Multi-optimization Criteria-based Robot Behavioral Adaptability and Motion Planning

DOE Project Number: 82794

PI: Dr. François G. Pin, pin@ornl.gov, (865) 574-6130

Research Objectives:

Our overall objective is the development of a generalized methodology and code for the automated generation of the kinematics equations of robots and for the analytical solution of their motion planning equations subject to time-varying constraints, behavioral objectives and modular configuration.

Robotic tasks are typically defined in Task Space (e.g., the 3-D World), whereas robots are controlled in Joint Space (motors). The transformation from Task Space to Joint Space (through the inverse kinematics equations) must consider the task objectives (e.g., high precision, strength optimization, torque optimization), the task constraints (e.g., obstacles, joint limits, non-holonomic constraints, contact or tool task constraints), and the robot kinematics configuration (e.g., tools, type of joints, mobile platform, manipulator, modular additions, locked joints). Commercially available robots are optimized for a specific set of tasks, objectives and constraints and, therefore, their control codes are extremely specific to a particular set of conditions. Thus, there exist a multiplicity of codes, each handling a particular set of conditions, but none suitable for use on robots with widely varying tasks, objectives, constraints, or environments. On the other hand, most DOE missions and tasks are typically “batches of one”. Attempting to use commercial codes for such work requires significant personnel and schedule costs for re-programming or adding code to the robots whenever a change in task objective, robot configuration, number and type of constraint, etc. occurs.

The specific goal of our project is to develop a single “generic code” that can automatically generate the inverse kinematics equations of robots, is useable for all types of robots (mobile robots, manipulators, mobile manipulators, etc.) with no limitation on the number of joints and the number of controlled Task-Space variables, and can solve the motion planning equations in analytical form with real time (at loop rate) changes in number and type of constraints and in task objectives, while also adapting to changes in kinematics configurations (change of module, change of tool, joint failure adaptation, etc.).

Our overall approach is based on a novel method for the resolution of under-specified systems of algebraic equations, involving both a novel formulation of the general solution as:

$$S = \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix} \begin{bmatrix} m \\ m \end{bmatrix}, \begin{bmatrix} \bar{q}(t_1, K, t_{m \square n+1}) \end{bmatrix} = \begin{bmatrix} t_i \bar{g}_i \\ t_k \end{bmatrix} \begin{bmatrix} m \square n+1 \\ m \square n+1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and a Lagrangian-based constrained optimization

$$L(t_i, \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix}, \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix}) = Q(t_i) + \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix} \begin{bmatrix} m \square n+1 \\ m \square n+1 \end{bmatrix} t_i \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} r \\ r \end{bmatrix} \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix} C^j(t_i)$$

where

$$Q(\begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix}(t_i)), i = 1, m \square n + 1,$$

is the general criterion to be optimized, and

$$C^j(\bar{q}, \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix}(t_i)) = 0, j = 1, r;$$

represents a set of r general constraints. The unique aspect of this approach is that it leads to analytical solutions that simultaneously fulfill the optimization criterion and all constraints applicable at the particular time-step (loop rate) [1, 2, 3, 4].

Research Progress and Implications:

As of the first year and a half of this 3-year project, we have completed the methodology development and corresponding code for three major modules of the overall software system. The first is the automatic generation of the forward kinematics equations for robots with arbitrary numbers of actuated joints. From the Denavit-Hartenburh (DH) parameters representing each link-joint module, the transformation matrices representing the modules are automatically generated to produce the global transformation matrix of the complete system, from which the forward kinematics equations are extracted, in symbolic form. From this symbolic representation, automated differentiation is performed to automatically generate, in symbolic form, the Jacobian matrix for the overall system. This approach is valid for any link-joint system and, therefore, it applies to tools as well as

manipulators and mobile manipulators. Because the automated generation can be invoked at any time during operation, immediate adaptation of the kinematics and motion planning equations is now possible when changes of configuration occur (e.g., modular robots, change of tools, etc.). Because it impacts the entire field of robotics by allowing automated generation of robots equations, this module, which has been named the “JFKengine” (Jacobian and Forward Kinematics engine) is being prepared for wide dissemination, with full source code, through availability via a public website. A technical report on the JFKengine operation has been issued [5] and a Journal article is in review for publication [6]. As an example, Fig. 1 displays a sample of the motion planning capability when a tool is being picked up, used on a trajectory, and then released. At the pick-up and release events, the JFKengine is invoked and the system’s equations automatically updated to reflect the new kinematics. While the tool is held, the end point of the tool is being controlled through the inverse kinematics and motion-planning algorithm.

Figure 1. Results of a motion planning simulation involving tool changing. The trajectory of the manipulator starts on the right without a tool; the arm then picks up a 2-joint articulated tool (hence changing the kinematics) and moves it out toward the bottom left of the picture (the tool isn't shown as part of the manipulator until it is grasped). It subsequently puts the tool down again (making the arm effectively shorter) and moves to the next location to the upper left of the figure.

The second aspect of the overall methodology that has been completed involves a dynamics simulation system allowing visualization with full 3D geometric accuracy. It models Newtonian mechanics with coulomb friction. By exploiting recently developed rigid-body simulation techniques, it is able to simulate complex systems (even with multiple interacting robots) in real-time. This dynamic simulation capability opens the possibility of research using hybrid real and simulated systems – 'hardware-in-the-loop' simulations, and will allow us to explore the issues involved with use of the same motion planning code to drive real robots or their simulation, in particular during changes of configurations, constraints, and task objectives.

The third major activity has focused on enhancing the earlier version of the basic solver code, i.e., the code generating the entire solution space for the under-specified system of equations and performing the constrained optimization. The code was verified for proper resolution in comparisons with the standard Moore-Penrose pseudo-inverse in sample problems involving manipulators and mobile manipulators with various configurations and degrees of freedom, and for proper handling of constraints that could change at loop-rate, including avoidance of multiple obstacles and joint limits. Emphasis this year has been on developing analytical solution methods for additional types of constraints, such as linear or planar motion constraints for tools or non-holonomic constraints for mobile platforms. For example, Fig 2 shows a sample problem involving the motion of a mobile platform with a car-

like kinematics, i.e., with a very strong non-holonomic constraint. The significance of the approach successfully handling non-holonomic constraints (which due to their non-integrable nature cannot be handled by other approaches), is to reinforce that the capability of the novel constrained optimization scheme to handle widely varying types and numbers of constraints and criteria, that can change at loop rate, through a single resolution code. A journal article presenting the developed methodologies in this constrained optimization area is in review for publication [7].

Figure 2. Simulation results of the motion planning for a car-like robot with an 8-link manipulator mounted towards the front left. The platform is displayed as a rectangle and the manipulator as a series of line segments. The manipulator is to follow a straight-line trajectory from right to left. This causes the platform to first move backwards and then turn to drive forward, displaying the “cusp” maneuver typical of non-holonomic steering mechanisms.

Planned activities:

During the remainder of this second year of activities, the basic solver and optimization code will be finalized to be suitable for use directly in both simulation mode and in actual target robot control hardware. Additional analytical solutions approaches for handling, in bounded algorithmic time, novel sets of constraints and optimization criteria will be derived for inclusion in the overall resolution code. With completion of the integrated version of the complete code, we will then perform extensive computational experiments to establish characteristic orders, complexity, and completeness of the algorithms. In FY-04, we will focus on investigations of non-linear task objective criteria and corresponding analytical resolution. For each solution strategy and implementation, we will perform experimental investigations of complexity and bounded time through comparative computational experiments to assess preferred resolution approaches for these objective functions. We will also focus on proof-of-principle experiments using kinematics configurations of existing D&D robots at ORNL, including implementation on an actual robot control system to demonstrate pervasiveness of the single-code approach.

Information access:

Further background and details of the methodologies can be found in the following publications: 1 - F. G. Pin et al., *Journal of Robotic Systems* **13** (11), 723–736 (1996)

2 - F. G. Pin and F. A. Tulloch. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 22–28, 1996, Vol. 1, 468–473

3 - F. G. Pin et al. Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, California, May 8–13, 1994, 96–102.

4 - F. G. Pin et al. Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 21–27, 1997, Vol. 4, 2914–2919.

5 - K. N. Fisher, D. L. Jung, F. G. Pin and W. E. Dixon “JFKEngine: A Jacobian and Forward Kinematics Generator,” ORNL/TM-2002/230 (2002).

6 - F. G. Pin, K. N. Fisher and D. L. Jung, “Automated Generation of the Forward Kinematics Equations and Jacobian for Modular and Re-configurable Robots”, in review for *IEEE Trans. Robotics and Automation*.

7 – F. G. Pin and D. L. Jung, “Motion Planning of Kinematically Redundant Robots with Time-varying Constraints,” in review for *IEEE Trans. Robotics and Automation*.