Clean Up

Peter S. March, Chetan Kapoor and Delbert Tesar
Robotics Research Group, The University of Texas at Austin
Mail Code R9925, Austin, TX 78729-1100 USA
Email: **pmarch@mail.utexas.edu**

Abstract-*Robotic systems are often used in the clean up of nuclear facilities due to the uncertain environments and high levels of radiation. The wide variety of tasks involved in nuclear facilities clean up will require that systems constantly be reprogrammed as tasks are added or changed. This creates the need for a generalized motion planning software capable of being applied to a wide variety of systems and tasks. This paper describes the basics of motion planning and the development of motion planning software. The motion planning software is then applied to a demonstrative Deactivation and Decommissioning (D&D) task to illustrate enhanced performance.*

I. INTRODUCTION

The Robotics Research Group (RRG) at the University of Texas at Austin is a member of the University Research Program in Robotics (URPR) consortium conducting research in robotic applications for nuclear facilities clean up. Robotic technology promises to greatly reduce costs and the on-site involvement of humans in these hazardous environments[1]. Specifically, the applications for robotics are:

- Decommissioning and Dismantlement of Nuclear Facilities
- Mixed Waste Operations
- Tank Waste Retrieval
- Maintenance of Nuclear Reactors
- Plutonium Processing (Glovebox Operations)

Each of these areas requires a combination of tele-operated and fully automated robotic tasks. For example, a system might be remotely controlled to retrieve a work piece. Then, an automated task could be used to cut the work piece.

The software described in this paper was developed as a part of the Operational Software Components for Advanced Robotics (OSCAR) environment[2]. OSCAR is a set of C++ libraries for control and simulation of robotic systems and is developed at the University of Texas at Austin. OSCAR is robot independent and supports multi-criteria decision making for improved performance.

II. THEORETICAL BACKGROUND

Motion planning is the development of smooth trajectories for robotic systems[3]. These trajectories can be computed either in joint space or end-effector (Cartesian) space. Thus, the system can be moved between initial and final joint positions or commanded to move along a path in end-effector space. Currently, most robotic systems rely on linear interpolation routines to move from initial to final positions. These routines do not produce a smooth motion in the system and can lead to wear and tear on the actuators and vibrations in the system.

Trapezoidal motion specification is used for this study. This method involves defining one of the derivatives of the motion curve to have a trapezoidal form. The derivative that the trapezoidal form is defined in is known as the order of the motion system.

The techniques used to generate the trapezoidal motion profiles and their integrals have been fully generalized by Tesar and Matthew[4]. For example, a second order system will have a motion profile similar to the one shown in Figure 1.

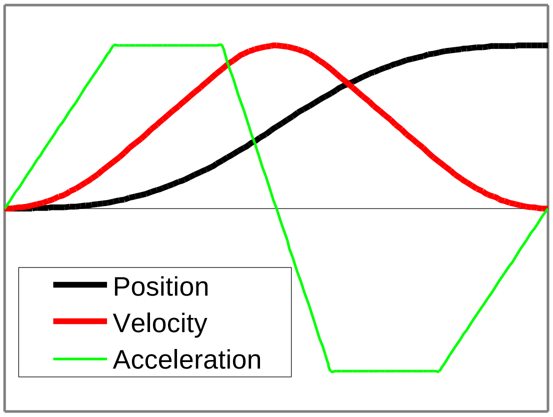

Figure 1. 2nd Order Motion Profile

The position as a function of time is then determined by integration. Higher order systems will result in smoother motion, because the profiles will remain continuous in higher derivatives.

III. SOFTWARE DEVELOPMENT

There are several important requirements for the software implementation of a motion planner. The planner should be generalized enough that it can produce trajectories in both joint space and end-effector space. These trajectories should be computed over a specified time interval and with respect to certain constraints (maximum velocity, maximum acceleration, etc). Finally, the planner should have off-line and on-line motion planning capabilities.

The difference in off-line and on-line motion planning relates to the need for tele-operated and fully automated tasks. In off-line motion planning, the trajectories are all computed beforehand. These trajectories can then be stored and reused for automated tasks. In on-line motion planning, the motion profiles are created in real-time as the operator is controlling the robot. This requires that the kinematics and motion profiles be computed in real-time and passed to the robot by the planner.

Figure 2 shows the hierarchy of C++ classes developed for the motion planner. The MotionPlan class is the base class and contains the necessary routines to generate a trapezoidal motion profile. Both the online and offline motion planners use these routines. The MotionParameters class takes in information about the time interval for the trajectory, the initial and final conditions, and the constraints imposed on the variables. The MotionParameters class then passes this information to the OfflineMotionPlanner. If the trajectory can be completed in the given time interval with the given constraints, the OfflineMotionPlanner computes and outputs a trajectory that can be executed by the robot controller. For this study, the development of the OfflineMotionPlanner was completed.

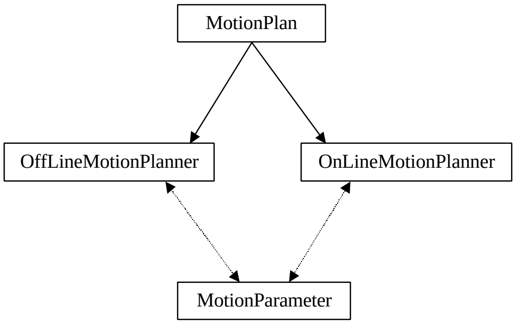

Figure 2. Class Hierarchy for Motion Planning

IV. EXPERIMENTAL RESULTS

In order to demonstrate the benefits of motion planning, the software was tested on a Robotics Research Corporation K/B 2017 17 DOF dual-arm robot (shown in Figure 3). This robot contains built in motion planning at the joint level that calculates trajectories between individual set points sent to the robot. However, this can still lead to excessive accelerations when these set points are far apart. Thus, motion planning across the entire trajectory is also important.

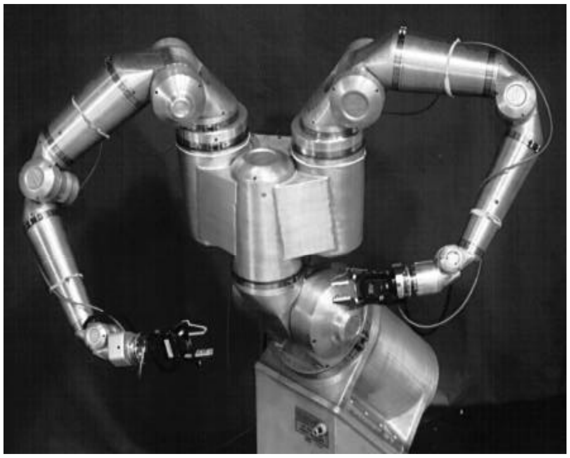

Figure 3. Robotics Research Corporation K/B 2017 17 DOF dual-arm robot

I. Experimental Setup

Two tasks were chosen for this experiment. The first task involves the left arm of the robot going through a large movement. This movement is calculated at the joint level and will test joint space planning. In the second test, the end-effector of the left arm traces a square in space. This will test end-effector space planning.

For both of these experiments, the torques and positions of the actuators were measured and recorded every 20 milliseconds. These experiments were repeated for the robot using linear interpolation, a 1st order motion profile, and a 2nd order motion profile. In each of these tests, the same number of steps were calculated so that the trajectories would be completed in the same time interval.

Tesar and Matthews[4] formulated five motion curve criteria to measure the performance of 1-DOF cam systems. While these criteria do not capture the entire dynamics of a robotic system, they can be applied to the actuator level to compare performance. The following are the criteria used in this work:

- y'_{max} - This is maximum velocity of the motion and provides a good indicator of the inertial energy.
- y''_{rms} - This is the root mean square of the acceleration curve. High values are an indication of high inertial forces.
- $\Delta y''$ - This is the peak-to-peak value of the acceleration curve. It indicates the range of inertial loading.
- $\Delta\tau$ - This is the peak-to-peak value of the torque curve. It indicates the range of torque demands.
- τ'_{max} - The maximum value of the derivative of the torque curve.

It is desirable to have lower values for each of these criteria. For brevity, only the results for the first joint (shoulder joint) of the left arm are presented. However, similar results were found at the other joints.

II. Joint Space Planning

In joint space planning, individual trajectories are generated for each joint between their starting and ending positions. The motion planner will provide very smooth motion profiles for each joint (see Figure 1) using this technique. However, it should be noted that the dynamics of the outer links would also have some effect on the performance of the first joint due to the coupling in the system. Table 1 shows the results of this experiment.

	Linear	1st Order	2nd Order
y'_{max} (rad/s)	0.40	0.59	0.87
y''_{rms} (rad/s^2)	1.78	0.57	0.76
$\Delta y''$ (rad/s^2)	39.61	1.48	2.04
$\Delta\tau$ (N-m)	114.0	57.0	85.8
τ'_{max} (N-m/s)	560.4	184.8	235.2

Table 1. Joint Space Results

Since lower values for these criteria indicate improved performance, this table shows that motion planning exhibits improvement over linear interpolation for all of these criteria except y'_{max}. The y'_{max} is larger for the motion planning tests, because the velocity is gradually ramped up to its maximum. This means that the maximum velocity will have to be larger to travel the same distance in the same time (i.e. the area under the velocity curve is constant). This result is shown in Figure 4.

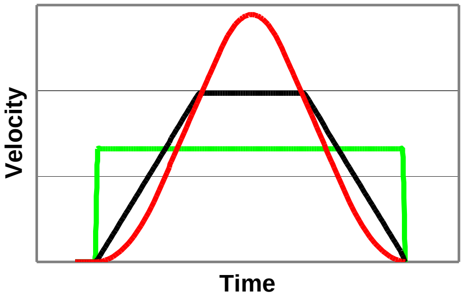

Figure 4. Velocity Profiles of Different Motion Curves

It is also apparent from Table 1 that the performance of the 1st order system was better than the 2nd order system. This is most likely a result of the higher inertial energy in the system caused by the higher velocities.

III. End-Effector Space Planning

In end-effector space planning, the motion curves are generated in Cartesian space for 6 DOF (three position, three orientation). A separate motion curve is generated for each of these DOF, and the joint values are then calculated at each set point through inverse kinematics. Table 2 shows the results of the second experiment.

	Linear	1st Order	2nd Order
y'_{max} (rad/s)	0.53	0.69	0.88
y''_{rms} (rad/s^2)	2.98	2.16	2.46
$\Delta y''$ (rad/s^2)	39.62	27.72	24.43
$\Delta\tau$ (N-m)	144.3	104.3	136.4
τ'_{max} (N-m/s)	566.0	286.7	374.6

Table 2. End-Effector Space Planning

Since the motion in this experiment is planned at the end-effector, there is little control over what is happening at the joint level. The improvement of motion planning over linear interpolation for this test is less dramatic, because a smooth motion at the end-effector does not necessarily mean a smooth motion at the joint level. However, motion planning once again improved performance in every category besides y'_{max}.

IV. DOE Demonstration

The Robotics Research Group at The University of Texas previously completed a demonstration using the K/B 2017 for a D&D task[5]. This demo involved an automated material reduction task where one arm holds a work piece and the other cuts it. This demonstration uses both joint space planning and end-effector space planning.

The demonstration previously used a linear interpolation routine to move between the set points necessary to complete the task. The motion planning software was integrated into the demo to compute 1st and 2nd order trajectories between these set points. The revised demo (with motion planning added) was observed to be much smoother with visibly reduced vibrations and shocks. Figure 5 shows the torque profiles from joint three of the left arm (the arm used to cut the work piece) generated from the demo.

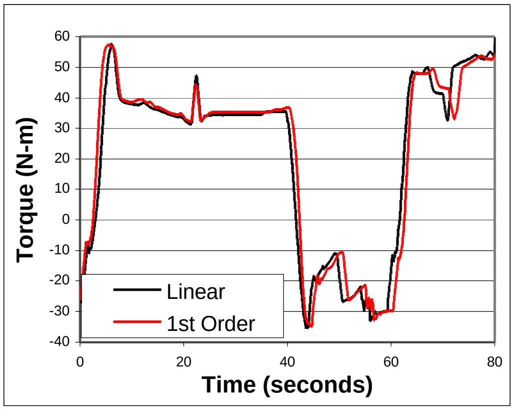

Figure 5. Torque Profiles from DOE Demo

Due to the large range of torques found in this demo, the benefit of motion planning is hard to infer from the graph. However, some of the peaks in the graph are noticeably smaller and not as sharp for the 1st order system. Another thing to notice in this plot is the way the 1st order profile lags behind the linear towards the end of the demo. This is a result of the calculation time for the motion planning trajectory. Considering the length of the demonstration, this time lag is very small.

CONCLUSIONS/DISCUSSIONS

A generalized motion planner is an essential component for the use of robotic systems for nuclear facilities clean up. The software developed in this study allows the user to quickly reprogram a variety of systems for a variety of automated tasks. The addition of motion planning has also b\een shown to improve system performance. This software has also been added to a current robotic application to demonstrate the ease of integration. The further development of on-line motion planning will allow the robot to move smoothly during tele-operation.

However, the software developed in this study only addresses the problem of trajectory generation. In essence, trajectory generation is only one element of motion planning. Path planning (determining how to move from initial to final configurations) and an understanding of the dynamics and couplings of the system as a whole are also important. The problem with only including trajectory generation is particularly evident in the results from end-effector planning. While the trajectory generation still improved the performance, the improvement was less impressive than with joint-space planning. In order to truly improve the performance of a system, it will be necessary for a motion planner to incorporate all the aspects of motion planning.

REFERENCES

1. National Research Council. 2001. “Research Opportunities for Deactivating and Decommissioning Department of Energy Facilities.” Washington, DC: National Academy Press.

2. Kapoor, C., Tesar, D., "A Reusable Operational Software Architecture for Advanced Robotics," <u>Proceedings of the Twelfth CSIM-IFToMM Symposium on theory and Practice of Robots and Manipulators</u>, Paris, France, July 1998.

3. Craig, J.J, “Introduction to Robotics: Mechanics and Control,” Addison-Wesley, 1986.

4. Tesar, D., and Matthew, G., “The Dynamic Synthesis, Analysis, and Design of Modeled Cam Systems,” Lexington Books, D. C. Heath & Company, 1976.

5. Harden, T., Pryor, M., Rajan, R., Yoo, J., Kapoor, C., and Tesar, D. “Prototyping D&D Tasks Using a Dual-Arm Robotic System.” <u>Proceedings of the ANS 9th International Topical Meeting on Robotics and Remote Systems</u>, March 2001.