OAK RIDGE NATIONAL LABORATORY

FACT SHEET

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Automated Inverse Kinematics and Motion Planning of Robotic Systems with Time-Varying Constraints, Objectives, and Configurations

Contrary to the repetitive tasks performed by industrial robots, the tasks by autonomous robots functioning in unstructured environments (environmental restoration, Decontamination and Decommissioning (D&D), battlefield, etc.) can be characterized as "batches-of-one," in which robots must be capable of adapting to changes in constraints, tools, environment, criteria, and configuration, in real-time. 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 exists 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.

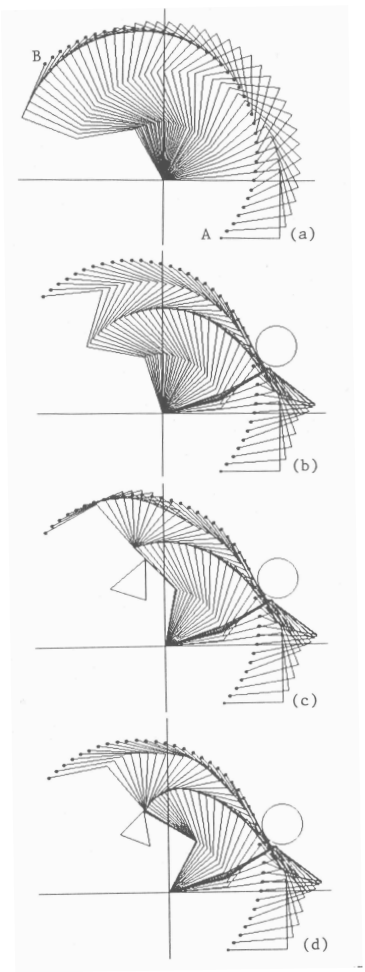

Fig. 1. Motion planning solutions for a circular trajectory (from A to B) with multiple obstacles.

We have designed an approach—that we have named FSP (for Full Space Parameterization)—to resolve this shortcoming and have developed a single "generic code" that can automatically generate the kinematic equations of robots from simple D-H representations, 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 (e.g., non-holonomic constraints, joint limits, obstacle avoidance) and in task objectives (e.g., high precision, strength optimization, torque optimization), while also adapting to changes in kinematic configurations (change of module, change of tool, joint failure adaptation, etc.). The method is based on a novel constrained optimization solver for the general solution of under-specified systems of algebraic equations that is suitable for solving, analytically, the inverse kinematics of (possibly kinematically redundant) robots.

Figures 1 and 2 show applications of the FSP code to sample 2D and 3D manipulators with joint limit and obstacle constraints.

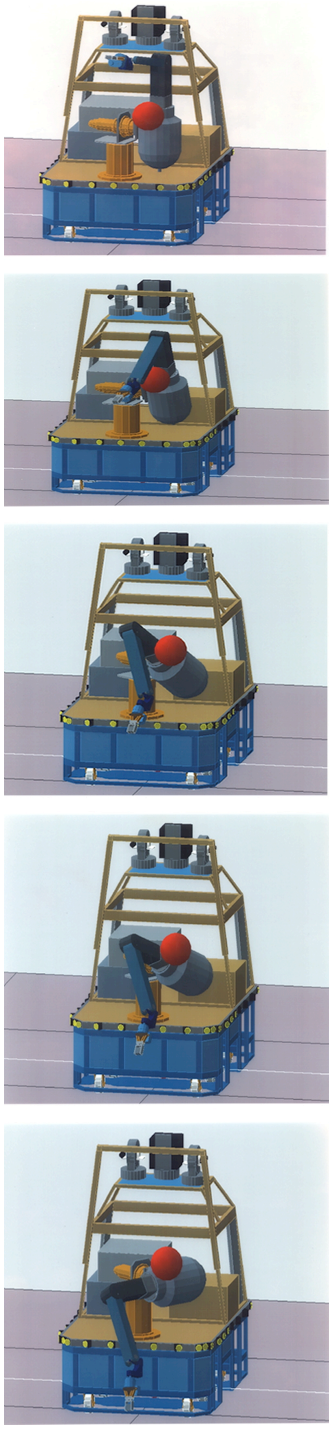

Fig. 2. Motion planning with obstacle avoidance for a 10-degree-of-freedom system.

Figure 3 displays the motion of a 9-joint manipulator mounted on a car-like wheeled platform with a non-holonomic constraint performing a task involving the acquisition of a tool (a 2-dof drill) to drill two holes and release of the tool. The trajectory to be followed by the end-effector is labeled in three sections A, B, and C. First, the end-effector moves along segment A to pick up the drilling tool. The rectangular platform is initially facing right while reversing to the left in a shallow turn. The drilling tool is a 2-dof articulated tool (pitch and yaw). When the drill is connected, bringing the system to a total of 14 dof including the platform, the kinematic equations and the Jacobian matrix are automatically regenerated to incorporate the extra 2 links. Next, the end-effector (now the end of the drilling tool) moves along the trajectory segment B, controlling the orientation of the drill to maintain a fixed drill-bit axis during drilling of the first hole. This motion is repeated for a second hole. During all these motions, constraints such as joint limits and the non-holonomic constraint on the platform are active, and the cusp-like motion of the platform typical of car-like forward- and reverse-maneuvering is clearly visible. In segment C of the trajectory the manipulator moved to put down the drill. Once the drill has been released, triggering another re-computation of the kinematic equations and Jacobian matrix, the platform moves off to the left.

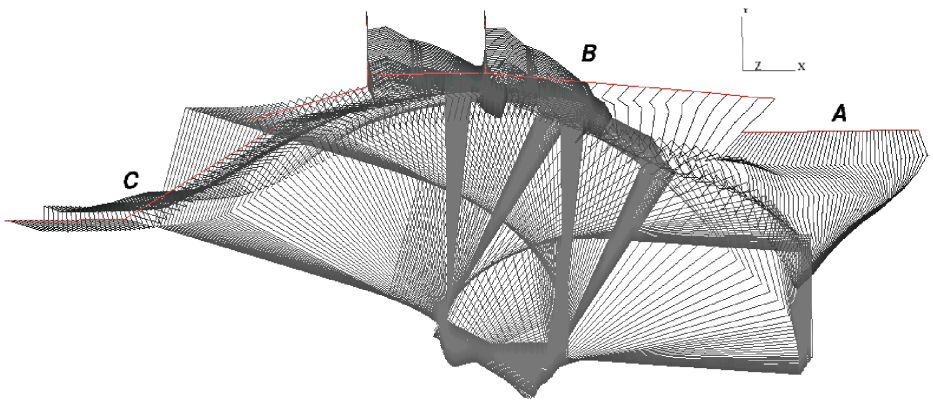

Fig. 3. Trajectory for hole-drilling task with tool change and constraints.

In Fig. 4, four kinematically redundant manipulators are operating simultaneously in a surgical cell. The surgical manipulators attached to the ceiling plate are remotely operated by the surgeon. The other two 9-dof manipulators are autonomous and perform nurse-type tasks, such as handling and changing tools and supplies or holding human tissues, while maintaining collision avoidance with the surgical arms. The instances of detected proximity, each triggering a collision avoidance constraint, are illustrated by the yellow “sticks” between the various links. Both the teleoperated and the autonomous arms utilize the constrained optimization algorithm for motion planning and redundancy resolution.

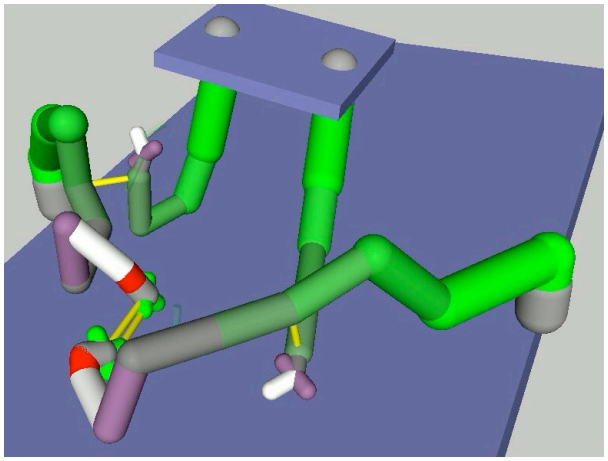

Fig. 4. Multiple manipulators in a surgical robotic cell.

Point of Contact:

Dr. François G. Pin (865-574-6130, pinfg@ornl.gov)
Oak Ridge National Laboratory, P.O. Box 2008,
Oak Ridge, TN 37831-6305