In the Proceedings of the 1995 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Pittsburgh, PA August 5-9, 1995, Vol. 3, pp. 234-240

Vision and Force Driven Sensorimotor Primitives for Robotic Assembly Skills

J. Daniel Morrow† Bradley J. Nelson† Pradeep K. Khosla‡

†Robotics Ph.D. Program
‡Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Integrating sensors into robot systems is an important step towards increasing the flexibility of robotic manufacturing systems. Current sensor integration is largely task-specific which hinders flexibility. We are developing a sensorimotor command layer that encapsulates useful combinations of sensing and action which can be applied to many tasks within a domain. The sensorimotor commands provide a higher-level in which to terminate task strategy plans, which eases the development of sensor-driven robot programs. This paper reports on the development of both force and vision driven commands which are successfully applied to two different connector insertion experiments.

1 Introduction

Creating sensor-based robot programs continues to be a formidable challenge. Two contributing factors are programming difficulty and lack of sensor integration. Addressing these problems simultaneously is important because they are coupled: introducing sensors exacerbates the programming problem by increasing its complexity. We propose the development of a sensorimotor layer which bridges the robot and sensor spaces with the task space. In this paper, we introduce the ideas behind sensorimotor primitive (SMP) development and provide examples of both force and vision driven SMP's. In addition, some of these SMP's are implemented and used to construct sensor-based control strategies for executing two different connector insertions (D and BNC).

Our goal is to build a richer set of command primitives which effectively integrate sensing into the command set for a particular class of tasks (e.g. rigid-body assembly). The goal is to provide the task programmer with higher-level commands which incorporate sensing and are relevant to the task domain. The critical benefits of sensor integration are 1) hiding low-level details of processing sensor information, and 2) embedding generic task domain knowledge into the command set. The first goal is achieved by creating small, reconfigurable modules which process

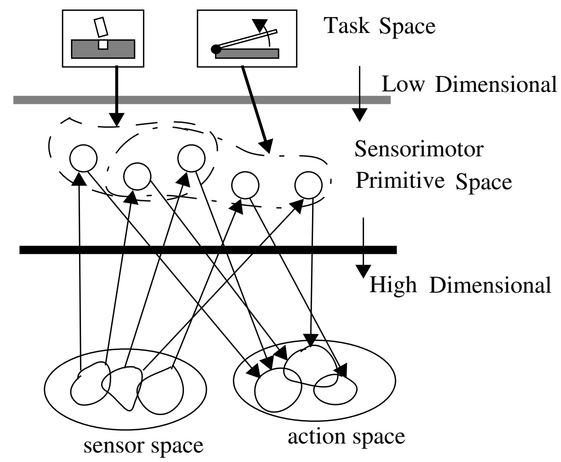


Figure 1: Sensorimotor Space

and use sensor information; this allows us to leverage the application of a sensor since it is encapsulated. The second goal is achieved when the models used to interpret sensor information are applicable to many tasks within a domain. Whenever a sensorimotor primitive is developed, some knowledge about the task (and about the domain if the knowledge is sufficiently general) is encapsulated as well. The problem is identifying common models for sensor interpretation which apply to a variety of related tasks. To the extent that these models are applicable only to very similar tasks, the task domains will be exceedingly small for which the command set is applicable. The challenge is to construct a sensor-integrated command set with enough embedded knowledge to reduce the difficulty of the programming task while retaining enough generality to have wide applicability.

1.1 Related Work

Many researchers [16][19] refer to skill libraries or task-achieving behaviors as a source of robust, skill-achieving programs. This postpones (but does not remove) the very difficult issue of how to synthesize such skill libraries. The sensorimotor layer is a direct effort to ease the programming of robust, sensor-based skills.

Other researchers have suggested robot control primitives. Lyons [10] has developed a theory of computation for sensor-based robotics; his *robot schema* computational element is very similar to our Chimera reconfigurable module [17]. Brockett [1] suggests a postscript-type programming language for robotics in which task strategies can be described independently of a particular robot system. Deno et al [3] discuss control primitives which are inspired by the hierarchical nature of the neuromuscular system, but these do not have a strong connection to the task. Paetsch and von Wichert [14] apply a set of heuristic behaviors in parallel to perform peg insertion with a dextrous hand. Smithers and Malcolm [16] suggest behavior-based assembly as an approach in which uncertainty is resolved at run-time and not during planning, but they do not address the issue of behavior synthesis. Most of these approaches to primitives (except [14]) are either task-specific or robot-centered. We are building on portions of this past work to make a stronger and more general connection of sensor-based control primitives to a task domain.

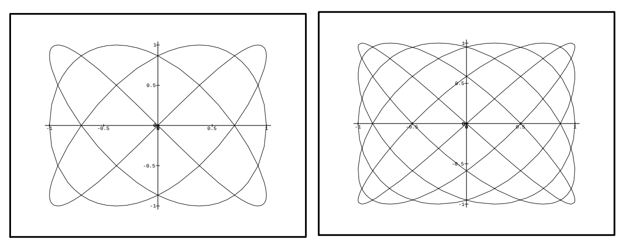
Planning robot motions based on geometric models [8] has been pursued as a method of task-level programming which reduces the programming burden. A problem with this method is that resulting strategies often fail because of inevitable errors in the task model used for planning. We believe, like Smithers and Malcolm [16], that uncertainty should be resolved at run-time, not during planning. Morris and Haynes [11] have argued that geometric models do not contain enough information about how to perform a task. They argue for using the contact constraints of the assembly task as key indicators for guiding task strategies. Similarly, we base our sensor-driven primitives on task constraints.

Much work in using force feedback has centered on detailed contact models [20]. Schimmels and Peshkin [15] have synthesized admittance matrices for particular tasks. Strip [18] has developed some general methods for peg insertion based on contact models. Donald [4] developed methods to derive plans based on error detection and recovery which are guaranteed to either succeed or recognizably fail. Erdmann [5] has investigated task information requirements through abstract sensor design. Castano and Hutchinson [6] have proposed task-based visual servoing in which virtual constraints, based on the task, are maintained. Canny and Goldberg [2] have been exploring RISC (reduced intricacy in sensing and control) robotics in which simple sensing and action elements are coupled. Many of these approaches focus on developing sensor use strategies for a particular task. We are trying to generalize sensor use for a task domain by building sensor-driven commands which are based on common task constraints in both vision and force.

2 Trajectory Primitives

Trajectory primitives are encapsulations of robot trajectory specifications. We have developed three trajectory primitives which are used in our experiments. The *movedx* primitive applies a cartesian velocity over time to achieve the specified cartesian differential motion. The *ldither* (*rdither*) primitive implements a linear (rotary) sinusoidal velocity signal at the specified frequency for the specified number of cycles. This is useful during assembly operations to locally explore.

Complex trajectories can be specified by combining trajectory primitives. For example, combining sinusoidal dithers in orthogonal directions can be used to implement an "exploration" of an area; the resulting position patterns are called *Lissajous* figures. In order to densely cover an area, the frequency ratio (n>1) between orthogonal dithers should be selected as (N+1)/N, where N is the number of cycles (of the smaller frequency sine wave) before the Lissajous pattern repeats. Figure 2 shows the Lissajous figures for two orthogonal dither signals with different values of the frequency ratio, n. Note that the positional space is well-covered by these patterns. A smaller n (closer to 1) provides more dense coverage but requires more cycles (and hence longer time) to execute.



(a) 3 cycles n=1.333 (b) 5 cycles n=1.2

Figure 2: Lissajous patterns

3 Sensorimotor Primitives

A sensorimotor primitive is a parameterized encapsulation of sensing and action which can be used to build task strategies or skills. A skill is a particular parameterized solution to a specific task (e.g. peg in hole) and is composed of sensorimotor primitives. In order to develop sensorimotor primitives, the common element(s) which relate tasks in the domain must be identified. For assembly tasks, force-driven primitives which provide for the acquisition, maintenance, and detection of different types of contact constraints are useful. Likewise, vision-driven primitives can be used to enforce positioning constraints which can be sensed on the image plane. We rely on these constraints in both the force and vision spaces to provide guidelines for sensor-driven commands.

3.1 Vision-Driven Primitives

The vision-driven primitives are based on visual servoing techniques. An image-based visual servoing approach is used, rather than a position-based approach, to avoid calculating the inverse perspective mapping of the scene at each sampling period. Thus, we must provide reference inputs to our visual servoing system in feature coordinates. To do this, desired 3D object positions must be mapped into image coordinates using a simple perspective projection model of the particular visual sensor. These primitives enforce positioning constraints on the task using poorly-calibrated camera/robot systems; errors on the image plane are used to drive the manipulator. A complete description of this visual servoing method can be found in [13].

Vision primitives are used to enforce positioning constraints on the image plane which are relevant to the task. The effective bridge between task space and robot/sensor is constructed by enforcing key task positioning constraints which can be sensed on the image plane by tracking and controlling a finite number of critical points in the task.

Image plane translation. A fundamental vision primitive is the resolution of image-plane errors through translation commands; this enforces "translation" constraints in the image plane. Castano and Hutchinson [6] have proposed a similar method to perform tasks. We use this primitive with a two-camera arrangement to enforce 3 DOF position constraints. In addition, the primitive is written so that individual combinations of axes can be controlled. This enables the flexibility needed to de-couple translation commands for certain tasks (e.g. grasping along a specific approach direction). This primitive was implemented and used in the connector insertion strategies.

Image plane rotation. Another common positioning primitive is to align lines in the image plane. Insertions, for example, can be very sensitive to errors in the insertion axis alignment. An edge-detection algorithm can robustly extract edges from an image and a primitive can use this information along with an approximate task model to align edges.

Fixed point rotation. Rotation about the normal of the image plane causes a translation of all points not on the optical axis. Therefore, one primitive involves selecting a particular point fixed with respect to the end-effector which is relevant to the task and maintaining its position (in the image) during a rotation.

Visual grasping. One primitive which can be very useful is a vision-guided primitive from a camera mounted on the gripper -- so-called "eye-in-hand" primitives. This can be used to align the gripper with cross-sections which are extracted from binary vision images. Automatic centering and obstacle avoidance could be implemented with such a primitive.

3.2 Force-Driven Primitives

Guarded move. This primitive is the common guarded move in which straight-line motion is terminated by contact. The contact is detected by a force threshold in the direction of motion. The basic constraint model is a transition from free space (and complete motion freedom) to a point/plane contact where one direction DOF has been removed.

Sticking move. The "stick" primitive involves the transition, upon contact, to a "maintenance" velocity which will maintain contact with a surface when used with a damping controller. In addition, the cartesian position is monitored in the direction of the maintenance velocity, and if the displacement exceeds a specified threshold, the primitive detects this and terminates. This prevents the robot from continuing to move when contact has been lost, and encapsulates the maintenance of point/plane contact with loss detection.

Accommodation. The sub-matrices of a 6x6 damping matrix which provides accommodation control can be viewed as sensorimotor primitives. The most common one is linear accommodation: complying to linear forces by performing translations. A sensorimotor primitive which introduces angular accommodation in response to torques and forces implements a remote-center-of-compliance [20] useful for peg insertion tasks.

Correlation. Active sensing primitives, which use the commanded action to process the sensor signal, are effective ways of extracting information from biased and noisy sensor signals [7]. We have employed a correlation technique to detect when the reference command is perturbed by the damping controller, indicating the presence of a motion constraint. The correlation (C) is computed with the following equation:

$$C = \frac{\left(\sum_{i=0}^{N} f\left(\frac{i2\pi}{N}\right) g\left(\frac{i2\pi}{N}\right)\right) N}{\sum_{i=0}^{N} \left| f\left(\frac{i2\pi}{N}\right) \right| \sum_{i=0}^{N} \left| g\left(\frac{i2\pi}{N}\right) \right|} \quad (1)$$

For two fully correlated sinusoidal signals, the correlation value is $\pi^2/8$. Because the correlation technique is based on phase differences, the normalization is required to compensate for magnitude changes in the signals which affect the computed value.The correlation value is tested against a threshold and an event is triggered when the correlation drops below the threshold. The full development of this primitive is discussed in [12].

4 Experimental Results

Our experimental results are based on two connector insertions: a 25-pin D-shell connector and a BNC connector. Figure 3 shows diagrams of the part of each connector held by the gripper. Both connectors were stably grasped by our pneumatic, two-fingered gripper; no special fingers were constructed.

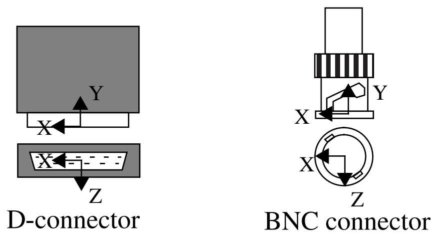


Figure 3: Connector Diagrams

The strategies are implemented as finite-state machines (FSM). Figure 4 shows an example FSM strategy in task-space which accesses the primitives in the sensorimotor space. The connector insertion strategies (Figure 5 and Figure 6) are shown in the task space with the primitives explicitly shown in the FSM. Given the small scale of the contact, we cannot reasonably derive strategies based on detailed contact-state analyses. Instead, heuristic strategies were developed based on the available command set and sensing. The strategies are based on the available command primitives (some sensor-driven, some not) and are implemented as finite-state machines (FSM). Although the different connector geometries lead to very different strategies, the same command primitives can be used to implement these strategies.

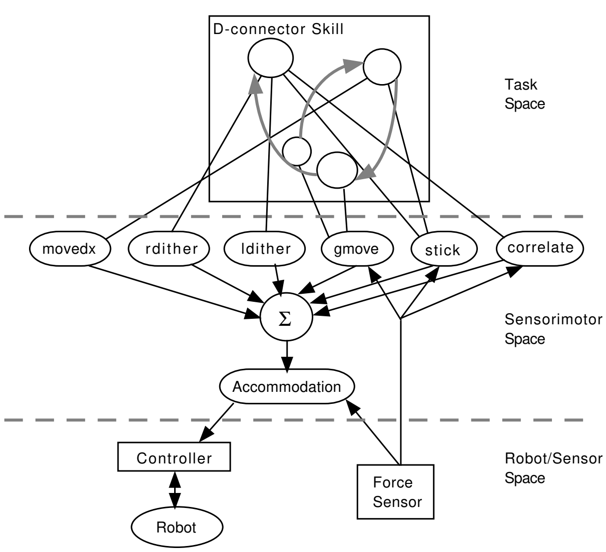


Figure 4: Finite-State Machine Strategy

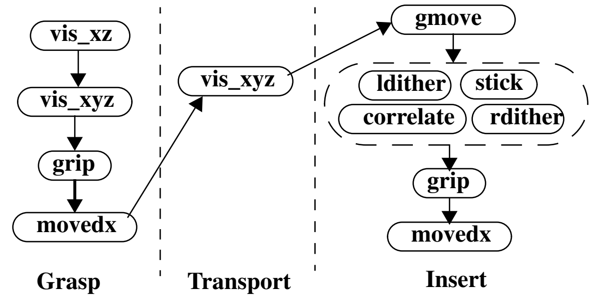


Figure 5: D-connector Insertion Strategy

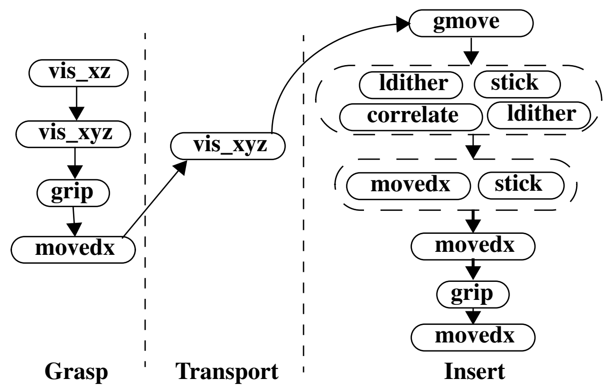


Figure 6: BNC Connector Insertion Strategy

In Figure 5 and Figure 6, each of the "bubbles" in the FSM is a Chimera module implementation of a real-time computing process. The **vis_** modules implement visual servoing primitives; for example, **vis_xz** implements visual servoing along the x and z axes. The **grip** module operates the gripper. The other modules (**gmove, stick, movedx, ldither, rdither**) are described in the force-driven or trajectory primitive sections. The task strategy implemented by primitives results in a command velocity, V_{cmd}, which is perturbed by the accommodation controller to permit contact. The perturbed velocity, V_{ref}, is used to generate joint setpoints for the robot joint controller. All of the experimental results are shown as plots of V_{ref}.

For each connector insertion task (Figure 5 and Figure 6) the strategy involves three phases: 1) grasp the connector, 2) transport to the mating connector, and 3) perform the insertion. The grasp and transport steps are dominated by vision-feedback; the insertion step is dominated by force feedback. The first step, grasping, relies on approximate angular alignment of the connector axes (X, Z) with the camera optical axes. Visual setpoints are identified in the images and controlled through visual feedback. The transport step also involves using visual feedback to position the grasped connectors above the mating connector for insertion. The insertion step is different for each task because of their different geometries; however, these two different strategies are implemented with the same set of primitives. For the D-connector, the insertion begins with a guarded move to make contact with the top of the connector. This is followed by a sticking move (to maintain the contact) and a mixture of rotational and linear sinusoidal dithering (at different frequencies), and correlation monitoring of the linear dithering. The dithering introduces enough variation in the command to resolve small uncertainties left over from initial positioning. The correlation of the commanded and force-perturbed reference velocities provides a means to reliably detect when the connector has seated. This "success-detection" method does not rely on attaining absolute position goals, but rather on attaining and detecting a motion constraint. The strategy for the BNC connector begins similarly, with a guarded move. This is followed by a sticking move (to maintain contact) and a mixture of two linear dithers to implement a Lissajous pattern exploration around the insertion axis. We found this to be necessary as the vision primitive was not able to reduce positioning errors enough to guarantee insertion using a (nominal) straight-line move. Again, correlation is used to detect when the connector "seated." For the BNC, however, there is an additional step: the bayonet shaft stubs must be mated. This can be performed with a 180 degree rotation and terminated using the stick primitive to detect when the connector advances along the insertion axis. Finally, a 90 degree rotation about insertion axis locks the bayonet connector.

Figure 8 and Figure 7 show cartesian velocity traces for experimental trials for each connector insertion task. The three stages of the tasks are labelled in each plot and the breakpoints for different stages are sensor-driven. The use of both force and vision in performing these tasks significantly increases their robustness. Earlier D-connector results using only force feedback [12] required more stringent initial positioning requirements and ignored the connector grasping phase of the task. Visual feedback makes the (final) insertion step very reliable since position errors are significantly reduced.

The grasp stage is essentially the same for both connectors. Figure 9 shows a vision-guided grasp of a connector with three distinct stages: the approach in X and Z, followed by the approach along -Y, followed by the depart move along +Y. Continuing to visually servo the X and Z directions during the approach along -Y is important to compensate for calibration errors; this is clearly shown in Figure 9 where the V_x and V_z are non-zero toward the end of the -Y approach move. Figure 9 also clearly shows when the part is grasped and the final depart move. The transport stage is very similar except that it does not have the grasp and depart phases.

Figure 10 shows a close-up view of the D connector insertion results from Figure 7. The key parts of the strategy are labelled in the plot. There is a 0.5s delay during the grip primitive to allow the fingers time to open. Figure 11 shows a close-up view of the BNC insertion stage results from Figure 8. The key parts of the coax insertion strategy are labelled in the plot. In this case the first insertion stage, which is detected through correlation, was achieved almost immediately. In order to compute a proper correlation value, at least 1 full cycle of dithering is completed before the correlation primitive generates a valid correlation value (hence the dithering will not be terminated for at least one cycle). Clearly visible on the plot is the movement along -Y towards the end of the first rotation. This is detected by the *stick* primitive and signals the transition to the last rotation state which locks the bayonet connector.

Both of these strategies were easily described with sensorimotor primitives and were successful in repeatedly performing the tasks in spite of imprecisely calibrated

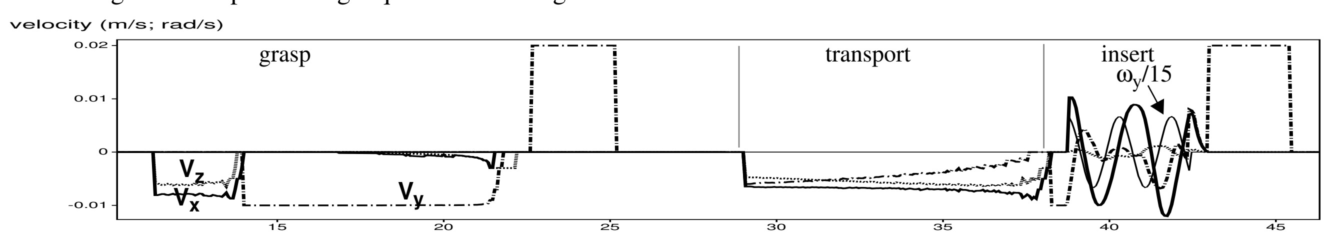


Figure 7: D-connector Insertion Results

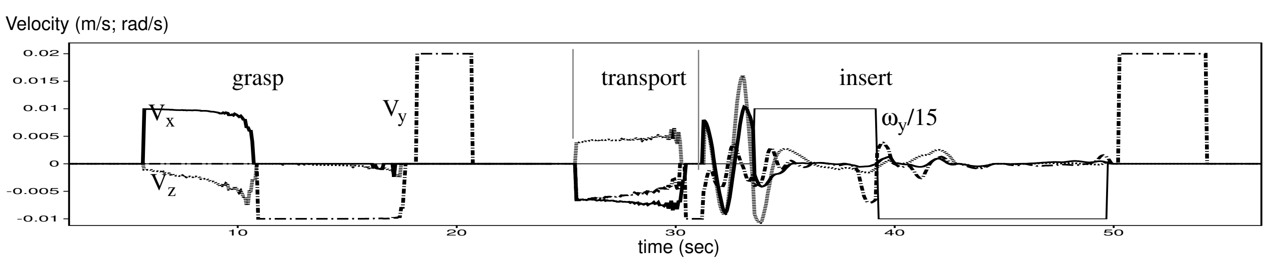


Figure 8: BNC Connector Insertion Results

sensors and lack of firm fixturing. The most crucial error, insertion axis alignment, was usually small from the beginning position of the parts (the connector was grasped standing, not lying on its side). Nonetheless, the rapid motion of the pneumatic gripper fingers sometimes introduced some skew in the insertion axis angle. Fortunately, the guarded move and subsequent dithering usually re-aligned the axis so that insertion proceeded without error. However, the tendency of the connector to rotate about the task Z-axis, due to only friction coupling, made the correlation detection threshold more difficult to set. If the part is firmly fixtured, one expects the correlation of command and reference signals to drop sharply when motion constraints are encountered (e.g. when seating the connector). However, our lack of firm fixturing resulted in significant motion of the task box during these stages of the task.A larger correlation threshold had to be set in order to reliably detect the seating stage. More precise fixturing and tooling would alleviate these problems considerably, but our strategies were able to succeed in spite of them. Besides the common guarded move, the *stick* primitive, which encapsulates contact maintenance, and the *correlation* primitive, which detects the difference in commanded and reference velocity signals, proved very useful in both tasks. The *stick* command was used differently in the two tasks. In the D-connector, it was used to detect an error condition if contact was lost; in the BNC, it was used to detect when the bayonet “mated” with connector shaft stubs.

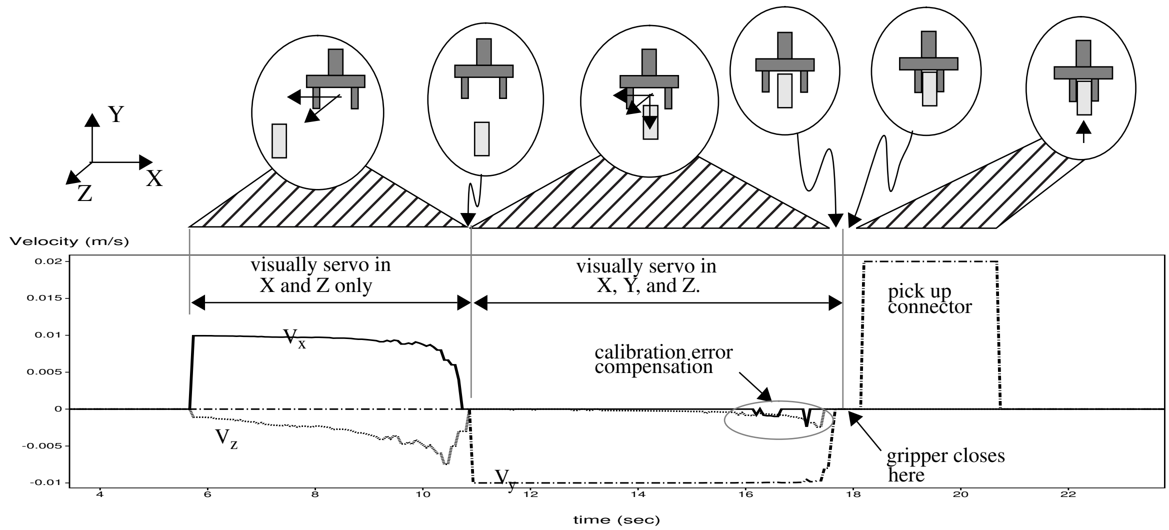


Figure 9: Vision-Guided Grasp

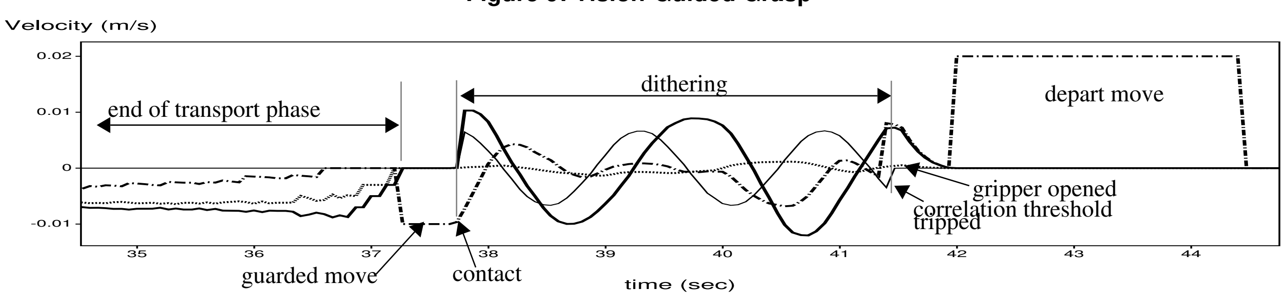


Figure 10: D-connector Insertion Stage

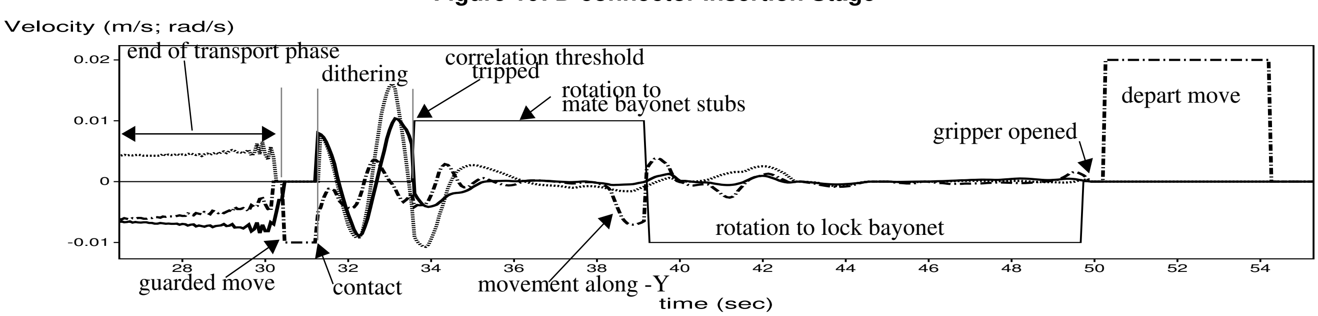


Figure 11: BNC coax Insertion Stage

5 Conclusions

We have proposed a sensorimotor layer for easing the programming of sensor-based skills through sensor-integrated, task-relevant commands. The main idea is to encapsulate common uses of sensing and action for re-use on similar tasks within a domain. We have outlined several sensorimotor primitives in this paper and applied those primitives to sensor-based skills for performing connector insertions. Results from successful experimental trials were presented to show the feasibility of the ideas.

Our initial goal of encapsulating sensor applications so they can be re-applied to different tasks was successful. However, the set of sensorimotor primitives is still very small, and a richer set must be developed and applied to a larger class of tasks. One problem with these connector tasks is the scale of contact is so small that it precludes detecting contact states with the force sensor (or by vision). These types of tasks usually require heuristic strategies, like those employed here, which are not guaranteed to succeed. In order to develop additional sensorimotor primitives, we need to identify larger-scale tasks (or improve our sensors) so that key task events can be adequately sensed. We intend to continue exploring the use of contact constraints for guiding force-driven primitive development. For vision primitives, we will explore the use of different feature types (e.g. edges) in order to implement some of the primitives discussed earlier.

One of the significant research issues in this area is how to achieve generalization of sensor application. We are approaching it from a task-constraint point of view: both contact (force) constraints and visual constraints. To avoid making the primitives *task-specific*, constraints must be identified which are common across tasks within a domain. This, and other approaches to identifying task similarities, are the subjects of on-going research.

6 Acknowledgements

J.D. Morrow was supported in part by the Computational Science Graduate Fellowship Program of the Office of Scientific Computing in the Department of Energy. B. J. Nelson was supported in part by a National Defense Science and Engineering Graduate Fellowship through the U.S. Army Research Office through Grant Number DAAL03-91-0272 and by Sandia National Laboratories through Contract Number AC-3752D. We acknowledge Richard Voyles for helpful discussions. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

7 References

[1] R.W. Brockett, "On the computer control of movement," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, 24-29 April, 1988, pp. 534-540.

[2] J. Canny and K. Goldberg, "A RISC approach to robotics," *IEEE Robotics and Automation Magazine*, vol. 1, no. 1, March 1994, pp. 26-28.

[3] D.C. Deno, R.M. Murray, K.S.J. Pister, S.S. Sastry, "Control Primitives for Robot Systems," pp. 1866-1871, *Proc. of IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, 1990.

[4] B. Donald, "Planning multi-step error detection and recovery strategies," *Int. J. Rob. Res*., vol. 9, no. 1, Feb. 1990, pp. 3-60.

[5] M. Erdmann, "Understanding Action and Sensing by Designing Action-Based Sensors," IJRR Draft, Nov 30, 1993.

[6] A. Castano and S. Hutchinson "Visual Compliance: Task-Directed Visual Servo Control," *IEEE Trans. on Robotics and Automation*, vol. 10, no. 3, June 1994, pp. 334-342.

[7] S. Lee and H. Asada, "Assembly of Parts with Irregular Surfaces Using Active Force Sensing," pp. 2639-2644, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, 1994.

[8] T. Lozano-Perez, "Task Planning," Ch. 6, *Robot Motion: Planning and Control*, MIT Press, Cambridge, MA, 1982.

[9] T. Lozano-Perez, M.T. Mason, R.H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots", *Int. J. of Rob. Res.*, Vol 3, No 1, pp. 3-23, Spring, 1984.

[10] D. Lyons, "RS: A Formal Model of Distributed Computation For Sensory-Based Robot Control," COINS Tech. Rept. 86-43, University of Massachusetts at Amherst, Sept. 1986.

[11] G.H. Morris and L.S. Haynes, "Robotic Assembly by Constraints," pp. 1507-1515, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, 1987.

[12] J. D. Morrow and P.K. Khosla, "Sensorimotor Primitives for Robotic Assembly Skills," to appear in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995.

[13] B.J. Nelson, N.P. Papanikolopolous, and P.K. Khosla, Visual servoing for robotic assembly. *Visual Servoing - Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. ed. K. Hashimoto. River Edge, NJ: World Scientific Publishing Co. Pte. Ltd. pp. 139-164.

[14] W. Paetsch, and G. von Wichert, "Solving Insertion Tasks with a Multifingered Gripper by Fumbling," pp. 173-179, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, 1993.

[15] J. M. Schimmels and M.A. Peshkin, "Admittance Matrix Design for Force-Guided Assembly," *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 2, pp. 213-227, April, 1992.

[16] T. Smithers and C. Malcolm, "Programming Robotic Assembly in terms of Task Achieving Behavioural Modules," DAI Research Paper No. 417, University of Edinburgh, Edinburgh, Scotland, December 1988.

[17] D.B. Stewart, Schmitz, D.E. and Khosla, P.K. The Chimera II real-time operating system for advanced sensor-based control systems. *IEEE Trans. Sys., Man Cyber.* 22(6):1282-1295., 1992

[18] D. Strip, "Technology for Robotics Mechanical Assembly: Force-Directed Insertions," *AT&T Technical Journal*, Vol. 67, Issue 2, pp. 23-34, March/April 1988.

[19] W.O. Troxell, "A robotic assembly description language derived from task-achieving behaviors," *Proc. of Manufacturing International '90*, Atlanta, GA, 25-28 March 1990.

[20] D. Whitney, "A Survey of Manipulation and Assembly: Development of the Field and Open Research Issues," in *Robotics Science*, ed. M. Brady, MIT Press, 1989.