MODELING, IDENTIFICATION AND CONTROL, 1988, VOL. 9, NO. 2, 99–108
doi:10.4173/mic.1988.2.4

A positional deviation sensor for training robots

FREDRIK DESSEN and JENS G. BALCHEN†

Keywords: *Robot manipulators, direct training, positional sensing.*

A device for physically guiding a robot manipulator through its task is described. It consists of inductive, contact-free positional deviation sensors. The sensor will be used in high performance sensory control systems. The paper describes problems concerning multi-dimensional, non-linear measurement functions and the design of the servo control system.

1. Introduction

Over the years, several principles have been introduced for programming industrial robots. The methods most commonly used, involve showing the robot a sequence of positions by physically leading the robot through its task. This has been done either by remote control through a teach pendant or by moving the arm manually. The advantage of these methods compared to off-line programming is the continuous feedback given to the programmer when watching the manipulator moving through space. This is most clear in the case of manual programming where the operator not only receives visual feedback close up, but also has direct control of the manipulator through his hands. The method is very popular in applications such as paint spraying, where it allows trajectory recording in real time. However, the success of this approach relies on careful design of the manipulator. The arm should be lightweight, preferably gravitational effects should be compensated, and joint friction during the teach process must be low. These considerations may increase the price of the robot in question.

Lately, several approaches have been made to include the possibility of manual control in arbitrary industrial robot systems. This is usually done by mounting a force sensing handle on the manipulator end effector. The force applied is then used to indicate the desired motion of the manipulator. Unfortunately, this approach leads to a stability problem due to interaction between the operator and the manipulator (Hirzinger, 1982). The bandwidth of such a servo assisted manual control system may often be low, and it is still necessary to apply considerable force in order to make the manipulator move.

The ideal system for manual lead-through programming would be a system where the operator can hold the tool freely in his own hands and complete the task by himself. By this method, no special care needs to be taken when designing the manipulator. The programmer will not be hampered at all by the manipulator, only by the weight and mass of the tool. This approach relies on the existence of a special positional measurement system. One possibility is using a lightweight, low friction dummy arm with servo displacement sensors. A second possibility is using remote sensing devices based on optical, magnetic, sound- or RF-wave measurements.

Received 21 October 1987.

† Norwegian Institute of Technology, Division of Engineering Cybernetics, 7034 Trondheim, Norway.

This paper presents a new method for contact-free positional sensing of a tool or teach-handle moved freely by the operator (Balchen 1986). The system consists of a small-range positional sensor mounted at the tip of the manipulator. The approach is outlined in the next section of this paper. Later sections describe the positional sensor and the tracking system in more detail. These sections also describe some of the problems that were met during the design of the system. A previous version of this paper was presented at the NATO ARW 'Sensor Devices and Systems for Robotics', October 1987, and a detailed investigation of the sensor is made in (Dessen 1988).

2. Outline of the system

Consider a robot manipulator with a sensor mounted at its tip. Let the sensor be able to measure the position of a tool, which is held by an operator. The situation is illustrated in Fig. 2.1. Here the position of the manipulator wrist relative to its base is represented by the homogeneous transformation matrix $\boldsymbol{T}$. In the same way the tool position relative to the wrist and the base are denoted by ${}^{T}\boldsymbol{C}$ and $\boldsymbol{C}$ respectively. The situation is described by the transformation equation

$$\boldsymbol{C} = \boldsymbol{T}\,{}^{T}\boldsymbol{C} \tag{2.1}$$

In Fig. 2.2, matrix ${}^{T}\boldsymbol{C}$ is split into two fixed transformations, $\boldsymbol{M}$ and $\boldsymbol{F}$, and one of three alternative displacement matrices, ${}^{T}\boldsymbol{D}$, ${}^{M}\boldsymbol{D}$ or ${}^{C}\boldsymbol{D}$. Here, $\boldsymbol{M}$ defines the sensor frame, $\boldsymbol{F}$ is a tool description matrix, and ${}^{M}\boldsymbol{D}$ expresses tool displacement as seen from the sensor frame. ${}^{T}\boldsymbol{D}$ and ${}^{C}\boldsymbol{D}$ are related to ${}^{M}\boldsymbol{D}$ by

$${}^{T}\boldsymbol{D} = \boldsymbol{M}\,{}^{M}\boldsymbol{D}\,\boldsymbol{M}^{-1} \tag{2.2}$$

$${}^{C}\boldsymbol{D} = \boldsymbol{F}^{-1}\,{}^{M}\boldsymbol{D}\,\boldsymbol{F} \tag{2.3}$$

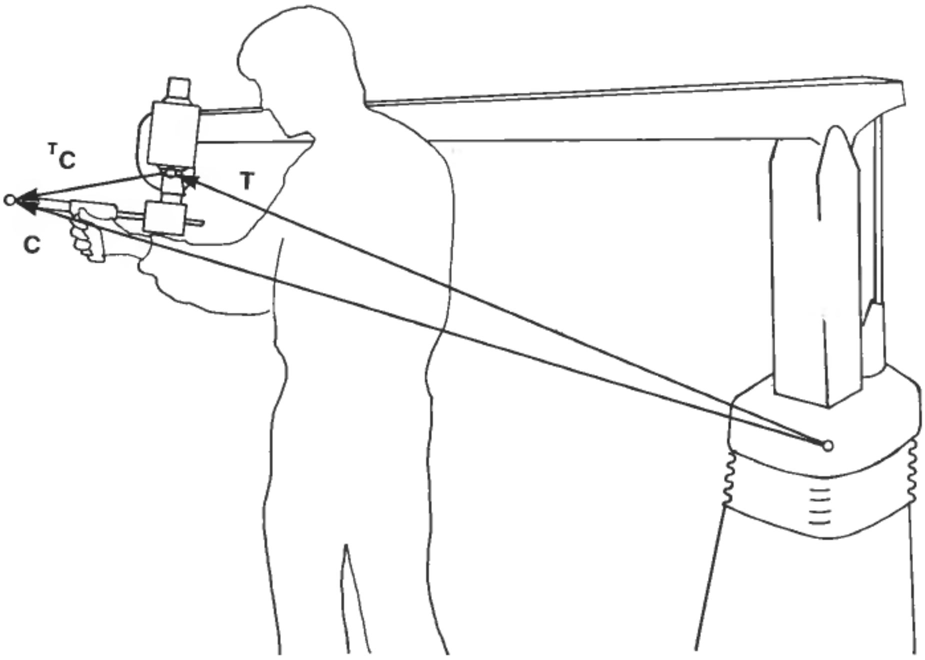

Figure 2.1. Training by manual tool motion.

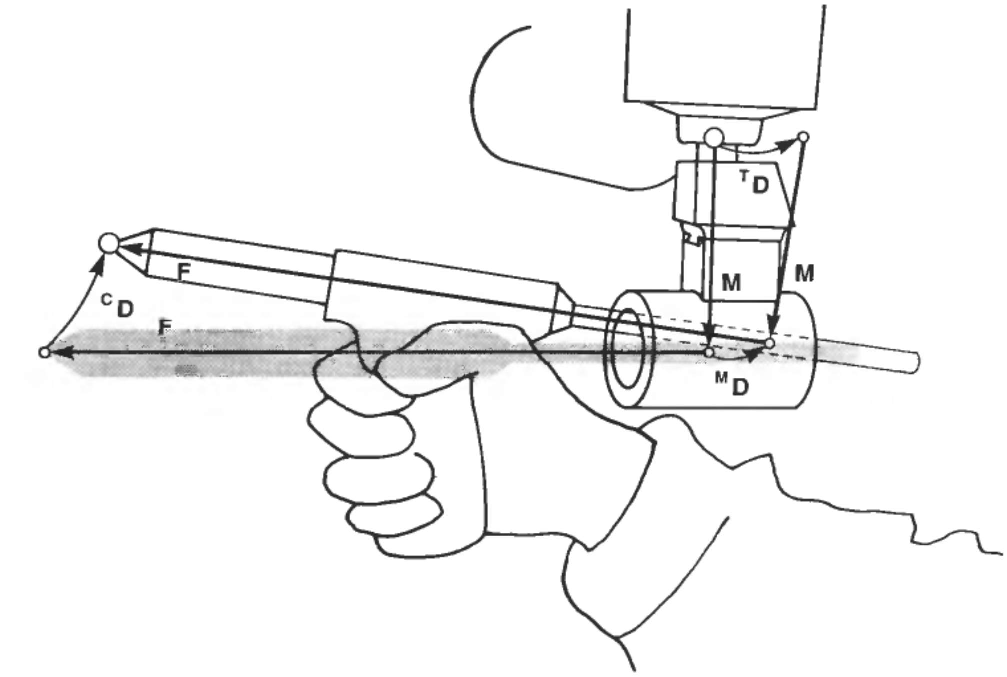

Figure 2.2. Position of the tool relative to the wrist, i.e. $^T\boldsymbol{C}$, may be obtained by following either of the paths $\boldsymbol{MF}^C\boldsymbol{D}$, $\boldsymbol{M}^M\boldsymbol{DF}$ or $^T\boldsymbol{DMF}$.

Using the displacement matrix $^M\boldsymbol{D}$, (2.1) can be written

$$\boldsymbol{C} = \boldsymbol{TM}^M\boldsymbol{DF} \tag{2.4}$$

Since $\boldsymbol{T}$ may be computed from the manipulator joint displacements, and $^M\boldsymbol{D}$ is given by the sensor output, tool position relative to base may at any time be found using (2.4). The computed tool motion may be recorded in order to create a robot program.

Because of the limited range of the positional sensor, we want the manipulator to follow the tool as closely as possible. This tracking is done by a coordinating controller. A complete system for tool tracking and trajectory recording is shown in Fig. 2.3. The input to the positional controller may be represented by the vector

$$^M\boldsymbol{d} = [d_1, d_2, d_3, \delta_1, \delta_2, \delta_3]^T \tag{2.5}$$

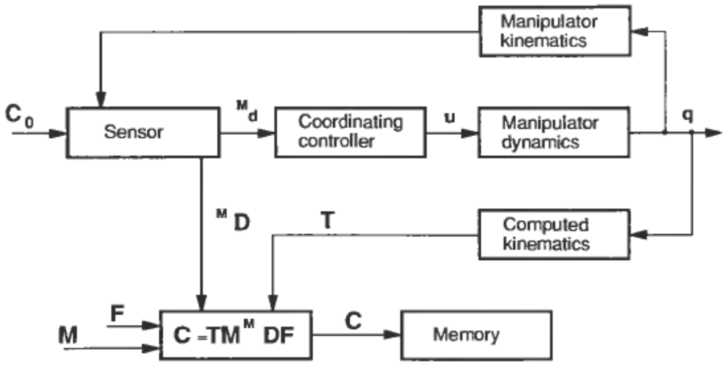

Figure 2.3. Tool tracking and motion recording. The tool motion, *C*, is reproduced and stored in memory.

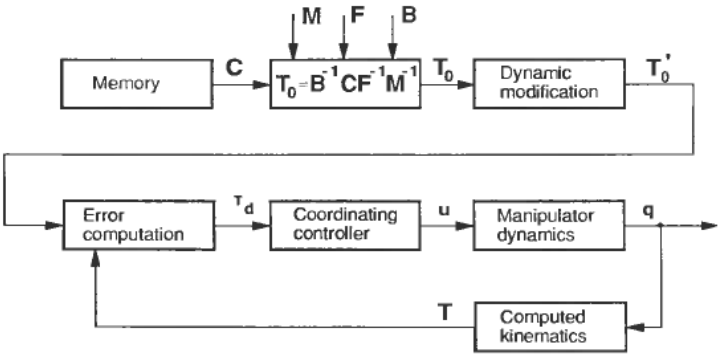

Figure 2.4 Motion playback.

which is extracted from the tool displacement matrix, ${}^{M}\boldsymbol{D}$ (Paul 1981). The generalized coordinates, $\boldsymbol{q}$, are used to compute the wrist position matrix, $\boldsymbol{T}$.

The motion playback and control system is designed to make a tool repeat the motion described by $\boldsymbol{C}$, which is stored in memory. However, the motion sequence may be modified in several ways before it is used as a reference. In general, $\boldsymbol{M}$ and $\boldsymbol{F}$ may have been altered since the time of recording, and it may be desirable to refer motion to a new coordinate system, $\boldsymbol{B}$. These modifications enter as shown in Fig. 2.4.

3. Description of the sensor

So far, very little has been said about the displacement sensor itself; only that it is mounted at the tip of the manipulator, and that it possibly has a limited range.

The sensor consists of two cylindrical parts, made of ferromagnetic material. Fig. 3.1. By means of a solenoid mounted inside the larger, hollow part, the two cylinders are magnetized in opposite, axial, directions. This creates a radial magnetic field between the two parts, which varies according to their relative position. By measuring the magnetic field at selected points, relative motion is monitored.

A section of the sensor is shown in Fig. 3.2. Here, 4 pick-up solenoids (M2, M4, M6 and M8) are used to measure translation along the p_2 and p_4-axes (Fig. 3.3). From this, translation along the ${}^{M}y$-axis and rotation about the ${}^{M}x$-axis (Fig. 3.1) may be computed. Similarly, 4 solenoids are used to monitor p_1 and p_3-motion. Axial translation, p_5, is measured by M9 and M10. It is seen that displacements p_1 through p_5 each are measured by two opposite solenoids. For each pair, the relative difference may be computed. The results form a vector of measurements, y_1 through y_5, where y_i largely corresponds to p_i. Means for measuring p_6-motion are not shown in the figures. Depending on the application, it will either be missing or consist of a 4-solenoid, resolver-like setup.

Given the 5 translational quantities, $p_1 \cdots p_5$, and the rotational displacement, p_6, the computation of the displacement transformation ${}^{M}\boldsymbol{D}$ is straightforward. However, the complete expression is quite bulky, and will not be quoted here. For tracking purposes, i.e. to make the sensor housing follow the pin, only an approximation of ${}^{M}\boldsymbol{D}$ will be needed. From Figs. 3.1 and 3.3 it is seen that the first order

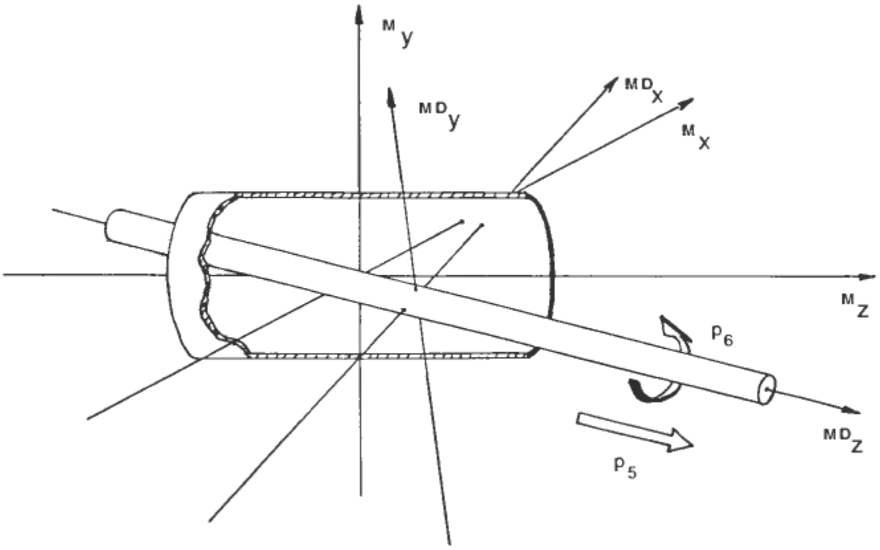

Figure 3.1. Coordinate systems of the two parts of the sensor.

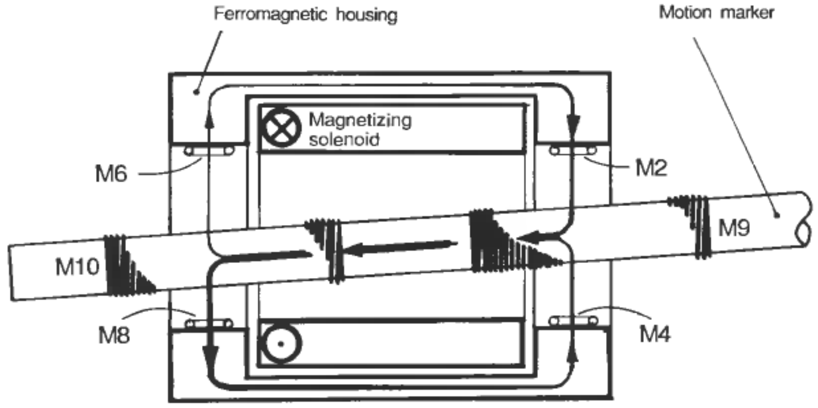

Figure 3.2. Measurements. Flux passing through each solenoid varies according to motion.

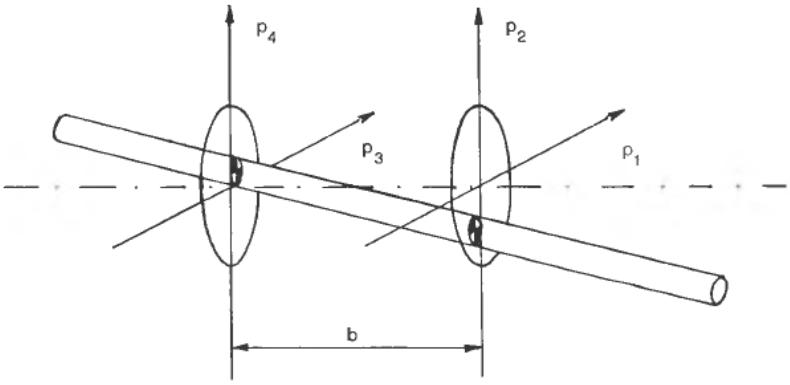

Figure 3.3. Internal radial deviation quantities.

expansion of ${}^{M}\boldsymbol{D}$ near ${}^{M}\boldsymbol{D} = \boldsymbol{I}$ is

$$ {}^{M}\boldsymbol{D} \approx \boldsymbol{I} + {}^{M}\Delta = \begin{bmatrix} 1 & -\delta_3 & \delta_2 & d_1 \\ \delta_3 & 1 & -\delta_1 & d_2 \\ -\delta_2 & \delta_1 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

where

$$ \begin{gathered} d_1 = \frac{p_4 + p_2}{2}; \quad d_2 = \frac{p_2 + p_4}{2}; \quad d_3 = p_5 \\ \delta_1 = \frac{p_4 - p_2}{b}; \quad \delta_2 = \frac{p_1 - p_3}{b}; \quad \delta_3 = p_6 \end{gathered} \tag{3.2} $$

These matrix elements also correspond to the elements of vector $\boldsymbol{d}$, in (2.5).

For recording purposes, the exact value of ${}^{M}\boldsymbol{D}$ must be computed. These computations again rely on the exactness of the sensor cutput model, i.e. how sensor output, $\boldsymbol{y}$, is related to the intermediate displacement vector, $\boldsymbol{p}$. This relationship is represented by mapping $\boldsymbol{Q}$: $R^6 \rightarrow R^6$ so that

$$ \boldsymbol{y} = \boldsymbol{Q}\boldsymbol{p} \tag{3.3} $$

During the design of the displacement sensor, much effort was made to make the mapping as linear as possible, and with little coupling between vector elements. The attempt was fairly successful, but in order to compute ${}^{M}\boldsymbol{D}$ exactly, both non-linearity and coupling must be taken into account.

The discussion of sensor equations will be restricted to the first four elements of $\boldsymbol{p}$. This is possible since, because of the symmetry, variations in p_5 or p_6 has no effect on the first four elements of the measurement vector, $\boldsymbol{y}$. The subsystem may be represented by a mapping $\tilde{\boldsymbol{Q}}: \boldsymbol{R}^4 \rightarrow \boldsymbol{R}^4$ so that

$$ \tilde{\boldsymbol{y}} = \tilde{\boldsymbol{Q}}\tilde{\boldsymbol{p}} \tag{3.4} $$

where $\tilde{\boldsymbol{y}} = [y_1, y_2, y_3, y_4]^T$ and $\tilde{\boldsymbol{p}} = [p_1, p_2, p_3, p_4]^T$. The mapping may in some applications be approximated by a scalar constant:

$$ \tilde{\boldsymbol{y}} = c_0 \tilde{\boldsymbol{p}} \tag{3.5} $$

A more precise diagonal matrix approximation includes factors depending on $\tilde{\boldsymbol{p}}$.

$$ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{a_1}{a_1^2 + a_2^2 + p_1^2 + p_2^2} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{3.6a} $$

$$ \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = \frac{a_1}{a_1^2 + a_2^2 + p_3^2 + p_4^2} \begin{bmatrix} p_3 \\ p_4 \end{bmatrix} \tag{3.6b} $$

Here a_1 denotes inner radius of the outer cylinder and a_2 radius of the tool marker. The factors represent amplification as a function of radial deviation, and are computed separately for each cross section in Fig. 3.3.

Even more accurate approximations include coupling between the two subsystems. However sensor calibration shows that the factors in (3.6) dominate. The introduction of coupling terms makes the inversion of (3.4) more difficult. Since the coupling is loose, however, the procedure may be simplified. Two approaches are now outlined:

1. Iterative solution: (3.4) may be inverted using a Newton–Raphson scheme. In this case, the inverse Jacobian matrix involved may be approximated by using (3.6.).
2. Approximate, symbolic solution: The two subsystems in (3.6) may be inverted separately. Afterwards, approximate inverse coupling terms may be inserted and adjusted using an offline curve fitting scheme.

The final shape of the approximation of (3.4) will depend on the need for accuracy, and the kinematical structure in each application. For instance, in the tracking system described in §§ 2 and 4, only (3.5) is used. On the other hand, situations may exist where extremely accurate expressions are necessary. These may partly be based on look-up tables rather than extensive calculation of the magnetic field.

4. Experimental control system

The sensor described in § 3 allows a radial pin displacement of ±1 cm, and an axial displacement of ±2 cm approx. The success of the complete system relies on the presence of a control system which makes the manipulator follow the tool closely at all times. This section deals with the design of the control system.

Before going into details, it is of interest to have a rough idea of the performance that may be expected. For this purpose we look at a simple 1 d.o.f. positional control system, as shown in Fig. 4.1. The controller may be of the proportional (P) or proportional + integral (PI) type. Asymptotical plots of the transfer function from positional reference to control error, $N(s) = e(s)/x_0(s)$, are shown in Fig. 4.2. It is assumed that the bandwidth ω_0 is feasible. By studying the asymptotical behaviour of $N(s)$ for $\omega < \omega_0$, it is seen that, with a proportional controller,

$$\lim_{s\to 0} \frac{1}{s} N(s) = \frac{1}{\omega_0} \tag{4.1}$$

and with a PI-controller,

$$\lim_{s\to 0} \frac{1}{s^2} N(s) = \frac{1}{\omega_0^2} \tag{4.2}$$

Since the range of the deviation sensor is limited to ±1 cm approximately, it is found that maximum velocity obtained when using a P-controller will be $v_P = 1$ cm ω_0. Under PI-control, a maximum acceleration of $a_{PI} = 1$ cm ω_0^2 is obtained. With a bandwidth of $\omega_0 = 10$ rad/s, which is typical for a hydraulically driven paint spraying robot, we obtain the values

$$v_P = 10 \text{ cm/s} \qquad a_{PI} = 100 \text{ cm/s}^2 \tag{4.3}$$

It is seen that the mere use of proportional control gives low performance. It is also worth noting that maximum acceleration under PI-control is proportional to ω_0^2,

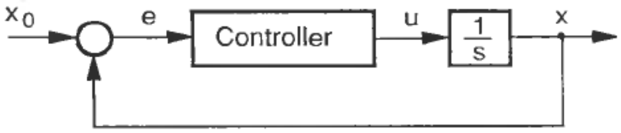

Figure 4.1. Simplified servo loop.

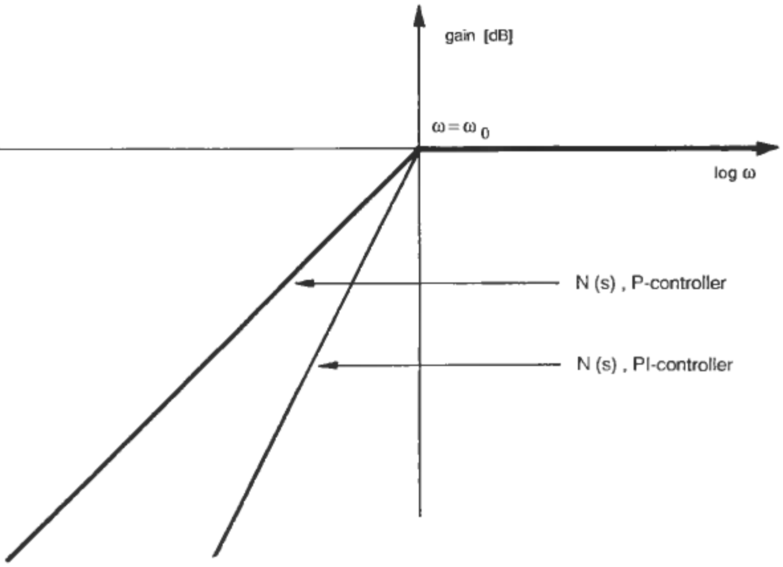

Figure 4.2. Results for different controllers.

which makes it worthwhile to increase ω_0 as much as possible. This will be considered in the next section.

The tool tracking system has so far been implemented on a Trallfa TR-400S paint spraying robot. The first version of the control scheme is designed to resemble the original, joint coordinate, positional control scheme as much as possible. This implies that the positional deviation measurements are converted into an equivalent vector, $\boldsymbol{e}$, of servo control errors. This vector can be used by the original controllers.

The conversion is done by using the transformations listed below:

$\boldsymbol{p} \leftarrow \boldsymbol{y}$	(3.5) extended
${}^{M}\boldsymbol{\Delta} \leftarrow \boldsymbol{p}$	(3.1) (3.2)
${}^{T}\boldsymbol{\Delta} \leftarrow {}^{M}\boldsymbol{\Delta}$	(2.2) (3.1)
$\Delta \boldsymbol{q} \leftarrow {}^{T}\boldsymbol{\Delta}$	simplified manipulator Jacobian
$\boldsymbol{e} \leftarrow \Delta \boldsymbol{q}$	simplified servo to joint transformation

The complete scheme has been implemented in integer assembly code on a TMS 99105 microprocessor. A sampling rate of 100 Hz is used, and the program is finished in less than 1 ms. Site tests indicate that the control scheme works, and that the performance corresponds to (4.3).

This first implementation also shows that it may not be wise to include the original joint controllers in the tool tracking scheme. This is due to the following:

1. Design criteria for the tool tracking system and the original play-back system differ. The original controllers are designed to follow a high speed reference trajectory without overshooting. This implies that the control error integrator is restricted, which in turn reduces the steady state acceleration (4.3). In the case of tool tracking, it is not so important whether the manipulator overshoots or not.

2. Optimal controllers would, in the two cases, differ because of differences in the underlying control performance indices. An optimal playback controller would be designed considering the characteristics of the tool used and of the task itself, whereas the tool tracking system must conform to the limitations of the displacement sensor used.
3. During playback, joint displacement is compared to a reference in order to compute the servo control error. During tool tracking, error signals are produced at the tip of the manipulator. Whenever manipulator link elasticities are significant, this may cause an additional stability problem (Cannon and Schmitz 1984).

The first point is easily solved by creating separate tool tracking controllers with higher error integration rates. Points 2 and 3 may be solved by considering more complex system models. This will in turn lead to more complex control schemes.

5. Control system improvements

Thus far the complete system for robot programming has been outlined, and the displacement sensor has been described in some detail. In § 4 the tool tracking system was described, and some notes were made on tracking performance. To conclude this paper, some suggestions will be made on how to improve the performance.

From the discussion leading to (4.3) it is evident that the steady state tracking speed or acceleration depends on sensor range and servo bandwidth. Hence control system improvements may be divided into two classes:

1. Optimizing the use of the sensor's 'workspace'
2. Maximizing the control system bandwidth

Besides actually increasing the sensor range, point 1 leads to the problem of designing a controller which minimizes some performance index

$$B = f({}^{M}\boldsymbol{d}) + g(\boldsymbol{x}, \boldsymbol{u}) \tag{5.1}$$

Here the functional f expresses the sensor's work space in some way, and g the effort made by the robot manipulator. It may be difficult to obtain an exact solution to the problem. However, sub-optimal controllers may be designed by modifying well known control schemes, such as the motion rate (Whitney, 1969) or the acceleration control (Luh, Walker and Paul, 1980a) schemes.

Increasing system bandwidth, as in point 2, involves considering more complex dynamic models for the system. The model may be expanded in three directions:

(*a*) Servo dynamics

(*b*) Manipulator kinetics

(*c*) Link and joint elasticities

Point (*b*) involves designing variable controllers. This may be done either by finding a means of making the existing controllers vary according to the manipulator configuration, or by applying complete schemes such as the 'computed torque method' (Luh, Walker and Paul, 1980b). According to (Cannon and Schmitz 1984), positional sensing at the manipulator wrist may lead to a special stability problem due to link elasticities. Calculations show that this reduces the bandwidth of the Trallfa

TR400 manipulator by a factor of 2 approx. The bandwidth may be increased by taking the effect of elasticity into account.

It is seen from the previous discussion that several approaches can be made in order to increase the performance of the robot programming system described. It is the authors' opinion that the system may soon be used for the real time teaching of high speed paint spraying robots. In this case, some, or all of the points made in this concluding section must be considered.

References

Balchen, J. G. (1986). Norwegian Patent Application 870419.

Cannon, R. H. and Schmitz, E. (1984). Precise control of flexible manipulators. *Robotics Research: The First International Symposium, New Hampshire*, August, pp. 841–862.

Dessen, F. (1988). Direct training of robots by a positional deviation sensor. Dr.ing.-thesis, Norwegian Inst. of Tech., Trondheim.

Hirzinger, G. (1982). Robot-teaching via force-torque-sensors. *6th Eur. Meeting on Cybernetics and Systems Research EMCSR 82, Vienna*, April 13–16.

Luh, J. Y. S., Walker, M. W. and Paul, R. P. C. (1980a). Resolved acceleration control of mechanical manipulators. *IEEE Trans. Autom. Control*, **25**, 468–474.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. (1980b). On-line computational scheme for mechanical manipulators. *J. Dynamic Systems, Measurement and Control*, **102**, 69–76.

Paul, R. P. C. (1981). *Robot Manipulators*. (MIT Press, Cambridge, MA).

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Trans. Man–Machine Syst.*, **10**, 47–53.