HISTORICAL PERSPECTIVE AND STATE OF THE ART IN ROBOT FORCE CONTROL

Daniel E. Whitney
The Charles Stark Draper Laboratory, Inc.
Cambridge, MA 02139

ABSTRACT

This presentation combines historical lineage, assessment of the state of the art, and representative videotapes on robot force control. The difference between continuous and logic branching strategies is described. The development of various impedance strategies and hybrid methods is traced and compared. The problem of stability is discussed and remedies are related to higher strategy issues.

Introduction

Robot force control involves integration of task goals, trajectory generation, force and position feedback, and modification of the trajectories. It requires understanding contact tasks so that effective strategies and trajectories can be planned and feedback data can be understood. It also requires control so that the robot's responses will be stable. Finally, it requires filtering and estimation to remove unwanted signals like noise and robot motion errors so that useable feedback information can be obtained. These issues--task analysis, strategy generation, control stabilization, and filtering--must be dealt with together if effective force control systems are to be created.

This paper briefly sketches the development of the main lines of robot force control research and describes the principal problems remaining to be solved. Robot force control actually begins with remote manipulator and artificial arm control in the 1950s and 1960s. The stability issues emerged immediately, but the strategy and task understanding issues remained submerged because people controlled these systems in "natural" ways. In the late 1960s and 1970s, the first computer controls of force feedback were attempted. Various approaches to creation of strategies emerged. All have depended on people to formulate the details and those which were tested experimentally encountered the stability problem. A way out of this problem has been found in flexible sensors, but as yet no automatic generation of strategies has been achieved.

Manipulators and Artificial Arms

In the 1940s, Goertz[1] invented mechanical master-slave manipulators for radioactive hot lab work. In the 1950s he implemented these as electric-servo manipulators with force reflection: the operator guided the master with his hand and felt the contact forces experienced by the slave which were reflected through the joints of both devices.

In the 1960s, Mann[2] led the development of a force feedback powered artificial elbow. The joint motor was driven by signals from muscle electrodes and a strain gauge in the joint. Thus the amputee could exert muscle effort to counter the force just as he did with his natural arm.

Time delay of as little as 250 ms in these systems caused them to go unstable.[3] While proper design can eliminate the problem in artificial arms, it caused the demise of a proposed use for force-feedback manipulators: their utilization in space from ground-based control stations. Radio wave propagation delay is not the reason, but rather the practical delays caused by coding, decoding, error checking, channel sharing, and so on. Similar delays exist today in robot control systems, caused by dynamic bandwidth limitations of robot arms and by computation delays.

Computer Controlled Arms

In the late 1960s and early 1970s attention turned to replacing the human operator with a computer. A person on the ground could give goal commands for execution by a computer/arm in space, thus eliminating the delay.[4] Two important problems emerged: (1) how to structure multi-axis arm systems that sensed forces and used them to modify position commands, and (2) how to relate the requirements of a task to the motions required and forces anticipated so that force-motion strategies could be formulated.

Two tracks emerged in pursuit of these issues, which I call Logic Branching and Continuous, respectively.

Logic branching begins with the work of Ernst[5] and extends through Barber,[6] Hill,[7] and Paul.[8] It is primarily scalar in nature and consists of strings of statements in a computer control language:

```
Move in X until force exceeds F
Twist about Z until torque exceeds T
•
•
•
```

The essence of this is a set of discrete moves terminated by contact. One of its descendants today is the attempt to create rule-based force control systems.

Continuous force control begins with work by Nevins,[9] Whitney,[10] and Groome,[11] and is based on multi-axis force-torque information obtained from 6 axis sensors[12] combined with multi-axis response motions based on coordinated continuous axis motions.[13] Continuous force control was applied to assembly and edge-following tasks and was based on a feedback matrix that converted a sensed force vector into a response velocity vector. This response velocity was combined with the original velocity command to create a new command. Since continuous contact was maintained between the arm and the environment, the timing and stability of controlled motions was important. Table 1 summarizes the similarities and differences between Logic Branching and Continuous Force Feedback.

Table 1. Comparison of logic branching and continuous force feedback.

	"Style"	Scalar/ Vector	Direction/ Magnitude	Static/ Dynamic	Relation to Task Physics
Logic Branching	IF-THEN similar to computer programming	Scalar or serial among the components	Primarily direction-oriented. Little use of magnitude except in binary sense	Primarily static. Dynamics, gains, inertias do not affect strategy	Series of moves & tests based on intermittent contacts, dead reckoning, search, and deliberate forcing of a sequence of events related to task geometry
Continuous force feedback	Similar to servo or control theory	Vector or simultaneous among the components	Primarily magnitude-oriented. Actual vector magnitude & direction used	Primarily dynamic	Continuous sequence of vector forces & motions based on continuous contact, forces raised by task geometry, friction, and compliance, and sequence of desired relations between force and motion

Since these early efforts, several other methods have developed. A categorized sampling of these methods includes:

- Damping methods
  - Accommodation[10]
  - Joint Compliance in a Cartesian Computer Language[8]
- Position Methods
  - Active Compliance[14,15]
  - Passive Compliance[16,17]
- Impedance or Energy Methods[18,19]
- Explicit Force Control[20]
- Implicit Force Control[21]
- Hybrid Force-Position Control[22,23]

The next section contains brief descriptions of each of these.

Force Feedback Architectures

Most of the force feedback systems developed to date can be fitted into the overall architecture shown in Figure 1. The robot is commanded along some nominal path or velocity which is modified by motion updates created by the strategy. At some point contact occurs between the robot and its environment. Their collective deformation and stiffness give rise to forces which react directly on the robot's joints. The contact forces are also sensed and fed to the strategy. Some architectures deal with commanded forces as well. Programming languages and control algorithms can both be represented this way.

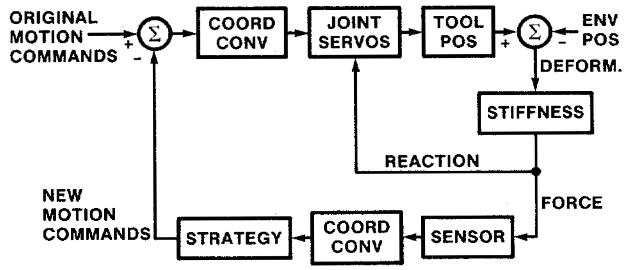


Figure 1. Architecture of robot force feedback.

Several architectures will be described briefly. Figure 2 shows stiffness control. In this and other diagrams, J is the arm's Jacobian matrix, $\underline{X}$ and $\dot{\underline{X}}$ are vector positions and velocities in hand or world coordinates, $\underline{\tau}$ is a vector of arm joint torques, $\underline{X}_E$ is the position of the contacted environment, K_E is the net stiffness of all contacting items (arm, sensor, environment, etc.), and $\underline{F}$ is the resulting contact force. Vectors are 6×1 and matrices are 6×6. To implement stiffness control we create matrix K_{F1} with dimensions of compliance (position/force).

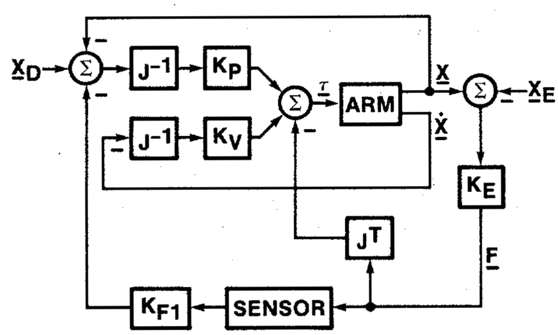
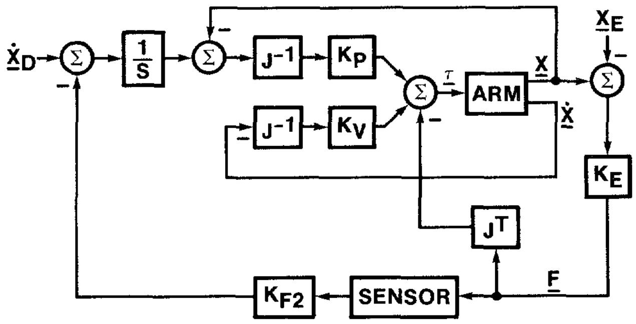


Figure 2. Stiffness control.

Figure 3 shows damping control. Compared to stiffness control, this is an integrating controller in which sensed forces give rise to velocity modifications. These velocities result from multiplying sensed forces by matrix K_{F2} which has dimensions of admittance (velocity/force).



Equilibrium contact force = $[K_{F2}^{-1}]\,\dot{X}_D$

Figure 3. Damping or accommodation control.

Figure 4 shows one form of impedance control. It is a generalization of the previous two. Alternatively, impedance control comprises a PID force feedback loop.

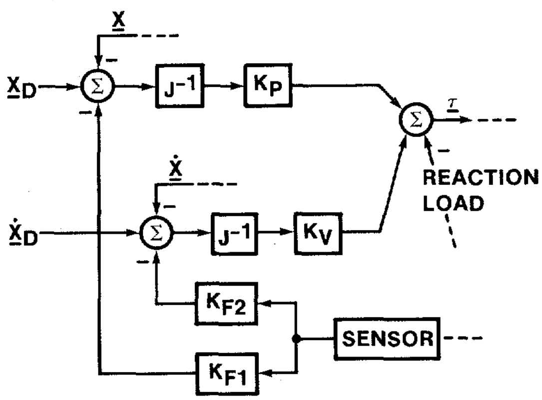


Figure 4. Impedance control.

Figure 5 is explicit force control. Unlike the previous techniques, this one has a desired force input rather than position or velocity input.

FORCE CONTROL

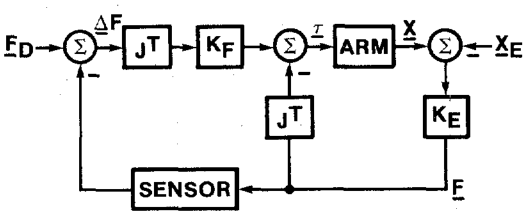
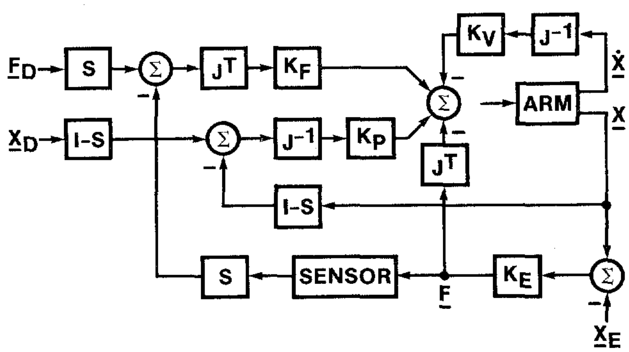


Figure 5. Explicit force control.

Figure 6 is a hybrid of position and force control. Matrix S selects which world or hand axes are to be force controlled and which are to be position controlled. Naturally, damping control could be used instead of position control.

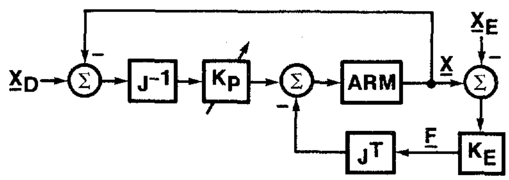


S = Diagonal matrix of ones and zeroes

I = Identity matrix

Figure 6. Hybrid control.

Figure 7 shows implicit force control. Here no sensor is involved. Instead the joint servo gains are adjusted to give the hand a particular stiffness matrix. One could apply this logic to the velocity feedback gain K_V instead, producing a desired admittance matrix at the hand.



For desired hand stiffness K_H:

$$K_P = J^T K_H J$$

$$K_H = J^{-T} K_P J^{-1}$$

Figure 7. Implicit force control.

Some Comparisons

While these methods look different, they can often be configured to achieve the same results. A few examples are given for comparison.

a) Peg-in-hole: see Figure 8. We want the peg to correct lateral position errors by responding to the chamfer contact force by sliding laterally without turning. We also want angular errors to be corrected by the peg turning in response to torques exerted by contact forces in the hole. Three implementations are compared.

Stiffness control: command motion along the peg's axis in hand Z coordinates. Make K_{F1} diagonal in coordinates centered at the peg's tip. When the peg reaches the bottom of the hole, some net contact forces will remain, proportional to the initial errors and $K_{F1}K_E$.

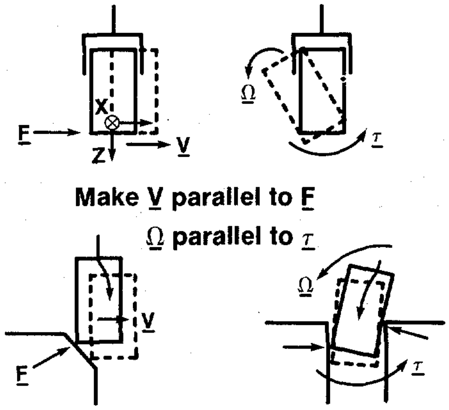


Figure 8. Force-motion response for peg-hole assembly.

Damping control: command motions as in stiffness control and construct K_{F2} like K_{F1} except that $K_{F2}(3,3)$ should be small. Theoretically, no contact forces will remain when assembly is done except in the insertion direction, where the force's magnitude will be $\dot{Z}_D/K_{F2}(3,3)$. In other words, the feedback gain should be large in the direction (hand Z) where the task is expected to yield, and low (hand X, Y, θ_X, θ_Y) where the task is expected to resist deflection.

Programming Language:

```
COMPLY FORCE X 0  } i.e., move until
COMPLY FORCE Y 0  } no force is felt
COMPLY TORQUE X 0 } i.e., rotate
COMPLY TORQUE Y 0 } until no torque
                    is felt
STOP FORCE Z - 1
MOVE Z
```

Implementation of such commands will require some logic similar to that shown in Figure 3 or Figure 6.

b) Pack blocks in a corner of a box: see Figure 9. We will lower each block in the Z direction until it contacts the bottom of the box. We then want the hand to steer automatically toward the corner with the black dot regardless of small initial position or orientation errors.

Damping control: command motion along hand Z. Construct matrix K_{F2} with nonzero entries as follows:

(1,2) = a
(2,1) = a
(2,3) = -a
(4,4), (5,5), (6,6) = b

The first two entries will guide the hand along whichever side adjacent to the dotted corner it hits, steering it toward the dot. The third entry will steer the hand toward a side after the bottom is contacted. The last three will align the sides of the block to the sides and bottom of the box.

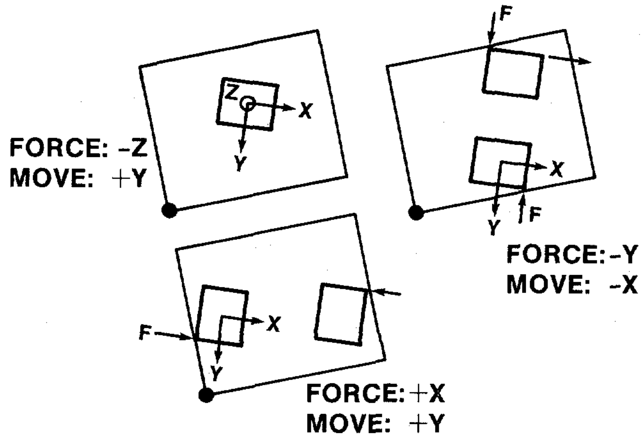


Figure 9. Force-motion response for box packing.

Programming language:

```
STOP FORCE Z -1
MOVE Z
COMPLY X 0
STOP FORCE Y -1
MOVE Y
STOP FORCE X 1
MOVE -X
```

Both techniques theoretically will work as long as friction is small and initial θ_Z error is less than 45° minus the friction angle.

c) Follow an edge while maintaining a preset contact force F_{XD}: see Figure 10.

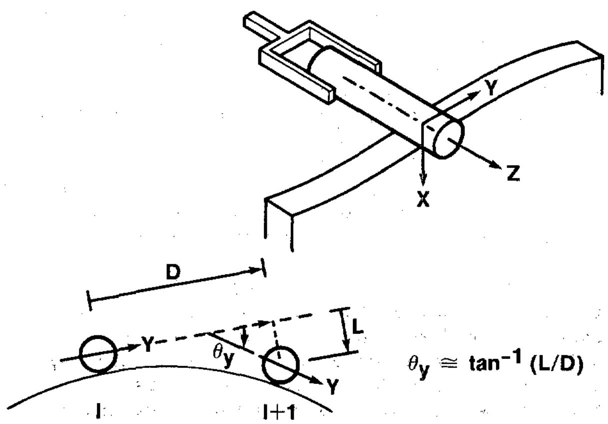


Figure 10. Edge following.

Stiffness control: move along X until the surface is contacted, then command additional ΔX motion until

$$F_{XD} = \{K_E[I - (I + K_{F1}K_E)^{-1} \cdot K_{F1}K_E]\}_{(1,1)}\Delta X$$

Move along Y a distance D. Establish a new tangent direction by moving along X as before and noting distance L to contact. Rotate about Y according to

$$\theta_Y = \tan^{-1}(L/D)$$

Damping control: move with speed $\dot{X}_D$ until surface is contacted. Maintain $\dot{X}_D$ as

$$\dot{X}_D = F_{XD}K_{F2}(1,1)$$

Make $K_{F2}(2,1) = a$. Then $\dot{Y} = a\dot{X}_D$ will be commanded automatically as soon as contact occurs. If contact is lost, $\dot{Y}$ will fall to zero and the arm will drive toward the surface automatically until contact is re-established. Maintain Y tangent to the surface the same way as in stiffness control.

Hybrid control:

Set $F_X = F_{XD}$

$\dot{Y} = \dot{Y}_D$

$\dot{Z} = 0$

$\dot{\theta}_X = \dot{\theta}_Z = 0$

$\dot{\theta}_Y$ = (reorientation strategy as above)

Stability Issues

In the above examples, we noted which entries in the feedback matrices should be large, small, positive, negative, or zero, but we did not specify any magnitudes. The latter must be chosen carefully with stability in mind. The same considerations apply in programming language implementations, since in both cases one must specify a change in position, velocity, or torque in response to sensed forces. In both cases this will be done in a sampled data way. Here we will analyze simple stiffness and damping force feedback situations.

a) Stiffness control: see Figure 11. In this single axis implementation, the controller seeks to advance or retract θ_k until the deflection $\theta_r - \theta_k$ is appropriate for a spring of stiffness K_F acted on by force F_S. (Note that K_F is a stiffness while K_{F1} is a compliance, or inverse stiffness.) Equilibrium exists when $F_k = F_s$. The response to a step input of θ_e is

$$\theta_k = \frac{a}{1+a}\left(1 - \gamma^k\right)\theta_e$$

where

$a = K_E/K_F$
$\gamma = 1 - b - ab$
b = gain
k = time index

In Figure 12 the stability condition $|\gamma| < 1$ is plotted. From this we can conclude that large a implies small b and vice versa. Note that large a would be desirable. That is, a good stiffness controlled system should be compliant (small K_F) and be able to deal with stiff environments (large K_E). Figure 12 shows that this will be obtained stably only with low gain and possibly sluggish behavior.

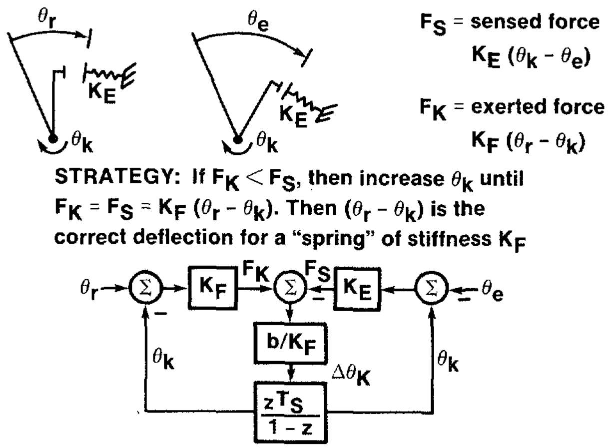
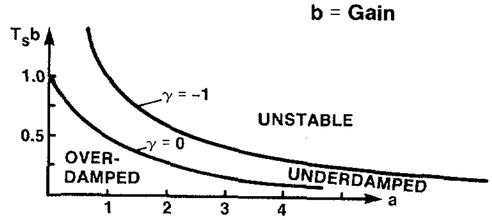


Figure 11. Stiffness control.

$$\theta_k = \frac{a\theta_e}{1+a}[1 - \gamma^k]$$

STABLE IF $|\gamma| < 1$

$\gamma = 1 - T_s b - aT_s b$
$a = K_E/K_F$
b = Gain



Large K_E/K_F is desirable

To work with large K_E/K_F, we must make gain low

To have high gain, we need low K_E/K_F

Figure 12. Stability analysis of stiffness control.

b) Damping control: see Figure 13. In this single axis implementation, a velocity modification $\dot{\theta}_m$ is commanded proportional to the sensed force.[10] The response to a collision at $\theta_e = 0$ with $\dot{\theta}_D = 1$ is

$$\theta_c(k+1) = \theta_c(k) - T_s\beta\theta_a(k)$$

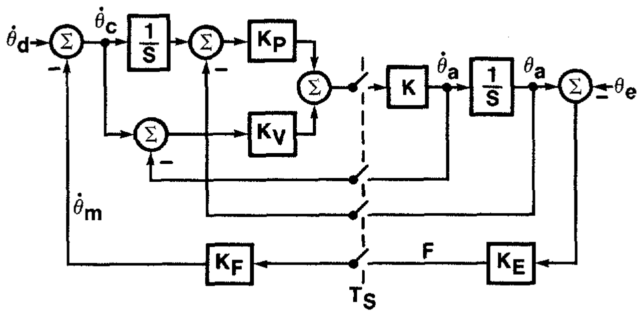
$$\theta_a(k+1) = \alpha\theta_c(k) + (1-\alpha)\theta_a(k)$$

where

$$\alpha = T_s K_p K$$

$$\beta = K_F K_E$$

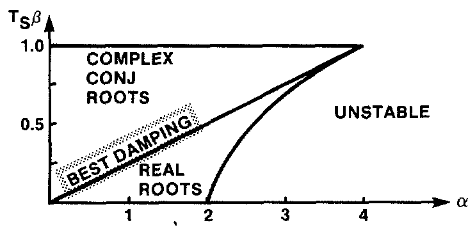
Figure 14 shows the stability conditions in terms of α and $T_s\beta$. Damped behavior occurs for small α and $T_s\beta$, implying low gain and small $K_F K_E$.



STRATEGY: Command a velocity modification $\dot\theta_m$ proportional to sensed force

Figure 13. Damping control.

$$\theta_c(k+1) = \theta_c(k) - T_S\beta\,\theta_a(k)$$
$$\theta_a(k+1) = \alpha\,\theta_c(k) + (1-\alpha)\,\theta_a(k)$$
$$\alpha = T_S K_P K,\ \beta = K_F K_E$$



Large K_F is desirable but causes instability unless K_E is small

Figure 14. Stability analysis of damping control.

If we want to deal with stiff environments, we must make K_F small, implying large contact forces or low commanded velocities.

Thus both of these implementations suffer sluggish behavior in the face of stiff environments. Note that this problem cannot be remedied by using smaller T_S. The above equations imply that whatever command is given to the arm at time k, the arm will respond completely before time k + 1. Thus the smallest T_S that can be considered corresponds to the bandwidth of the arm and its controller, which is in turn limited by such dynamic properties as arm inertia and actuator torque limits. Since every arm has a maximum bandwidth, the above equations, while simplified, still represent the physics adequately.

The equations suggest two remedies: small arms or hands with higher bandwidth, or deliberate passive compliance installed in the arm's wrist, for example, to make the effective K_E small. An effective implementation of this is the Remote Center Compliance.[16,17] It embodies matrix K_{F1} as configured for the peg-in-hole strategy. Thus the strategy will be executed automatically, without any sensing, computation, or arm command modifications. For this reason, this strategy is called passive compliance.

To implement the other strategies discussed above and still reduce K_E, we need a "soft sensor." This may be obtained by using a standard rigid force torque sensor mounted to a Remote Center Compliance, or by using an Instrumented Remote Center Compliance.[24]

The conclusion to be drawn from this discussion is that, regardless of the system architecture, force feedback ultimately is a servo, whether it is implemented as a computer algorithm or analog control loop. Stability issues will arise in any case, and deliberate compliance is so far the only solution consistent with fast response and low contact forces.

Unsolved Problems

Force control needs more work in these areas: filtering and estimating that allow more sophisticated algorithms, more complex controllers and better stabilization, and better theory for deciding what the feedback strategy should be for each task. Progress in each of these areas is represented by work in references 23, 25, 26, 27, and 28.

Mason's work[23] extends work of Raibert, Craig, and Whitney into a formal theory of force and motion constraints. A major goal is to be able to generate control strategies automatically. Whitney and Edsall[25] are applying signal processing and estimation theory to time series of force-torque data. The goal is to exploit the correlation in the time series for analysis, filtering, or prediction of robot behavior and task progress. Stepien et al[26] and Roberts et al[27] have revived the stability problem. The former analyze an entire robot and contact task and design a controller to stabilize it. The latter study the interplay of various compliances in the sensor and robot to determine the effect on performance of force control. Cannon[28] is modeling flexible arms in order to improve control of both non-contact and contact tasks. Both studies involve modeling wave-like behavior in the arm plus studying different types of sensors: at joints, along the arm, and at the hand. Position, force, and proximity sensors have been utilized. Attention is also being given to the long-neglected problem of collisions.

Today, force control is well behind vision in both sophistication of theory and level of application in industry. Sensors and computational capacity are not limiting progress. More effort is needed to identify and solve basic theoretical problems.

References

[1] Goertz, Ray C., "Fundamentals of General Purpose Remote Manipulators," Nucleonics, Vol. 10, Nov. 1952, pp. 36-42.

[2] Rothchild, R. A. and R. W. Mann, "An EMG-Controlled Force Sensing Proportional Rate Elbow Prosthesis," Proc. 1966 Symp. on Biomedical Eng., Milwaukee, 1966.

[3] Ferrell, W. R., "Remote Manipulation with Transmission Delay," NASA TN D-2665, 1965. See also "Delayed Force Feedback," Human Factors, Oct. 1966, pp. 449-455.

[4] Ferrell, W. R. and T. B. Sheridan, "Supervisory Control of Remote Manipulators," IEEE Spectrum, Vol. 4, pp. 81-88, Oct. 1967.

[5] Ernst, H. A., "MH1--A Computer Operated Mechanical Hand," MIT Sc.D. Thesis, 1961.

[6] Barber, J. D., "MANTRAN: A Symbolic Language for Supervisory Control of an Intelligent Remote Manipulator," MIT S.M. Thesis, 1967.

[7] Hill, J. W. and A. J. Sword, "Manipulation Based on Sensor-Directed Control," 17th Annual Human Factors Convention, Washington, 1973.

[8] Paul, R. P. and B. Shimano, "Compliance and Control," Proc. JACC 1976, pp. 694-699.

[9] Nevins, J. L. and D. E. Whitney, "The Force Vector Assembler Concept," Proceedings, First CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, Sept. 1973.

[10] Whitney, D. E., "Force Feedback Control of Manipulator Fine Motions," ASME J. Dyn. Sys. Meas. and Control, June 1977, pp. 91-97.

[11] Groome, R. C., "Force Feedback Steering of a Teleoperator System," S.M. Thesis, MIT, 1972.

[12] Drake, S. H. and P. C. Watson, "Pedestal and Wrist Force Sensors for Industrial Assembly," Proceedings, 5th Int'l. Symp. Industrial Robots, Chicago, 1975.

[13] Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," Trans. ASME J. Dyn. Sys. Meas. and Control, Dec. 1972, pp. 303-309.

[14] Salisbury, J. K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates," Proceedings, 19th IEEE Conf. on Decision and Control, Dec. 1980.

[15] Salisbury, J. K. and J. J. Craig, "Articulated Hands: Force Control and Kinematic Issues," Int. J. Robotics Research, Vol. 1, No. 1, 1982, pp. 4-17.

[16] Watson, P. C., "A Multidimensional System Analysis of the Assembly Process as Performed by a Manipulator," 1st N. Am. Robot Conf., Chicago, 1976.

[17] Drake, S. K. and S. N. Simunovic, "The Use of Compliance in a Robot Assembly System," Preprints, IFAC Symp. on Info. and Control Problems in Manuf. Tech., Tokyo, 1977.

[18] Hogan, N., "Control of Mechanical Impedance of Prosthetic Arms," Proc. 1980 JACC, San Francisco.

[19] Hanafusa, H. and H. Asada, "A Robot Hand with Elastic Fingers and Its Application to Assembly Process," Preprints, IFAC Symp. on Info. and Control Problems in Manuf. Tech., Tokyo, 1977.

[20] cf. Ref. 9.

[21] Borrel, P., "Modele de Comportement des Manipulateurs, Applications a l'Analyse de leurs Performances et a leur Commande Automatique," Thesis, Universite des Sciences et Techniques du Languedoc, Montpellier, France, 1979.

[22] Raibert, M. H. and J. J. Craig, "Hybrid Position/Force Control of Manipulators," Trans. ASME J. Dyn. Sys. Meas. and Control, Vol. 102, June 1981, pp. 126-133.

[23] Mason, M. T., "Compliance and Force Control for Computer Controlled Manipulators," Trans. IEEE Systems, Man and Cybernetics, Vol. SMC-11, No. 6, June 1981, pp. 418-432.

[24] DeFazio, T. L., D. S. Seltzer, and D. E. Whitney, "The Instrumented Remote Center Compliance," Industrial Robot, Vol. 11, No. 4, Dec. 1984, pp. 238-242.

[25] Whitney, D. E. and A. C. Edsall, "Modeling Robot Contour Processes," Proceedings 1984 ACC, San Diego.

[26] Stepien, T. M., L. M. Sweet, M. C. Good, and M. Tomizuka, "Control of Tool/Workpiece Contact Force with Application to Robotic Deburring," Proceedings IEEE Conf. on Robotics and Automation, St. Louis, 1985.

[27] Roberts, R. K., R. P. Paul, and B. M. Hillberry, "The Effect of Wrist Force Sensor Stiffness on the Control of Robot Manipulators," Proc. IEEE Conf. on Robotics and Automation, St. Louis, 1985.

[28] Cannon, R. H., "Rapid Precise Control or Nonrigid Manipulators," presented at Intelligent Controls: Achievements and Issues, SRI Int'l., Palo Alto, Nov. 1984.