

Skill of Intelligent Robot

Kunikatsu Takase
Electrotechnical Laboratory
Tokyo, Japan

ABSTRACT

This paper reviews the development of the intelligent robot and separates skills from high level intelligence. It demonstrates that skills can be represented as virtual mechanisms programmed in software. Virtual mechanisms are defined by controlling both motion and force of a robot arm in a task related cartesian coordinate system. By adding monitoring, a skillful robot system can be built. The skill of an intelligent robot is accumulated in the form of task - particular knowledge that is a specification of the mechanism.

1. INTRODUCTION

Various intelligent robots have been developed by combining computer decision making with mechanical mechanisms, especially with robot arms. In this field the efforts have been mainly directed to developing high level intelligence system such as for solving puzzles, interpreting drawings or recognizing three-dimensional objects. There has been little study of low level intelligence relating to skill or dexterity of a robot arm. In order to make the motion of a robot arm smoother and more adaptive, a series of studies - kinematics, trajectory calculation, sensory feedback of a robot arm - were undertaken in the Artificial Intelligence Project at Stanford University. But in general we are at the dawn of the study of the skill of robot arms.

The practical study of assembly automation which started as one of the applications of intelligent robots is developing as an independent area with remarkable results. Reliable assembly systems are being released one after another. Although those are comparatively special-purpose low intelligence systems, their skills are far more dexterous than those of intelligent robots. It seems the increased skill of the intelligent robot could be obtained only through the integration of task-particular knowledge that is normally

obtained, for example, during the development of special purpose systems. The "Move" command in AL, for example, and its modification are not sufficient for specifying the dexterous actions of robot arms. "Composing virtual mechanisms by software" seems to be the best control structure for the purpose.

Recently the performance of arithmetic LSI has improved remarkably enabling us to carry out the exact calculation of dynamic models of a robot arm in the servo cycle, and to compose virtual mechanisms in software.

2. INTELLIGENT ROBOTS

In the beginning of 1970's several Hand-Eye systems had been developed at universities and research laboratories in the world. Those robotic systems consisted of a camera and arm with 6 degrees of freedom and computer. At Stanford University a robot which could successfully solve the "instant insanity" puzzle was developed [1]. In this puzzle four cubes with different colored faces must be stacked up so that no two similar color appear on any sides. A robot made at Hitachi could understand simple drawings and build block structures as specified in the drawings [2]. At the Electrotechnical Laboratory a robot was developed, which could insert a beam into a box with small clearance using visual feedback [3]. In these robotic system intelligence had been applied to recognizing a 3-dimensional pattern, understanding a drawing, or solving puzzles.

The author has been a visiting scientist of Electrical Engineering at Purdue University, West Lafayette, IN, since August 1978 until August 1979.

In order to realize computer control of an arm, several elements had been studied: arm design, servo system, transformation between joint coordinates and cartesian coordinates, computer interface, and robot operating system. Although a good deal of effort was devoted to the computer control of the arm, the arm only could perform "pick and place" operations, that is, to pick up an object, to transfer it and then to place it. Some attempts were made to make the a robot arm more dexterous, for example, by adding tactile sensors in order to grasp objects more dexterously. None of those attempts was, however, successful in developing a reliable elements to be incorporated in the robotic system.

It was in the Assembly Automation Systems developed at Stanford University (later referred to as the AL System) that an intelligent robot was first able to act in the practical world, emerging from the stage of "play" in the block world [5,6,7]. Assembly processes, such as parts mating and inserting, required adaptive object handling capabilities found only in a human workers were demonstrated. These tasks could not be performed by conventional hard automation machines.

In many cases of low volume assembly production programmability was the key factor to enhance the utility of the assembly system. In the AL system much attention paid to the design of task description language. AL offered the programming structure in which a user could describe an assembly procedure in terms of multilevel commands ranging from the simple "move" command to task-level commands [4]. High-level assembly statements would be expanded into the sequence of move-level commands in a general manner.

In an assembly system robot arms have to be skillful enough to be able to carry out parts mating and other adaptive motions. In the AL system, precise motions were controlled, based on trajectory calculation and the modification of motions based upon force feedback or touch feedback to adapt to the outer world. To integrate these functions into the system, software servoing was used, where a digital computer controlled each motor drive level directly. By using such a control scheme, the capability of the robot arm was remarkably improved.

3. PRACTICAL ASSEMBLY AUTOMATION

While general assembly automation was being studied at Stanford University, other practical

research was started at several other laboratories and companies. A system, which consisted of position controlled robots and various specially designed tools was developed at Kawasaki Unimate to assemble a gasoline engine. The work demonstrated a practical approach to assembly automation. Hitachi developed a precise insertion robot which operated reliably based on active accommodation by a flexible force sensing mechanism [8]. A position controlled robot equipped with a compliance mechanism (RCC) at the wrist which could perform precise insertion quickly without force feedback was developed at Draper Laboratory [9]. At SRI a system which could carry out assembly using a passive accommodation table had been developed [10]. In these practical system, global motion was controlled by simple positioning robots and fine accommodation was accomplished by tools specially designed for the tasks.

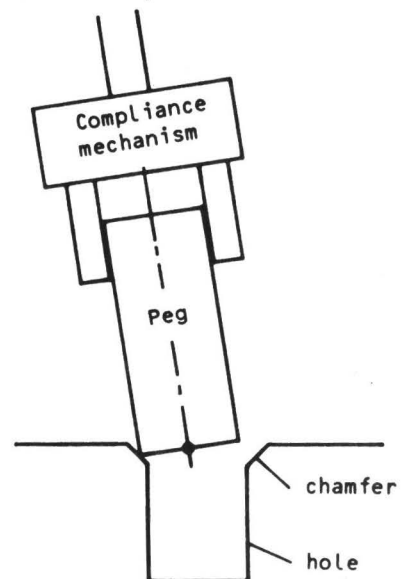


Fig. 1 Peg insertion by compliance mechanism

It would be instructive to compare the operation of a practical system with that of an intelligent robot. We will take peg insertion as an example. The robot attached with a RCC at the wrist will move a peg into a hole. As shown in Fig. 1, if there is a misalignment between the peg and the hole, the peg will be guided to the hole by the chamfer. The RCC is a compliance mechanism which provides the accommodation. If a lateral force is exerted at the tip of the peg, it will translate without rotation, and if a torque is exerted at the tip, it will rotate without translation. With the help of the chamfer any translation error

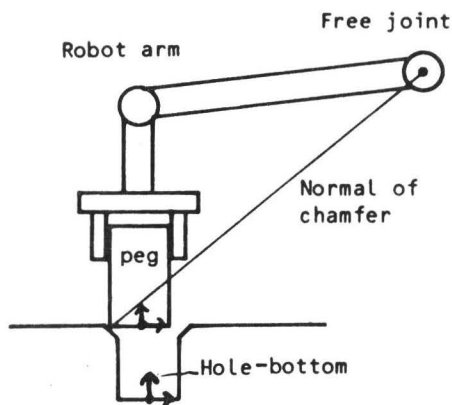


Fig. 2 Peg insertion by AL system

is corrected. During the insertion process into the hole any rotation error is corrected.

In the AL system the peg insertion could be described as:

MOVE peg TO hole-bottom
WITH FORCE = 0 ALONG X,Y OF hole-bottom

This command means "move coordinate frame peg to hole-bottom with two proper joints free so that the x and y components of force are zero." With this method the peg insertion sometimes becomes impossible. For example, if a free joint lies close by the normal of the chamfer as shown in Fig. 2, the angle between chamfer tangent and free surface becomes less than arctangent of friction coefficient. The peg will be locked. This is indeed a probable case. This disadvantage of the AL system shows not only that the system can not perform the peg insertion in this way, but also that the system lacks the systematic representation capability of the necessary skill. It is undesirable that the operation of a robot arm is dependent on its configuration. A user would like to write machine independent programs. In general a robot arm skill consists of patterns of action to the task environment, interpretation of results, and the decision of further action. We have to represent the knowledge about an action and its result, which is task particular, so that it is robot arm independent.

4. BASIC CONTROL SCHEME FOR SKILLFUL ROBOT ARM

4.1 Control in Cartesian Coordinate Systems

If the form of assembly is defined, it is most efficient to use the special devices or tools. Not only in assembly, but in general, it is

most efficient to design mechanisms suited for the task, and to control the operation of these mechanism using sensory feedback. Hitachi, Draper Laboratory, and SRI had taken this approach in developing assembly systems. Therefore it is senseless to simply compare the capabilities of special-purpose systems and intelligent robots in which generalities are most important. The skill representation capability of the intelligent robot should also be general so as to provide the skill of special-purpose tools. If we could build up virtual mechanisms by software, we would be able to utilize all task particular knowledge which is obtained from the study of special-purpose tools. And we will be able to exchange tools and skills by software.

What kind of robot arm control would be required in order to build up virtual mechanisms by software? It is motion and force control in cartesian coordinate systems. A few methods have been proposed to realize such a type of control [11,12]. We will describe here the direct servoing method in cartesian coordinates that seems to have the highest generality. As shown in Fig. 3 this system provides for the real time transformation between joint coordinates and cartesian coordinates, the transformation of acceleration and force in cartesian coordinates into joint torques. In addition, coriolis force, centrifugal force and gravity loading force are canceled in real-time.

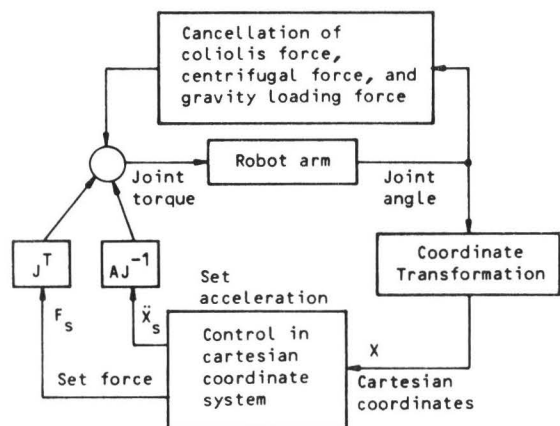


Fig. 3 Servo system for arm control in cartesian coordinate system

Each joint driving torque T is given by:

$$T = AJ^{-1}\ddot{X}_s + J^T F_s + T_n + T_g$$

where A : inertia matrix of the robot arm.

J : Jacobian matrix between joint and cartesian coordinate

$\ddot{X}_s$: set acceleration in cartesian coordinates.

F_s : set force in cartesian coordinates.

T_n : coliolis force and centrifugal torque

T_g : gravity loading torque

If the acceleration and force are given, joint driving torque is automatically computed by the system. The equation of motion becomes:

$$AJ^{-1}\ddot{X} + J^T F = AJ^{-1}\ddot{X}_s + J^T F_s$$

where $\ddot{X}$: real acceleration in cartesian coordinates.

F : real exerting force in cartesian coordinates.

Assuming that the acceleration caused by the

set, force F_s is small, that $\ddot{X} = \ddot{X}_s$, and that $F = F_s$. A compliance mechanism can be build up by defining F_s as:

$$F_s = \frac{1}{C} X$$

where C : compliance matrix

X : displacement from the origin

A damping mechanism is formed by setting:

$$F_s = \mu \dot{X}$$

where μ is damping coefficient. Motions are then controlled by setting the acceleration as:

$$\ddot{X}_s = \ddot{X}_r + \mu(\dot{X}_r - \dot{X}) + k(X_r - X)$$

where $\ddot{X}_r$: reference input for motion

k : stiffness or gain.

4.2 Real-time Computation

As the dynamic model of a robot arm is very complicated it has been considered impossible to compute joint torques in real-time. However, recent improvement both in calculation algorithms and in arithmetic elements has now made it realistic. Dynamic model in matrix representation, while simple, was very time consuming to compute. Bejczy developed an approximation model which was very fast to compute. The author formulated the equations of motion in a vector representation. Walker devised a recursive algorithm for evaluating the model in vector representation without redundant repetitive calculation [13]. As shown in Table, we can now calculate joint torques with this algorithm about 200 times faster than using the matrix method.

Table: List of time to compute joint forces and moment on the PDP 11/45 (Reproduced from [13])

Model	Language	Computation time
Matrix Model	Fortran	7.9 sec
Bejczy	Fortran	0.0025 sec
Walker	Fortran	0.0335 sec
Walker	Floating Point Assembler	0.0033 sec

Powerful arithmetic elements such as $16^{bit} \times 16^{bit}$ LSI multiplier with 200 nano second execution time, have also become available. It is not difficult to develop special-purpose processors with these arithmetic elements for performing fast vector calculation such as dot product or cross product. It would also be possible to make the calculation speed shown in Table ten times as fast. Above considerations show that sophisticated calculation for cartesian coordinates control could easily be carried out in less than 10 milli seconds.

4.3 Torque Controlled Robot Arm

Reliable force sensing systems have not been obtained yet, although force sensing, especially wrist force sensors have been actively studied. At the present stage it would be impossible to incorporate the force sensors into the arm control system controlling motion and force in cartesian coordinates. The author and co-workers have developed a robot arm driven by magnetic powder clutches and have

demonstrated that the robot arm with open loop torque control ability could exert a force with an error of ± 100 grams [14]. The author believes that it is possible to design a compact arm using torque motors, with a load capacity of 5 Kg and with a force exertion error of ± 100 grams. Torque control ability of a robot arm is essential if a cartesian coordinates control system is to be successfully implemented. The use of force sensors will be necessary for the precise control of force.

4.4 Execution Monitoring

If we specify the control of the arm in task related cartesian coordinates, we can make the execution of the task independent of any specific robot arm and we can easily analyze the causality between action and its results. In usual execution monitoring, reacting forces are monitored during geometrically specified motion. The control in cartesian coordinates enables another form of execution monitoring, for example, force or compliance are specified and the resultant position or velocity are observed. In the AL system, reaction forces are tested in a threshold manner. The AL command for a peg insertion is:

MOVE peg TO hole-bottom

WITH FORCE = 0 ALONG X, Y OF hole-bottom

ON FORCE (Z WRT hole-bottom) > 18*OZ DO STOP

In this case, once the z component of the force exceeds 18 oz during motion, the robot arm will stop. However if the characteristics of friction between the peg and hole is as shown in Fig. 3, the peg insertion will stop half way when the velocity exceed V_c . A programmer has to have knowledge about the friction. A control scheme that specifies force and observes the resultant motion would be more suitable for this task. The peg insertion task would be more reliable if the insertion speed versus insertion depth pattern is observed in the performance of the task.

To erect a thin bar on a table is a skill which can not be realized with simple threshold tests. If the cartesian coordinates are chosen as shown in Fig. 4 and the bar is rotated about X axis, an angle versus torque graph will be obtained as shown in Fig. 5. Angle α_0 , where the bottom plane of the bar comes in full touch with table surface, could be identified as the angle where the value of torque changes discontinuously in the graph.

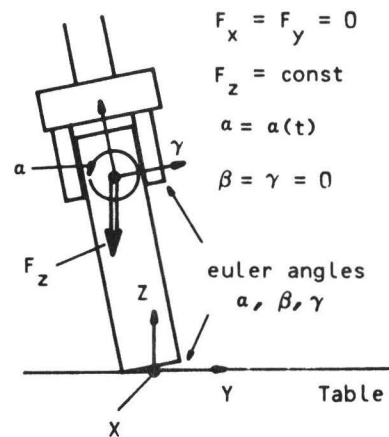


Fig. 4 Control mechanism for bar erection

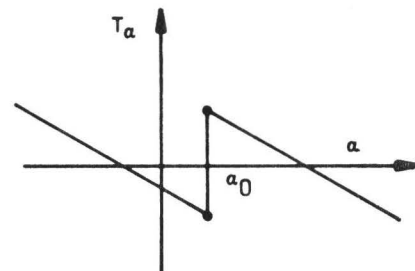


Fig. 5 Angle versus torque pattern in bar erection

CONCLUSION

The skill of a robot arm consists in the accumulation of task particular knowledge. This knowledge, that is a specification of the mechanism to perform a task, is embodied by the robot arm in the form of a virtual mechanisms in software. The realization of these virtual mechanisms is by motion and force control in cartesian coordinate systems and by monitoring the response patterns. The proposed method provides a systematic method by which to develop skills for an intelligent robot.

ACKNOWLEDGEMENTS

The author wishes to thank Prof. Richard Paul for helpful comments and for help in preparing the manuscript. He thanks Prof. King Sun Fu for encouragement and support at Purdue University. Document is prepared by Mellanie Boes and drawings by Marc Ream. He also thanks those people.

REFERENCES

- [1] Feldman, J., et al., "The of Vision and Manipulation to Solve the Puzzle," In Proc. IJCAI-71, London, September 1971.
- [2] Ejiri, M., et al., "A Prototype Intelligent Robot That Assembles Objects from Plane Drawings," IEEE Trans. Comp., Vol. C-21, pp. 161-170, 1972.
- [3] Shirai, Y., et al., "Visual Feedback of a Robot in Assembly," ETL A.I.R. Group Memo #1, 1972.
- [4] Finkel, R., et al., "AL, A Programming System for Automation," Computer Science Dept. Report CS-456, Stanford University, November 1974.
- [5] Pieper, D. L., "The Kinematics of Manipulators under Computer Control," Stanford Artificial Intelligence Project Memo No. 72, October 1968.
- [6] Paul, R., "Trajectory Control of a Computer Arm," In Proc. IJCAI-71, London, September, 1971.
- [7] Bolles, R., et al., "The Use of Sensory Feedback in a Programmable Assembly System," Stanford Artificial Intelligence Project Memo No. 220, Stanford University, Stanford, CA, October 1973.
- [8] Goto, T., et al., "Precise Insert Operation by Tactile Controlled Robot," In Proc. 4th ISIR, Tokyo, 1974.
- [9] Nevins, J. L., et al., "Computer Controlled Assembly," Scientific American, Vol. 238, No. 2, pp. 62-74, February 1978.
- [10] Rosen, C., et al., "Machine Intelligence Research Applied to Industrial Automation," Eighth Report, SRI International, Menlo Park, CA, August 1978.
- [11] Paul, R., et al., "Compliance and Control," In Proc. JACC-76, 1976.
- [12] Takase, K., "Task-Oriented Variable Control of Manipulator and its Software Servoing System," In Proc. of IFAC Symposium on Information-Control Problems in Manufacturing Technology, Tokyo, 1977.
- [13] Paul, R., et al., "Advanced Industrial Robot Control Systems," First Report, Purdue University, West Lafayette, IN, May 1978.
- [14] Takase, K., et al., "The Design of an Articulated Manipulator with Torque Control Ability," In Proc. 4th ISIR, Tokyo, 1974.