Insertions Using Geometric Analysis and Hybrid Force-Position Control on a PUMA 560 with VAL II

David R. Strip
Intelligent Machine Principles Division 1411
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185

Abstract

Automatic programming of insertions is an essential step in achieving a truly flexible manufacturing environment. We present techniques based on active compliance implemented with hybrid force-position control capable of inserting a wide variety of shaped pegs. These techniques provide a significant step towards an automatically programmed flexible manufacturing environment.

I. Introduction

It will be necessary to reduce the programming difficulty of key tasks before robots can be conveniently used to perform assembly operations in truly flexible manufacturing operations. One of these critical operations is insertion, exemplified by the familiar "peg-in-the-hole" problem. Much has been written about solving the case of a chamfered round peg in a round hole. Little is known about solving this problem for more complex shapes, let alone threaded or bayonet insertions. In our work we have developed a general approach to oriented insertions that uses geometric properties of the object to control the behavior of a hybrid force-position controlled robot.

Mason introduced a model for position and force control for manipulators . In this model the degrees of freedom of a manipulator are partitioned into orthogonal subspaces representing the force controlled and the position controlled motions of the manipulator. This model provides a concise means of describing complex tasks, although in some cases the description is difficult to interpret. Raibert and Craig implemented a controller based on Mason's model and performed some experiments within the capability of a two degree of freedom manipulator [Raibert and Craig, 1981]. In our work we have developed a means of hybrid force-position control for a PUMA 560 using the VAL II controller. Our technique allows six dimensional subspace partitioning into force and position controlled subspaces. The current implementation is restricted to subspace components being associated with the cartesian axes of the tool frame. Our implementation extends Mason's model in that it provides a "guarded move" [Will, 1975] capability for both force and position constrained movements. In this paper we describe how we implemented hybrid force-position control and how we applied it to performing force-directed oriented insertions based on geometric constraints.

II. Implementation of Hybrid Force-Position Control using VAL II

The relevant portion of the Sandia Intelligent Robotic Assembly System (SIRAS) is comprised of a PUMA 560 six degree of freedom manipulator equipped with an Astek (now Barry Wright Corp.) FS6-120A 6-axis force-torque sensing wrist, an unmodified Unimation VAL II controller, a PDP 11/73 arm monitor, and a DEC microVax II task control computer. All user interaction is through the microVax in SCHEME, a dialect of LISP. The microVax communicates with the PDP 11/73 monitor which handles all communications to and from the Unimation controller. The arm monitor also provides the interface between the force sensing wrist and the Unimation controller.

The VAL II language includes an ALTER mode in which the controller polls the ALTER port every 28 ms (the basic timing cycle of the controller) for a set of translational and rotational offsets for the tool frame from the nominal position dictated by the current movement command. This mode continues until an END ALTER command is received. These offsets can be either cumulative or not, causing the manipulator to act as a dashpot or a spring, respectively. Our approach to hybrid force-position control was to implement a program on the arm monitor that calls the ALTER program on the Unimation controller and provides cumulative offsets to the ALTER port based on readings from the force sensing wrist and the parameters from the SCHEME command. The format of the SCHEME command is

(MCOMPLY GAIN BIAS THRESHOLD CONSTRAINT)

where GAIN, BIAS, THRESHOLD, and CONSTRAINT are 1 × 6 vectors. The arm monitor interprets this command to mean "move for the next time interval at speed=force × gain + bias (where these terms are multiplied on a component by component basis, with one component for each translation and rotation about the tool frame axes). If the absolute value of any force component exceeds its threshold or if absolute value of the cumulative

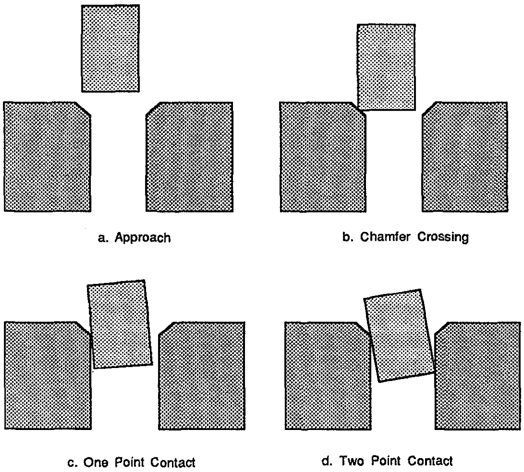


Figure 1: Stages in Insertion

movement exceeds the constraint, end alter and return a completion signal." In our implementation a 0 value for any component in the threshold or constraint vectors implies unlimited threshold or constraint.

This implementation provides a full six degrees of freedom of hybrid force-position control. It assumes, however, that forces and torques can only affect translations and rotations about the axis with which the force or torque is associated. A more general implementation in which the gain vector is replaced by a full 6×6 (accommodation) matrix would allow forces and torques to have effects off their natural axis. This more general implementation would allow solutions such as Starr's edge following, which was also based on the VAL ALTER command [Starr, 1986].

III. Application to the Peg in the Hole Problem

Whitney provides an analysis of the forces and torques encountered during the various phases of the insertion of a round peg into a chamfered hole [Whitney, 1982]. Whitney's analysis provides the means for establishing the design parameters of a remote center compliance (RCC), a device for providing passive compliance on an otherwise rigid manipulator.

A program for performing the peg-in-the-hole task was written using the hybrid force-position control command described above. With the peg positioned above the hole (Figure 1a) by means of a vision system, the arm is given the command

```
(MCOMPLY (1 1 1 .01 .01 .01) (0 0 10 0 0 0)
(0 0 30 0 0 0)(0 0 0 0 0 0))
```

The force sensor and the tool frame origins are both translated to the center of the bottom of the peg. This set of gains and biases commands the peg to move at a nominal speed of 10mm/sec in the positive z-direction only. On encountering the chamfer (Figure 1b) the sensor will see x- and y-forces (and in practice also small torques) that when multiplied by the gains will cause the peg to translate towards the center of the hole. As the peg drops into the hole (Figure 1c) any binding will create a z-force which, when multiplied by the gain, will slow the insertion rate while the associated torques about the x- and y-axes will cause the peg z-axis to tilt into alignment with the axis of the hole. When the peg hits the bottom the z-force should build up to -10 Newtons, which, when multiplied by the gain, would negate the bias and cause the arm to stop moving without exceeding the threshold. In a well-behaved system the program's stopping criterion would not be reached and the arm would appear to be "hung." Because the VAL II controller only samples the ALTER port every 28 ms and the force sensing wrist samples at 16 ms intervals there can be a considerable lag between the time a force measurement is made and the time it impacts the arm movement. This time delay combined with the stiffness of the arm requires that the z-threshold in our demonstration be 30 Newtons, although a threshold of less than 10 Newtons would be required to achieve the stopping criterion in a well behaved system. This threshold value always succeeds in stopping the system, contrary to intuition, although it does occasionally allow the peg to "bounce" one or two times at the bottom of the hole when the z-velocity is such that it hits bottom with greater than 10 Newtons force, which causes a negative z-velocity, but less than the 30 Newtons stopping threshold. These characteristics of the implementation demonstrate the limitations of implementing force controlled manipulation using commercial parts linked by software. Because the manipulator controller and the force sensor are not synchronized and operate at different sampling rates, the time lag from sensing to action is a random variable. The stiffness of the arm and workpiece are such that unless one is willing to perform the task at extremely low rates of movement (under 1mm per second z-travel), it is not possible to analytically develop the parameters for the MCOMPLY command given an analysis of the problem. An additional limitation in applying this force-position control technique is the inability to rotate the reference frame of the force sensor. While the reference frame can be translated to a new location, it cannot be rotated. Since the implementation is constrained to programs with independent effects on all the axes, an ability to rotate the reference frame would allow the solution of problems that can be represented by orthogonal force-position programs, but which are not aligned with the natural axes of the force sensor reference frame.

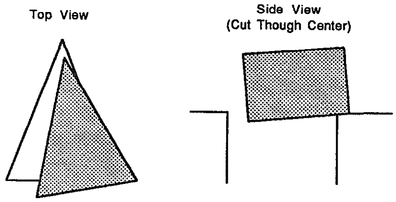
IV. Unchamfered and Oriented Insertions

RCCs provide a practical means of performing insertions using a single robotic motion and without the use of precision jigs. It does not appear to be practical (or, in some cases, possible) to generalize the RCC design to allow insertion, in a single robotic motion, of unchamfered round pegs or pegs which are not round in cross section and therefore require orientation.

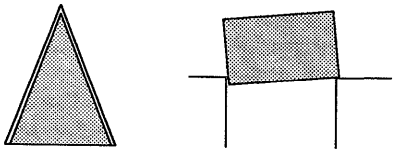
A multi-stage strategy for performing oriented insertions was developed based on observation of human strategy for the same task. The underlying principle of using constraints imposed by the geometry of the object is shared with the approach used by Shariat, Coifeet, and Fournier to plan a strategy for an inaccurate, flexible robot [Shariat *et al.*, 1985]. This strategy is based on the assumption that the objects being inserted are "large" in comparison to the scale of error in the vision and manipulator systems. If this does not hold, it would not be possible to determine orientation information about the object from the vision system and manipulation would require an entirely different approach.

The multi-stage strategy consists of three steps; approach, orientation, and insertion. In the approach step the "peg" is brought into contact with the block containing the hole. In making this approach the peg is oriented to match the orientation of the hole within the limits of the vision system and the manipulator. (In a factory environment these locations may be known through the use of jigs. The adaptability of the technique, however, would allow the use of fairly low precision (and thus low cost) jigs in contrast to traditional high tolerance jigging techniques.) In our laboratory this amounts to about 1/4 inch linear displacement and 4 or 5 degrees angular displacement. In bringing the parts into contact the peg is deliberately shifted to insure a "target point" of the object is over the hole. For an object like the isosceles triangle shown in Figure 2, this point is the corner with the sharpest angle. (There is more discussion of how to select this point later.) The object is then tilted into the hole as shown in Figure 2a. This ends the approach stage. The first stage of the insertion does not require compliance, active or passive, although force sensing may be used to simplify the programming of the approach since contact forces may be used to detect that the peg has contacted the block.

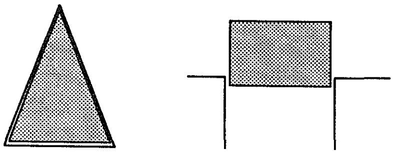
The orientation stage is broken down into two parts. During the first part the target point is driven towards its matching point in the hole. If the target point has been properly selected, active or passive compliance combined with the appropriate manipulator motion will move the point of the peg into the corner of the hole. The peg will rotate to approximately the correct orientation due to the torques on the peg from the contact with the side of the hole (Figure 2b). In the second part of the orientation step the peg is rotated about a line through the target point and perpendicular to the direction of travel into the corner. This rotation will return the peg to an approximately vertical position (Figure 2c). If this rotation is made compliantly with constant force maintained between the target point of the peg and the corner of the hole, the lower edge of the peg will (in general) meet the edges of the hole at an angle, introducing a torque on the peg that will further correct its orientation.



a. Approach



b. Orientation - Stage 1



c. Orientation - Stage 2

Figure 2: Stages in Oriented Insertion

The insertion stage for orientable objects is the same as the final stage of insertion for round pegs as analyzed by Whitney and described in Section III. [Whitney, 1982].

A program using this strategy was implemented in our laboratory using the hybrid force-position control technique described above. The use of hybrid force-position control instead of passive compliance encourages us to learn about the forces involved in the insertion process and leads to a more general understanding than we might get using passive compliance devices. In addition, the error of our vision system (particularly with respect to determining orientation angle) is greater than the travel limits of commercial RCCs known to us and therefore precluded their use without some form of active compliance. Figure 3 shows the variety of peg shapes successfully inserted with this insertion program. By using hybrid force-position control a single program can be used to perform insertion of a wide variety of shapes. While conceptually similar to the approach in [Shariat *et al.*, 1985], the use of hybrid force-position control (even for an inaccurate flexible robot) considerably simplifies the implementation of the insertion technique and allows an identical program to insert a variety of shapes. Hybrid force-position control also allows the programmer control over the forces exerted on the workpieces, which can be critical when manipulating fragile objects.

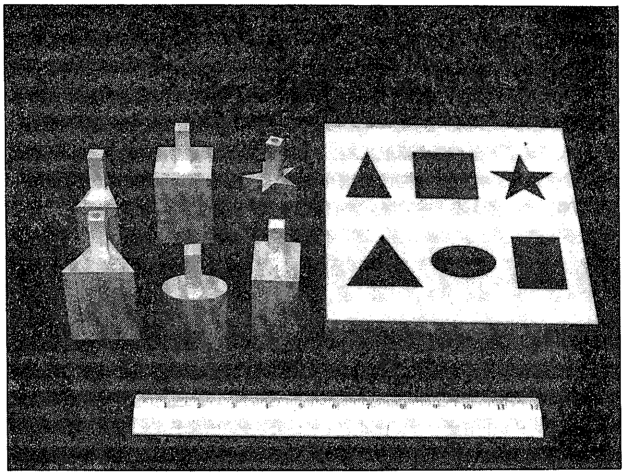


Figure 3: Shapes Successfully Inserted

V. Selecting the Target Point

In the section above we referred to a "target point" that was central to all the stages but did not explain how to select such a point. Humans have an intuitive understanding that allows them to select this point without conscious thought. In order to have our robot select these points, we need to understand their properties. The key role of the target point is to induce torques and forces on the peg from its contact with the edge of the hole. These torques and forces should be such that when the peg is rotated or translated to zero out these forces and torques as the target point approaches its corner in the hole, the orientation of the peg should move to alignment with the hole. For convex polyhedral objects the point with the smallest interior angle appears generally to be a good choice. A good target point for a convex object with a smooth boundary is the point with the smallest radius of curvature.

VI. Future Work

We have several directions in which we are taking these results. We would like to be able to automatically determine the "target point" and insertion strategy from a description of the object to be inserted. We want to be able to prove that a given technique is necessary and/or sufficient for performing insertions. Automated assembly will require the insertion of threaded and bayonet parts. Small parts (small relative to the scale of the manipulator and vision system accuracy) and parts with extremely tight tolerances will have to handled. We are working in each of these areas to develop a complete capability for automated insertions.

In related research we are examining the role of active versus passive compliance and the requirements for controllers to provide hybrid force-position control. We are also examining the potential for fine control at the end of the manipulator to perform the small movements required in insertions, rather than relying on moving the entire manipulator. In this vein we have mounted a Salisbury hand on a PUMA 560 robot, operating the hand in hybrid force-position control using a controller built in our lab. We are developing a dextrous end-effector based on our experiences with the Salisbury hand that will provide for fine movements, but with reduced degrees of freedom relative to the Salisbury hand to simplify control.

Acknowledgements

I would like to acknowledge the efforts of Greg Starr, who developed the concept for hybrid force-position control using the VAL-II ALTER command and provided the first implementation, and Bill Davidson, who implemented the improved version of the hybrid force-position control used in our lab.

References

[Mason, 1981] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Trans. Systems, Man, and Cybernetics,* Vol. SMC-11, No. 6, June 1981.

[Raibert and Craig, 1981] M. H. Raibert and J. J. Craig, "Hybrid Force-Position Control of Manipulators," *Trans. ASME J. Dynamic Systems, Measurement, and Control,* Vol. 102, No. 2, June, 1981.

[Shariat *et al.,* 1985] B. Shariat, P. Coifeet, and A. Fournier, "A Strategy to Achieve an Assembly by Means of an Inaccurate, Flexible Robot," *Computing Techniques for Robots,* Chapman and Hall, N.Y., N.Y., 1985.

[Starr, 1986] G. P. Starr, "Edge Following with a PUMA 560 Manipulator Using Val-II," *Proc. IEEE Int'l Conf. Robotics and Automation,* San Francisco, CA., April, 1986.

[Whitney, 1982] D. E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Trans. ASME J. Dynamic Systems, Measurement, and Control,,* Vol 104, No.1, March, 1982.

[Will, 1975] P. M. Will, and D. D. Grossman, "An Experimental System for Computer Controlled Mechanical Assembly," *IEEE Trans. Comput.,* Vol. C-24, No. 9, Sept., 1975.