Compliance Viewed as Programming a Damped Spring

Stephen J. Buckley *
Manufacturing Research
IBM Thomas J. Watson Research Center
Yorktown Heights, NY

Abstract

Parts mating often requires the use of compliant motions, which cause an object in the grasp of a robot to slide along obstacles in its environment. This paper is about the interface between a compliant motion programming system and a compliant motion control system. We propose that in this interface the robot can be modeled as a damped spring. This model allows the programming system to specify and reason about compliant motions without worrying about low-level control details. The utility of the damped spring model is demonstrated by applications in teaching and planning of compliant motion strategies.

1 Introduction

A *compliant motion* causes an object in the grasp of a robot to slide along obstacles in its environment, using them as guides toward a goal region. This type of motion is particularly useful for parts mating.

For a robot to perform a variety of compliant tasks, its compliance must be programmable. Figure 1 shows the logical structure of a robot system. The user gives task information to the programming system via the user interface, which is usually a high level language or a teleoperator-based teaching system. In the future, tasks will probably be presented via geometric models. Once the programming system has been presented with a user task, it sends compliant motion commands to the control system via the control interface. Finally, the control system sends hardware commands to the robot via the robot interface.

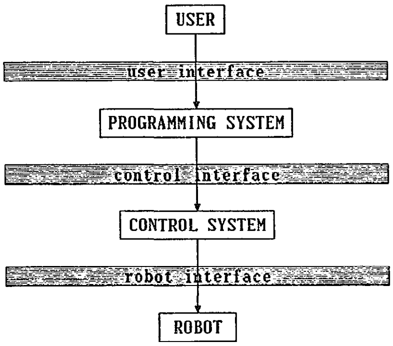

Figure 1: The logical structure of a robot system.

This paper is about the programming system, and in particular, its interface to the control system. It is the task of the control system to transform the actual dynamics of the robot to desired dynamics. We propose that for compliant motion the desired dynamics should take the form of a damped spring. We will demonstrate the utility of the damped spring model for two different types of programming systems, a robot teaching system, and a model-based planning system. We will not address control issues here. Work is in progress elsewhere to implement compliant control systems of this nature [Whitney 1985].

This section of the paper presents an example of a compliant motion strategy, and describes the damped spring model. Sections 2 and 3 sketch the application of the model to robot teaching and planning, respectively. Section 4 presents concluding remarks.

1.1 An Example

Figure 2 depicts a three-dimensional T-shaped part, in the grasp of a robot. An obstacle is shown which contains a hole with an adjoining slot. First, the T-shape is to be inserted into the hole. Then, the shaft of the T-shape is to be slid into the adjoining slot.

*Performed at the MIT Artificial Intelligence Laboratory. Supported in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-85-K-0124, in part by General Motors Research Laboratories, under Contract 50-10, in part by the National Science Foundation, under contract 8451218-DMC, and in part by the IBM Corporation.

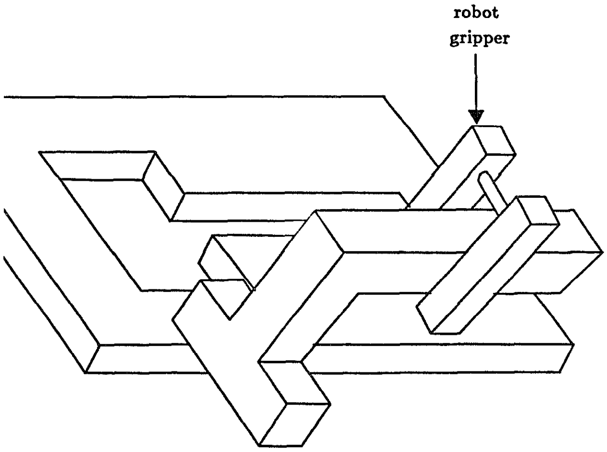

Figure 2: An insertion task. A three-dimensional T-shaped part is in the grasp of a robot. First, the T-shape is to be inserted into the hole of the obstacle. Then, the shaft of the T-shape is to be slid into the slot which adjoins the hole.

We will assume that the robot can translate in three dimensions, but cannot rotate.

Finding a solution to this problem is complicated by uncertainty in the initial configuration of the robot, in the position and force sensing of the robot, and in controlling the position and velocity of the robot. We will assume that these uncertainties are bounded.

We can use the *configuration space* representation [Lozano-Pérez 1983] to simplify the geometry of the problem. Let **r** be an arbitrarily chosen reference point on the robot. Consider the positions which **r** can take without causing a collision between the T-shape and the obstacle. Each face of the obstacle imposes a constraint on the free motion of **r**. These constraints are represented explicitly as configuration space surfaces in Figure 3. The configuration space surfaces form a sequence of two holes. The first hole represents the constraints on the T-shape while it is inserted into the hole in the obstacle. The second hole follows from an intermediate chamber at the bottom of the first hole, and represents the constraints on the shaft of the T-shape while it is sliding into the slot of the obstacle. Configuration space surfaces such as these comprise a representation of the task geometry that is equivalent to the original geometry, but more explicit. In the new representation, we can think of the robot simply as the reference point **r**. It is possible to represent arbitrarily complicated polyhedral environments using configuration space [Lozano-Pérez 1983].

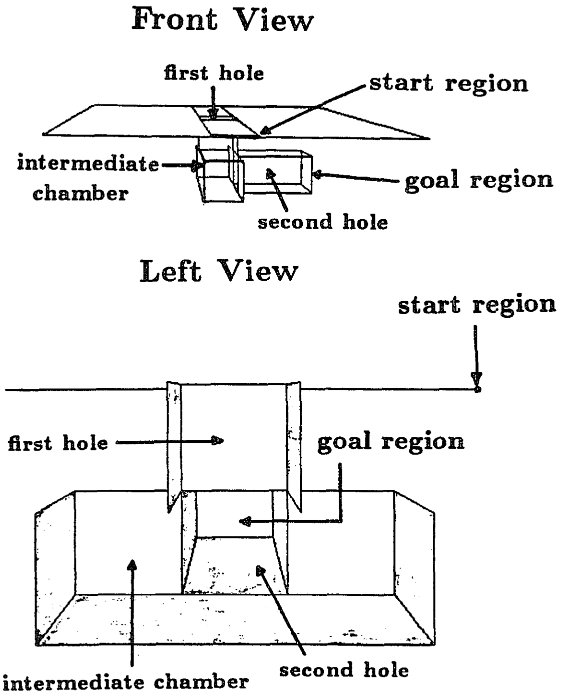

Figure 3: Front and left views of the configuration space representation of the T-shape insertion. The start region is an edge on the surface above the holes. The goal region is the bottom face of the second hole.

Assume initially that the T-shape is in contact with the top of the obstacle, laterally aligned with the hole. The configuration space representation of this start region is shown in Figure 3. The goal region is the bottom face of the second hole. We are to specify a sequence of compliant motions which moves the T-shape from the start region to the goal region despite bounded sensing and control errors in the robot.

The model-based compliant motion planner described in Section 3 was used to compute a solution for this problem. The planner returned a sequence of two compliant motions. Figure 4 shows the first commanded motion. The robot is to aim for any point in the black polyhedron. With this commanded motion, the robot will reach and stop on a face of the first hole, as shown in the figure. Friction between the T-shape and the hole face will cause it to stay there. The black polyhedron is behind and more narrow than the stopping region to account for possible trajectory errors. The stopping region then becomes the start region for the second commanded motion. Figure 5 shows the second commanded motion. If the robot aims for any

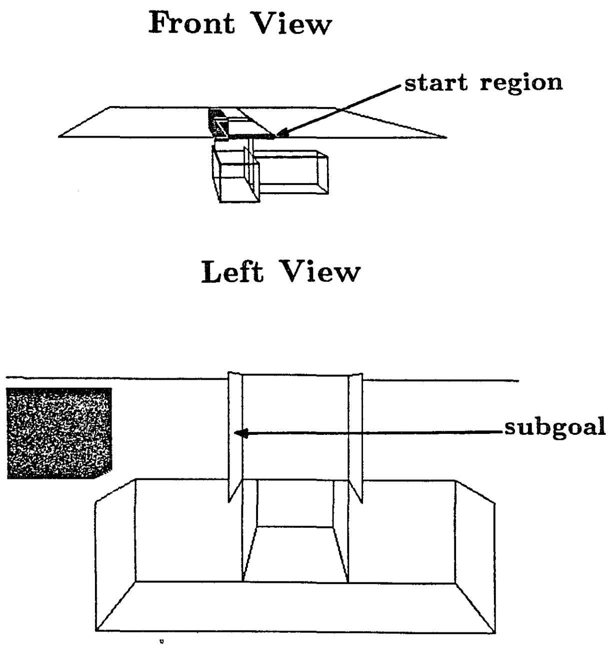

Figure 4: Front and left views of the first commanded motion. The start region is an edge on the surface above the holes. The subgoal for the motion is a face of the first hole, as shown. To attain the subgoal, the robot should aim for any commanded position in the black polyhedron.

commanded position in the black polyhedron shown in the figure, then the robot will enter the second hole, slide along the lower side of the hole, and stop in the goal region.

1.2 Assumptions

We will use polyhedral models to represent the geometry of the robot and its environment. Many objects can be accurately modeled by polyhedra. Curved objects can be approximated by polyhedral models with many faces.

To simplify the computations, we will assume that the robot can translate in three dimensions, but cannot rotate.

We will assume that the robot is equipped with three-dimensional position and force sensors, which have bounded uncertainty.

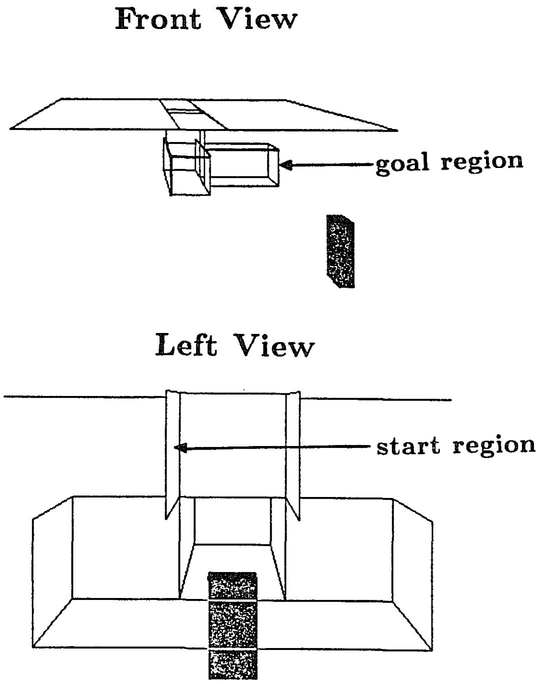

Figure 5: Front and left views of the second commanded motion. The start region is a face of the first hole, as shown. The goal region is the bottom of the second hole. To attain the goal, the robot should aim for any commanded position in the black polyhedron.

1.3 The Damped Spring Compliance Model

In order for a programming system to specify compliant motions, it is necessary to have an abstract model of compliance. Currently, most noncompliant robots are programmed by commanding position goals. We have extended this method to compliant robots by attaching an imaginary spring/damper combination between the robot and the commanded position, as illustrated in Figure 6. Then, we simply let the robot go to where the spring pulls it. This simple model is called the *damped spring* compliance model, and can be described mathematically by the equation

$$\mathbf{f}_r = b(\dot{\mathbf{x}} - \mathbf{p} + \mathbf{x}) \qquad (1)$$

where $\mathbf{f}_r$ is the reaction force on the robot, b is a damping constant, $\mathbf{p}$ is the desired position of the robot, $\mathbf{x}$ is the actual position, and $\dot{\mathbf{x}}$ is the actual velocity. Given a commanded position $\mathbf{p}$, this compliance model implies the following robot behavior under ideal con-

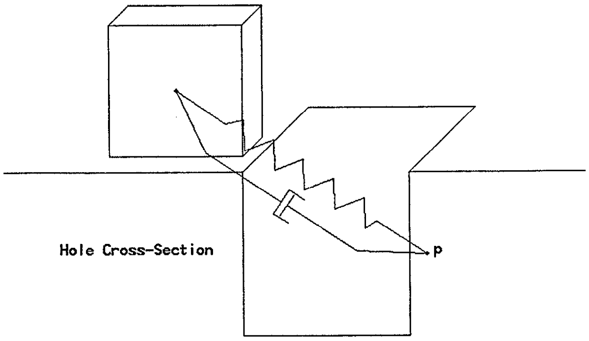

Figure 6: To specify the insertion of a block into a square hole, we attach an imaginary spring/damper combination between the block and a commanded position **p** behind the hole.

ditions: In free space, if **x** is not equal to **p**, the robot moves in a straight line from **x** to **p**. In contact, if **x** − **p** is not contained in the friction cone of the contact, the robot slides toward the projection of **p** onto the configuration space surface.

The damped spring compliance model is an extension of the *generalized spring* formulation of Salisbury [1980], and draws on ideas from Lozano-Pérez, Mason, and Taylor [1984]. It is also the first-order analog of the *generalized impedance* model [Hogan 1984]. We chose to ignore the second order terms present in Hogan's equation to avoid parabolic trajectories, which are more difficult to specify and reason about than straight-line trajectories. A lone spring would have been even simpler, but damping was necessary because the spring equation $\mathbf{f}_r = k\mathbf{x}$ does not contain a time parameter, and thus cannot be used to specify the trajectory of the robot. The damping constant b may affect the stability of the robot, but under stable conditions and it has no effect on the outcome of a motion, other than on the time it takes to execute the motion. Thus, the desired position **p** is the only parameter required from the user.

It is the task of the underlying control system to present the desired dynamics of Equation 1 to the teaching system. For short-term testing purposes, we executed compliant motions at low speed on an IBM 7565 robot using a simple feedback loop written in AML.

In the presence of sensing and control uncertainty, ideal trajectories cannot be attained reliably. Let ϵ_p be the maximum distance between a commanded position and the actual position attained by the control system

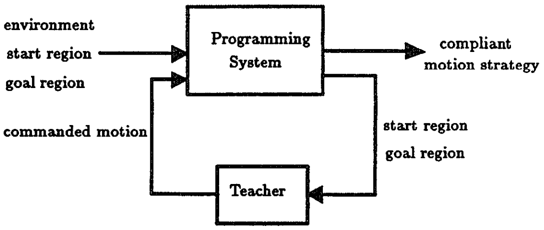

Figure 7: An operational view of the robot teaching system. The user submits a problem to the system, consisting of an environment model, a start region, and a goal region. Then, the user is asked to suggest a series of commanded positions.

in free space. Let θ_v be the maximum angle between a commanded velocity and the actual velocity attained by the control system in free space. In Buckley [1987], we show that, in terms of these control error bounds, the robot can track a desired trajectory to within

$$\epsilon_t = \epsilon_p + de^{\frac{\theta_v}{\tan\theta_v}} \sin\theta_v, \qquad (2)$$

where d is the distance between interpolation points along the commanded trajectory. If the trajectory controller is implemented as an analog circuit, then a continuous stream of control positions can be passed to the position controller, reducing ϵ_t to ϵ_p. The set of possible free space trajectories can thus be bounded by a cylinder of radius ϵ_t. When the robot strikes a surface, this cylinder is projected onto the surface, forming a planar cylinder of radius ϵ_t.

1.4 Motion Termination

The damped spring model allows one to program a compliant robot by issuing a commanded position. By choosing the commanded position carefully, one can often cause the robot to stop in a desired goal region by sticking. It is sometimes useful to specify other types of motion termination as well. Our system allows additional motion termination by position and force sensing. As the robot approaches a commanded position, it stops when its sensed position is contained in a specified set of termination positions, and the orientation of its sensed force is contained in a specified set of termination forces.

2 Application to Robot Teaching

This section describes the use of the damped spring model in an implemented robot teaching system. Figure 7 shows an operational view of the teaching system. The user submits a problem to the system, consisting of:

- a geometric model, representing the robot and its environment (e.g. workpieces, feeders, fixtures, tools).
- a start region, which contains all possible initial configurations of the robot.
- a goal region, in which the robot is to terminate under the desired compliant motion strategy.

The teaching system displays the start and goal regions graphically, and prompts the user. The user then submits a commanded position **p**. In principle, **p** could be entered by guiding the robot, or with a light pen. In our experiment, **p** was simply typed in. The user should choose **p** in the hope that it will cause the robot to reach the goal region from the start region. If this is impossible, then the user should choose **p** in the hope that it will cause the robot to reach the goal region from an achievable intermediate goal.

When **p** is entered, the teaching system computes a set R of configurations in which the robot is in contact with its environment, and from which the goal region can be reached reliably via **p**. R is called a *pre-image* of the goal region under the commanded motion **p** [Lozano-Pérez, Mason, and Taylor 1984]. The pre-image R is stored in a table along with **p**. R is now said to be *solved*, and is added to the goal region. If a subset S' of the start region is recognizably contained in R despite sensing uncertainty, then S' is solved, and can be removed from the start region. If the new start region is empty, then the problem is solved. Otherwise, the system displays the new start and goal regions, and user interaction continues.

By iteratively reducing the size of the start region, and increasing the size of the goal region, it is hoped that the user and system can together converge on a successful strategy.

Each solved region R in the final table is accompanied by a commanded position, which is to be issued upon reaching R. On execution, the robot looks up its present sensed position and force orientation in the table. If the table entry corresponds to a goal region, then the robot stops. Otherwise, it executes the corresponding commanded motion. This iterative lookup process implements a conditional test, which chooses the next commanded motion based on sensory input.

This teaching system has several advantages over previous teaching systems, including:

1. Compliant motions have traditionally been difficult to specify by teaching. In our teaching system, specifying a compliant motion is simply a matter of specifying a commanded position.
2. Conditional tests are difficult to specify by teaching. In our teaching system, conditional tests are inferred automatically.
3. Debugging a compliant motion strategy is time-consuming and costly. In principle, there is no need to debug the motion strategies produced by our teaching system; their reliability is ensured, assuming bounded uncertainty in the starting configuration of the robot, and in robot sensing and control.

Pre-images were first proposed by Lozano-Pérez, Mason, and Taylor as a subtask in a motion planner. Their proposal did not specify an implementation. Erdmann [1984, 1986] showed that for certain classes of termination conditions, pre-images can be computed by geometric backprojection. Erdmann implemented his scheme for planar robots, under the *generalized damper* compliance model [Whitney 1977]. Rotations in the plane were implemented by constructing slice projections for various ranges of rotations. We adapted Erdmann's algorithm to three Euclidean dimensions, using the damped spring compliance model. Under this compliance model, backprojection can be implemented by a series of three-dimensional set operations. The details of our algorithm are given in Buckley [1987, 1988].

3 Application to Robot Planning

This section describes the use of the damped spring model in a model-based compliant motion planner. Canny and Reif [1987] showed that the problem of planning compliant motions with uncertainty is exponential time hard. To simplify the problem, we approximate the environment of the robot as a finite state space. Each state is a set of configuration space vertices, edges, and faces. The planner searches for a compliant motion strategy by repeatedly choosing a state, and constructing arcs which connect the state to other states. An arc represents a set of commanded positions which are guaranteed to get from one state to another. Arc construction proceeds until a successful compliant motion strategy has been constructed from the start state to a goal state.

The main computational activity in the planner is arc construction. Thanks to the generalized spring compliance model, this can be implemented by a series of three-dimensional set operations. The details of our algorithm are given in Buckley [1987].

The planner was implemented and applied to the problem shown in Figure 2. The configuration space environment for this problem contains 118 vertices, edges, and faces (Figure 3). In this nontrivial environment, the planner synthesized the two-step motion strategy shown in Figures 4 and 5.

4 Conclusions

The damped spring model is a useful model for programming compliant motions, for the following reasons:

1. It is easy to specify compliant motions using the model. One needs to specify a commanded position and a termination condition, which consists of a set of termination positions and a set of termination forces. Specifying a commanded position is already a popular method of programming robots. Specifying termination conditions is not quite as commonplace, but in Sections 2 and 3 we have shown that a programming system can automatically compute termination conditions from task information.

2. Using the model, one can specify compliant motions with wide utility. Commanded positions in combination with termination conditions provide a great deal of flexibility.

3. Reasoning about damped spring motions is basically an application of set theory, as illustrated by the teaching and planning examples. With three degrees of freedom or less, set operations can be implemented using traditional computational geometry. Above three dimensions, there are some computational difficulties, which are the subject of ongoing research.

Acknowledgments

Tomás Lozano-Pérez advised this research. John Canny, Bruce Donald, and Mike Erdmann provided technical advice. Jim Korein gave advice on the presentation.

References

Buckley, S. 1987."Planning and Teaching Compliant Motion Strategies", Ph.D. dissertation, MIT Department of Electrical Engineering and Computer Science, also AI-TR-936, MIT Artificial Intelligence Laboratory.

Buckley, S. 1988. "Teaching Compliant Motion Strategies", to be published in *IEEE Journal of Robotics and Automation.*

Canny, J., and Reif, J. 1987. "New Lower Bound Techniques for Robot Motion Planning Problems", *28th Annual Symposium on Foundations of Computer Science*, Cambridge, Massachusetts, pp. 49-60.

Erdmann, M. 1984. "On Motion Planning With Uncertainty", S.M. dissertation, MIT Department of Electrical Enginering and Computer Science, also AI-TR-810, MIT Artificial Intelligence Laboratory.

Erdmann, M. 1986. "Using Backprojections for Fine Motion Planning with Uncertainty", *International Journal of Robotics Research* 5(1), pp. 19-45.

Hogan, N. 1984. "Impedance Control of Industrial Robots", *Robotics and Computer-Integrated Manufacturing*, Vol. 1, No. 1, pp. 97-113.

Lozano-Pérez, T. 1983. "Spatial Planning: A Configuration Space Approach", *IEEE Transactions on Computers* 32(2), pp. 108-120.

Lozano-Pérez, T., Mason, M., and Taylor, R. 1984. "Automatic Synthesis of Fine-Motion Strategies for Robots", *International Journal of Robotics Research* 3(1), pp. 3-24.

Salisbury, J.K. 1980. "Active Stiffness Control of a Manipulator in Cartesian Coordinates", *IEEE Conference on Decision and Control*, Albuquerque, pp. 95-100.

Whitney, D. 1977. "Force Feedback Control of Manipulator Fine Motions", *ASME Journal of Dynamic Systems, Measurement, and Control* 99, pp. 91-97.

Whitney, D. 1985. "Historical Perspective and State of the Art in Robot Force Control", *IEEE International Conference on Robotics and Automation*, St. Louis, pp. 262-268.