

# Full-Pose Calibration of a Robot Manipulator Using a Coordinate-Measuring Machine

Morris R. Driels, Lt W. Swayze USN and Lt S. Potter USN

Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California, USA

*The work reported in this article addresses the kinematic calibration of a robot manipulator using a coordinate measuring machine (CMM) which is able to obtain the full pose of the end-effector. A kinematic model is developed for the manipulator, its relationship to the world coordinate frame and the tool. The derivation of the tool pose from experimental measurements is discussed, as is the identification methodology. A complete simulation of the experiment is performed, allowing the observation strategy to be defined. The experimental work is described together with the parameter identification and accuracy verification. The principal conclusion is that the method is able to calibrate the robot successfully, with a resulting accuracy approaching that of its repeatability.*

**Keywords:** Robot calibration; Coordinate measurement; Parameter identification; Simulation study; Accuracy enhancement

## 1. Introduction

It is well known that robot manipulators typically have reasonable repeatability (0.3 mm), yet exhibit poor accuracy (10.0 mm). The process by which robots may be calibrated in order to achieve accuracies approaching that of the manipulator is also well understood [1]. In the calibration process, several sequential steps enable the precise kinematic parameters of the manipulator to be identified, leading to improved accuracy. These steps may be described as follows:

1. A kinematic model of the manipulator and the calibration process itself is developed and is usually accomplished with standard kinematic modelling tools [2]. The resulting model is used to define an error quantity based on a nominal (manufacturer's) kinematic parameter set, and an unknown, actual parameter set which is to be identified.

2. Experimental measurements of the robot pose (partial or complete) are taken in order to obtain data relating to the actual parameter set for the robot.
3. The actual kinematic parameters are identified by systematically changing the nominal parameter set so as to reduce the error quantity defined in the modelling phase. One approach to achieving this identification is determining the analytical differential relationship between the pose variables  $P$  and the kinematic parameters  $K$  in the form of a Jacobian,

$$\delta P = J \delta K \quad (1)$$

and then inverting the equation to calculate the deviation of the kinematic parameters from their nominal values

$$\delta K = [J^T J]^{-1} J^T \delta P \quad (2)$$

Alternatively, the problem can be viewed as a multidimensional optimisation task, in which the kinematic parameter set is changed in order to reduce some defined error function to zero. This is a standard optimisation problem and may be solved using well-known [3] methods.

4. The final step involves the incorporation of the identified kinematic parameters in the controller of the robot arm, the details of which are rather specific to the hardware of the system under study.

This paper addresses the issue of gathering the experimental data used in the calibration process. Several methods are available to perform this task, although they vary in complexity, cost and the time taken to acquire the data. Examples of such techniques include the use of visual and automatic theodolites [4, 5, 6], servocontrolled laser interferometers [7], acoustic sensors [8] and visual sensors [9]. An ideal measuring system would acquire the full pose of the manipulator (position and orientation), because this would incorporate the maximum information for each position of the arm. All of the methods mentioned above use only the partial pose, requiring more data to be taken for the calibration process to proceed.

Accepted for publication: 21 October 1991

Correspondence and offprint requests to: Prof. Morris R. Driels, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California 93943, USA.

## 2. Theory

In the method described in this paper, for each position in which the manipulator is placed, the full pose is measured, although several intermediate measurements have to be taken in order to arrive at the pose. The device used for the pose measurement is a coordinate-measuring machine (CMM), which is a three-axis, prismatic measuring system with a quoted accuracy of 0.01 mm. The robot manipulator to be calibrated, a PUMA 560, is placed close to the CMM, and a special end-effector is attached to the flange. Fig. 1 shows the arrangement of the various parts of the system. In this section the kinematic model will be developed, the pose estimation algorithms explained, and the parameter identification methodology outlined.

### 2.1 Kinematic Parameters

In this section, the basic kinematic structure of the manipulator will be specified, its relation to a user-defined world coordinate system discussed, and the end-point tool modelled. From these models, the kinematic parameters which may be identified using the proposed technique will be specified, and a method for determining those parameters described.

The fundamental modelling tool used to describe the spatial relationship between the various objects and locations in the manipulator workspace is the Denavit-Hartenberg method [2], with modifications proposed by Hayati [10], Mooring [11] and Wu [12] to account for disproportional models [13] when two consecutive joint axes are nominally parallel. As shown in Fig. 2, this method places a coordinate frame on

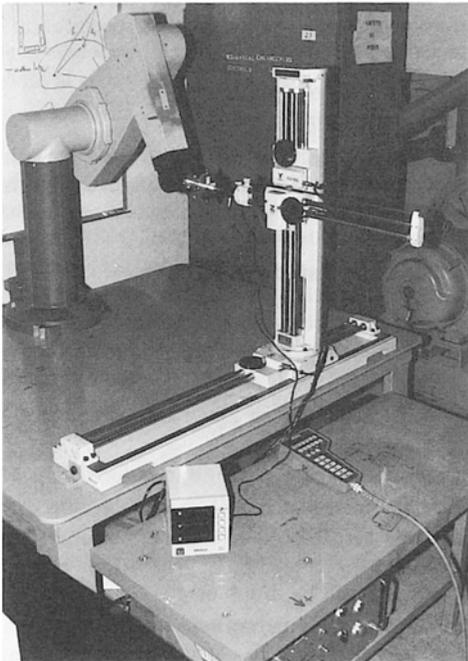


Fig. 1. Calibration equipment.

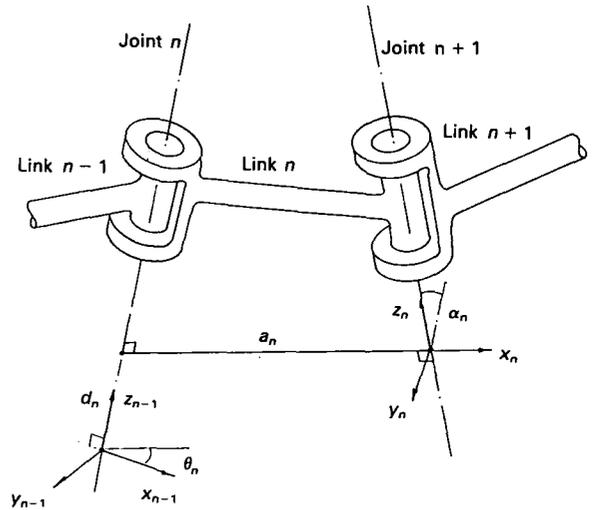


Fig. 2. Link coordinate frame allocation.

each object or manipulator link of interest, and the kinematics are defined by the homogeneous transformation required to change one coordinate frame into the next. This transformation takes the familiar form

$$A_n = \text{rot}(z, \theta_n) \text{trans}(z, d_n) \text{trans}(x, a_n) \text{rot}(x, \alpha_n) \text{rot}(y, \beta_n) \quad (3)$$

The above equation may be interpreted as a means to transform frame  $n-1$  into frame  $n$  by means of four out of the five operations indicated. It is known that only four transformations are needed to locate a coordinate frame with respect to the previous one. When consecutive axes are not parallel, the value of  $\beta_n$  is defined to be zero, while for the case when consecutive axes are parallel,  $d_n$  is the variable chosen to be zero.

When coordinate frames are placed in conformance with the modified Denavit-Hartenberg method, the transformations given in the above equation will apply to all transforms of one frame into the next, and these may be written in a generic matrix form, where the elements of the matrix are functions of the kinematic parameters. These parameters are simply the variables of the transformations: the joint angle  $\theta_n$ , the common normal offset  $d_n$ , the link length  $a_n$ , the angle of twist  $\alpha_n$ , and the angle  $\beta_n$ . The matrix form is usually expressed as follows:

$$A_n = \begin{bmatrix} C\theta_n C\beta_n - S\theta_n S\alpha_n S\beta_n & -S\theta_n C\alpha_n & C\theta_n S\beta_n + S\theta_n S\alpha_n C\beta_n & a_n C\theta_n \\ S\theta_n C\beta_n + C\theta_n S\alpha_n S\beta_n & C\theta_n C\alpha_n & S\theta_n S\beta_n - C\theta_n S\alpha_n C\beta_n & a_n S\theta_n \\ -C\alpha_n S\beta_n & S\alpha_n & C\alpha_n C\beta_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

For a serial linkage, such as a robot manipulator, a coordinate frame is attached to each consecutive link so that both the instantaneous position together with the invariant geometry are described by the previous matrix transformation. The

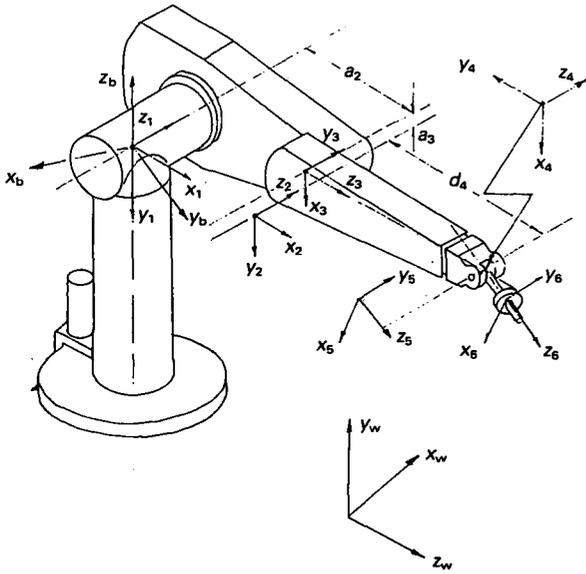


Fig. 3. PUMA frame allocation.

transformation from the base link to the  $n$ th link will therefore be given by

$$T_n = A_1 A_2 \dots A_n \quad (5)$$

Fig. 3 shows the PUMA manipulator with the Denavit-Hartenberg frames attached to each link, together with world coordinate frame and a tool frame. The transformation from the world frame to the base frame of the manipulator needs to be considered carefully, since there are potential parameter dependencies if certain types of transforms are chosen. Consider Fig. 4, which shows the world frame  $x_w, y_w, z_w$ , the frame  $x_0, y_0, z_0$  which is defined by a DH transform from the world frame to the first joint axis of the manipulator, frame  $x_b, y_b, z_b$ , which is the PUMA

manufacturer's defined base frame, and frame  $x_1, y_1, z_1$  which is the second DH frame of the manipulator. We are interested in determining the minimum number of parameters required to move from the world frame to the frame  $x_1, y_1, z_1$ . There are two transformation paths that will accomplish this goal:

*Path 1:* A DH transform from  $x_w, y_w, z_w$ , to  $x_0, y_0, z_0$  involving four parameters, followed by another transform from  $x_0, y_0, z_0$  to  $x_b, y_b, z_b$  which will involve only two parameters  $\phi'$  and  $d'$  in the transform

$$T_0^b = \text{rot}(z_0, \phi') \text{trans}(z_0, d') \quad (6)$$

Finally, another DH transform from  $x_b, y_b, z_b$  to  $x_1, y_1, z_1$  which involves four parameters except that  $\Delta\theta_1$  and  $\phi'$  are both about the axis  $z_0$  and cannot therefore be identified independently, and  $\Delta d_1$  and  $d'$  are both along the axis  $z_0$  and also cannot be identified independently. It requires, therefore, only eight independent kinematic parameters to go from the world frame to the first frame of the PUMA using this path.

*Path 2:* As an alternative, a transform may be defined directly from the world frame to the base frame  $x_b, y_b, z_b$ . Since this is a frame-to-frame transform it requires six parameters, such as the Euler form:

$$A_b = \text{rot}(z, \phi_b) \text{rot}(y, \theta_b) \text{rot}(x, \psi_b) \text{trans}(p_{xb}, p_{yb}, p_{zb}) \quad (7)$$

The following DH transform from  $x_b, y_b, z_b$  to  $x_1, y_1, z_1$  would involve four parameters, but  $\Delta\theta_1$  may be resolved into  $\phi_b, \theta_b, \psi_b$ , and  $\Delta d_1$  resolved into  $p_{xb}, p_{yb}, p_{zb}$ , reducing the parameter count to two. It is seen that this path also requires eight parameters as in path 1, but a different set.

Either of the above methods may be used to move from the world frame to the second frame of the PUMA. In this work, the second path is chosen. The tool transform is an Euler transform which requires the specification of six parameters:

$$A_6 = \text{rot}(z, \phi_6) \text{rot}(y, \theta_6) \text{rot}(x, \psi_6) \text{trans}(p_{x6}, p_{y6}, p_{z6}) \quad (8)$$

The total number of parameters used in the kinematic model becomes 30, and their nominal values are defined in Table 1.

## 2.2 Identification Methodology

The kinematic parameter identification will be performed as a multidimensional minimisation process, since this avoids the calculation of the system Jacobian. The process is as follows:

1. Begin with a guess set of kinematic parameters, such as the nominal set.
2. Select an arbitrary set of joint angles for the PUMA.
3. Calculate the pose of the PUMA end-effector.
4. Measure the actual pose of the PUMA end-effector for the same set of joint angles. In general, the measured and predicted pose will be different.
5. Modify the kinematic parameters in an orderly manner in order to best fit (in a least-squares sense) the measured pose to the predicted pose.

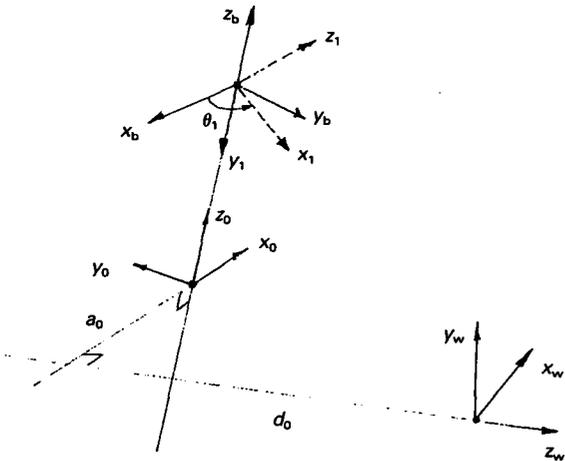


Fig. 4. Base transformations.

**Table 1.** Nominal parameters for the PUMA robot.

$\phi_b$ ( $^\circ$ )	$\theta_b$ ( $^\circ$ )	$\psi_b$ ( $^\circ$ )	$p_{xb}$ (mm)	$p_{yb}$ (mm)	$p_{zb}$ (mm)
180.0	0.0	90.0	-394.0	-383.0	474.0

Link	$\delta\theta_i$ ( $^\circ$ )	$d_i$ (mm)	$a_i$ (mm)	$\alpha_i$ ( $^\circ$ )	$\beta_i$ ( $^\circ$ )
1	0	0	0.0	-90.0	0
2	0.0	0	431.85	0.0	0.0
3	0.0	149.09	-20.33	90.0	0
4	0.0	433.0	0.0	-90.0	0
5	0.0	0.0	0.0	90.0	0

$\phi_b$ ( $^\circ$ )	$\theta_b$ ( $^\circ$ )	$\psi_b$ ( $^\circ$ )	$p_{x6}$ (mm)	$p_{y6}$ (mm)	$p_{z6}$ (mm)
90.0	0.0	0.0	0.0	0.0	134.0

The process is applied not to a single set of joint angles but to a number of joint angles. The total number of joint angle sets required, which also equals the number of physical measurement made, must satisfy

$$K_p \geq N \times D_f \quad (9)$$

where

$K_p$  is the number of kinematic parameters to be identified  
 $N$  is the number of measurements (poses) taken  
 $D_f$  represents the number of degrees of freedom present in each measurement

In the system described in this paper, the number of degrees of freedom is given by

$$D_f = 6 \quad (10)$$

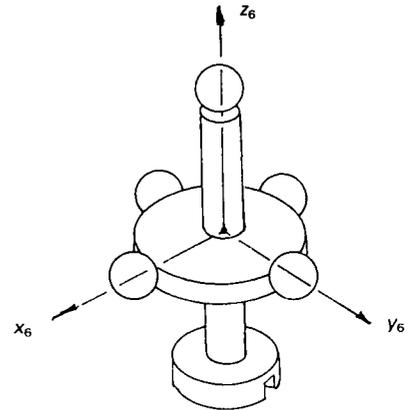
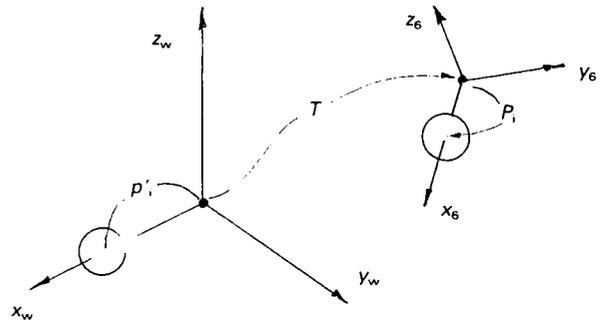
since full pose is measured. In practice, many more measurements should be taken to offset the effect of noise in the experimental measurements. The optimisation procedure used is known as ZXSSQ, and is a standard library function in the IMSL package [14].

### 2.3 Pose Measurement

It is apparent from the above that a means to determine the full pose of the PUMA is required in order to perform the calibration. This method will now be described in detail. The end-effector consists of an arrangement of five precision-tooling balls as shown in Fig. 5. Consider the coordinates of the centre of each ball expressed in terms of the tool frame (Fig. 5) and the world coordinate frame, as shown in Fig. 6. The relationship between these coordinates may be written as

$$\mathbf{P}'_i = T\mathbf{P}_i \quad (11)$$

where  $\mathbf{P}'_i$  is the  $4 \times 1$  column vector of the coordinates of


**Fig. 5.** Tool transform.

**Fig. 6.** Ball centroid in the tool and world frames.

the  $i$ th ball expressed with respect to the world frame,  $\mathbf{P}_i$  is the  $4 \times 1$  column vector of the coordinates of the  $i$ th ball expressed with respect to the tool frame, and  $T$  is the  $4 \times 4$  homogenous transform from the world frame to the tool frame.

If  $\mathbf{P}_i$  is known by precalibrating the tool, and  $\mathbf{P}'_i$  is measured, then  $T$  may be found, and used as the measured pose in the calibration process. It is not quite that simple, however, since it is not possible to invert equation (11) to obtain  $T$ . The above process is performed for the four balls, A, B, C and D, and the positions ordered as

$$\mathbf{P}'_A \mathbf{P}'_B \mathbf{P}'_C \mathbf{P}'_D = T[\mathbf{P}_A \mathbf{P}_B \mathbf{P}_C \mathbf{P}_D] \quad (12)$$

or in the form

$$\mathbf{P}' = T\mathbf{P} \quad (13)$$

Since  $\mathbf{P}'$ ,  $T$  and  $\mathbf{P}$  are all now square, the pose matrix may be obtained by inversion

$$T = \mathbf{P}' \mathbf{P}^{-1} \quad (14)$$

In practice it may be difficult for the CMM to access four balls to determine  $\mathbf{P}_i$  when the PUMA is placed in certain configurations. Three balls are actually measured and a fourth ball is fictitiously located according to the vector cross product

$$\mathbf{P}_4 = (\mathbf{P}_3 - \mathbf{P}_1) \times (\mathbf{P}_2 - \mathbf{P}_1) \quad (15)$$

Regarding the determination of the coordinates of the

centre of a ball based on measured points on its surface, no analytical procedures are available. Another numerical optimisation scheme was used for this purpose such that the penalty function

$$F_i = (u - x_i)^2 + (v - y_i)^2 + (w - z_i)^2 - r^2 \quad (16)$$

was minimised, where  $(u, v, w)$  are the coordinates of the centre of the ball to be determined,  $(x_i, y_i, z_i)$  are the coordinates of the  $i$ th point on the surface of the ball and  $r$  is the ball diameter. In the tests performed, it was found sufficient to measure only four points ( $i = 4$ ) on the surface to determine the ball centre.

### 3. Simulation

In performing a simulation study of the proposed experimental calibration process the objectives are as follows:

To confirm that the numerical algorithm proposed for the identification converges to the correct values

To determine the number of experimental data needed to identify the kinematic parameters to a required degree of accuracy

To estimate the resulting accuracy of the manipulator if the new kinematic model were to be embedded in the robot control computer.

Several computer programs were written to perform this simulation study. These programs and their relationship to each other are shown in Fig. 7. The first of these, POSE, simulates the generation of sets of PUMA joint angles using a random-number function. These values are then used in a forward kinematic solution to determine the pose. The kinematic parameter table used in this model includes known deviations from the PUMA nominal model. Since the ultimate objective of the calibration is to identify these parameters, the resulting joint angle sets and corresponding poses are

pseudo-experimental data. At this stage, estimates of the measurement noise are added to the data. The PUMA manufacturers indicate that joint angles may be read reliably to two decimal places, so they are stored in the output data file (PUMA-VAR.DAT) to this level of accuracy. The other uncertainty in the experiment is the precision of the CMM in determining the position of the centre of the measured balls. This is added to the upper right  $3 \times 1$  position component of the pose matrix, and takes the value of 0.01 mm, the manufacturer's quoted accuracy for the CMM.

Program POSE is run twice, to produce a joint angle data set to be used in the identification process (PUMA-VAR.DAT), and another set (POSEVER.DAT) to be used in a verification phase. Program ID6 performs the actual identification, and uses the ZXSSQ subroutine mentioned before. Convergence may be defined by a combination of several criteria, although in this case it was decided to halt iteration when the kinematic parameters were consistent to four significant figures from one iteration to the next. The output of ID6 is a file (RESULT.DAT) containing the 30 identified parameters, together with the RMS difference between the actual and identified parameters. This latter quantity is calculated as the aggregate length and angular parameter errors and indicates the accuracy of identification.

Finally, the second set of joint angle data (POSEVER.DAT) is used in a forward kinematic solution of the PUMA, together with the identified parameters, to estimate the accuracy that the manipulator would achieve if these identified parameters were to replace the nominal parameters in the robot control computer. This is accomplished in program VERIFY by simply calculating two forward kinematic solutions based on the actual and identified kinematic parameter data sets and calculating the differential displacement and orientation matrix [15]

$$\Delta = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

to calculate the position and orientation accuracy of the calibrated manipulator.

The complete simulation may be regarded as a tool to plan the experiments in which the independent variables are the number of observations and the range of joint angles allowed by the common work volume of the PUMA and the CMM, while the dependent variables are the accuracy of parameter identification and the resulting manipulator accuracy. Figs 8 and 9 show the variation in the position and orientation accuracy respectively, as a function of the number of observations used in the identification. As expected, the error decreases with increasing observations.

The results of the simulation study indicate that the calibration process is characterised by robust convergence, requires relatively few experimental observations, and will result in a manipulator accuracy of the same order as its repeatability. With this knowledge, the experimental phase of the work was performed.

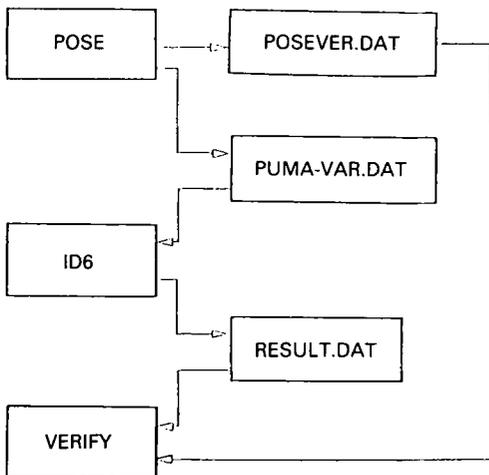


Fig. 7. Programs used in the simulation of the calibration process.

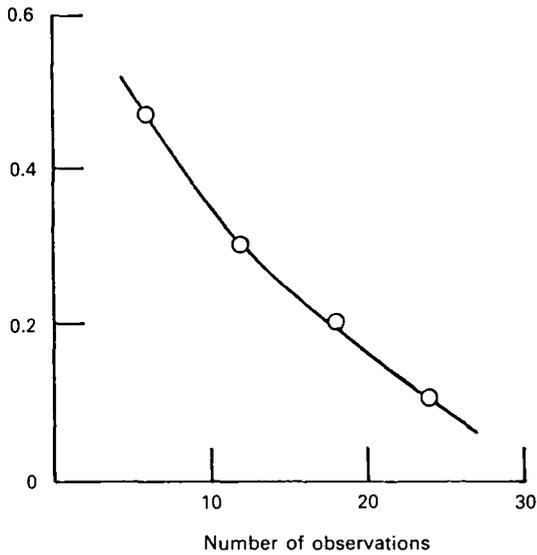


Fig. 8. Simulated position error.

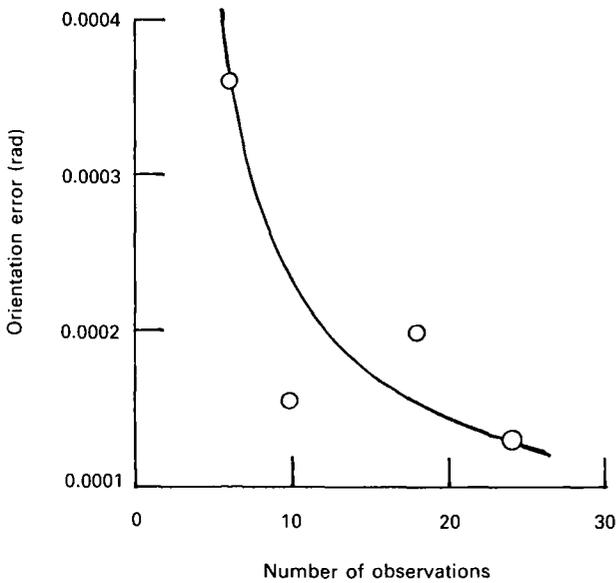


Fig. 9. Simulated orientation error.

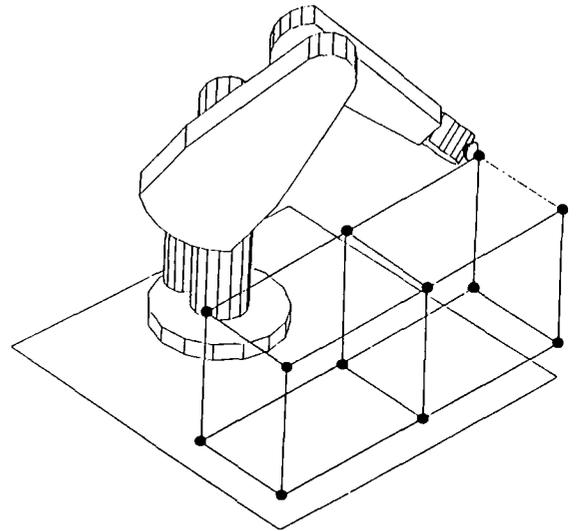


Fig. 10. Experimental pose locations.

to 48 possible poses. Each wrist joint angle was exercised as much as possible without causing a limit stop failure by the PUMA controller.

For each pose, a ball on the tool was identified and the CMM measured 4 locations on the surface of the ball and the experimental points. This value should be very small, and large values ( $\geq 0.001$  mm) indicated errors in supplying the program with CMM data. In this case the data was measured again. Using the method described in the theory section, the pose of the end-effector was calculated once data for 3 balls had been collected. Once the  $T$  matrix from equation (11) had been calculated, its validity as a homogeneous transform was checked by recognising that if

$$T = [n \ o \ a \ p] \tag{18}$$

then

$$n \cdot n = o \cdot o = a \cdot a = 1 \tag{19}$$

and

$$n = o \times a \tag{20}$$

$$o = a \times n \tag{21}$$

The program checks the size and orthogonality conditions stated above, and if they are within acceptable limits then the pose is saved, together with corresponding joint angles obtained from the PUMA controller. The calibration required two people to take the data and type in the results, at a rate of about 10 minutes per pose.

### 5. Results and Conclusions

The experimental data file is in the same format as the pseudo-data file PUMA-VAR.DAT shown in Fig. 7, and may

### 4. Experimental Work

In order to maximise the joint motion for the experimental poses, the common working volume of the PUMA and the CMM was regarded as a parallelepiped, as shown in Fig. 10. 12 points were located on the parallelepiped where pose measurements were to be taken. It is assumed that points distributed widely in the world space will result in large ranges of joint angles. The tool was located at each point in the lefty, righty, elbow-up and elbow-down configurations, leading

therefore be used directly in the identification routine ID6. Because of errors found in the experimental data, only 42 of the 48 measured poses were actually used.

This data set was divided into two 21-pose sets, where the first set was used in the identification while the second was used as an independent set with which to determine the resulting accuracy at poses other than those used for the identification. In order to compare with the simulation studies, sets of 7, 14 and 21 randomly selected data points were selected from the identification data set in order to determine the kinematic parameters. The identified parameters were then used in the verification set to determine the error between the measured pose and the pose calculated using the identified parameters. In this calculation, all 21 verification poses were used and the RMS error for position and orientation were calculated. The results are shown in Figs 11 and 12. It is seen that the simulation and experimental data predict the same trend, although the experimentally predicted accuracy is considerably larger than that predicted. This may be attributed to a larger measurement noise value than that assumed for the simulations. Tests showed that increasing the magnitude of noise resulted in poorer accuracy of identification and hence a less accurately calibrated robot.

The data shown in the simulation for the orientation error (Fig. 9) illustrate an interesting issue. The data suggests that the accuracy for 10 observations is less than that for 18 observations. Recall that each observation is a pose calculated for a randomly selected set of joint angles. It may be expected that some joint angles would lead to a "better" identification than another set, for example those selected near a kinematic

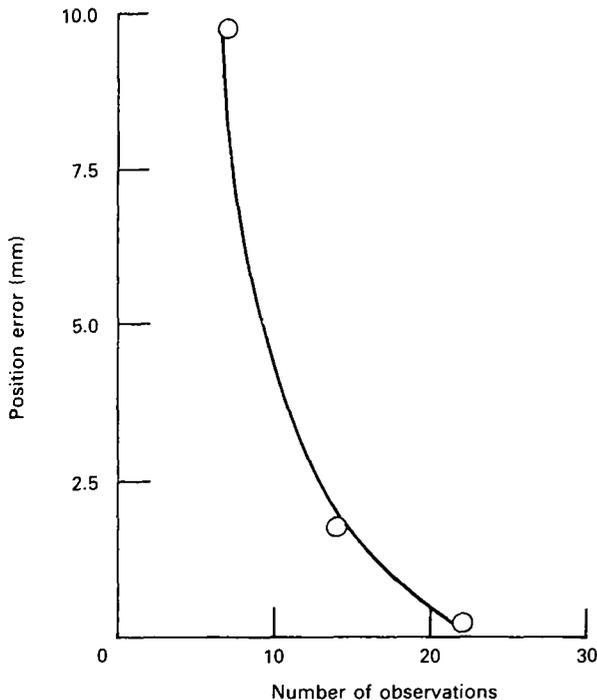


Fig. 11. Calibrated accuracy - position.

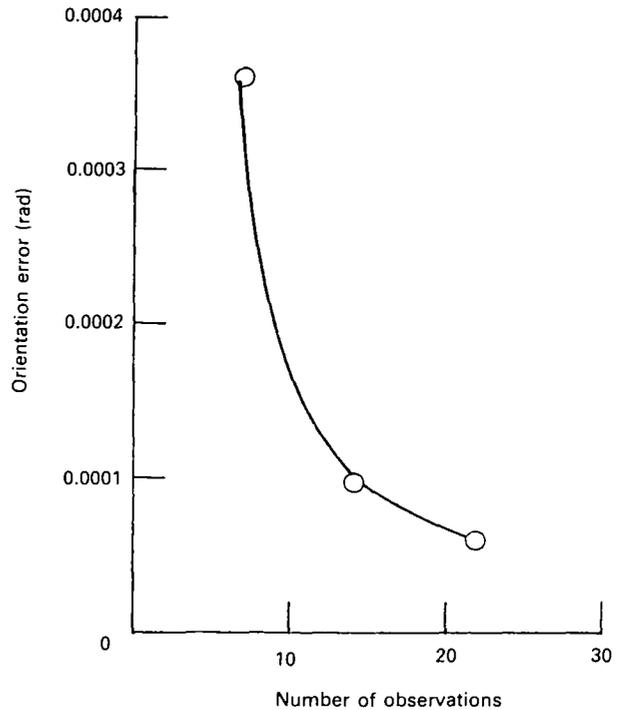


Fig. 12. Calibrated accuracy - orientation.

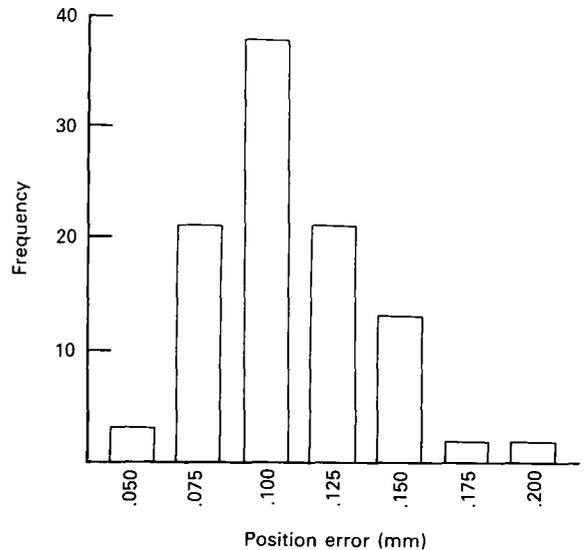


Fig. 13. Position error for 100 randomly selected joint angle sets.

singularity. The predicted accuracy would be variable, therefore, even for a fixed number of observations. The simulation may be used to illustrate this by performing 100 calibrations, each using a randomly selected set of 12 observations. Fig. 13 shows the results of this test, and indicate that there is a wide variation in resulting accuracy. All data points shown in Figs 8 and 9 should be regarded as having such distributions. Another implication of this test is that the experimental

Table 2. Nominal and identified kinematic parameters.

Parameter	Nominal value	Identified value
$\phi_b$	180.0	179.957 9
$\theta_b$	0.0	1.512 0
$\psi_b$	90.0	89.021 9
$p_{xb}$	-394.0	-393.983 8
$p_{yb}$	-383.0	-405.060 8
$p_{zb}$	474.0	466.838 1
$a_1$	0.0	-0.049 23
$\alpha_1$	-90.0	-89.997 7
$\delta\theta_2$	0.0	-0.488 8
$a_2$	431.9	432.121 6
$\alpha_2$	0.0	-0.030 3
$\beta_2$	0.0	-0.015 15
$\delta\theta_3$	0.0	-1.206 9
$d_3$	149.1	149.145 5
$a_3$	-20.3	-19.227 0
$\alpha_3$	90.0	90.051 2
$\delta\theta_4$	0.0	-0.914 4
$d_4$	433.0	432.889 9
$a_4$	0.0	0.004 0
$\alpha_4$	-90.0	-89.990 9
$\delta\theta_5$	0.0	2.236 4
$d_5$	0.0	-0.662 9
$a_5$	0.0	-0.025 8
$\alpha_5$	90.0	89.934 5
$\phi_6$	90.0	91.240 0
$\theta_6$	0.0	-0.097 9
$\psi_6$	0.0	-0.057 5
$p_{x6}$	0.0	0.186 3
$p_{y6}$	0.0	-0.232 9
$p_{z6}$	134.0	133.155 7

accuracy values are also expected to exhibit a similar statistical distribution, suggesting that for a fixed number of observations, a particular selection of joint angles leads to an accuracy several times better than that obtained from another set of joint angles. Since the acquisition of experimental calibration data is a time-consuming operation, the identification of these "optimum" joint angles would be of considerable value. Current research is addressing this issue.

Table 2 shows the kinematic parameters identified using the complete set of 42 poses. These values should be closer to those obtained using 21 poses, although this cannot be verified because all poses were used in the identification.

The results of the kinematic identification indicates that the resulting accuracy of the manipulator is of the order of 0.3 mm, and it may be concluded therefore that calibration using this method will improve the accuracy to about the same value as the manipulator repeatability.

## References

1. B. Mooring, Z. Roth and M. Driels, *Fundamentals of Manipulator Calibration*, John Wiley, New York, 1991.
2. J. Denavit and R. Hartenberg, "A kinematic notation for lower pair mechanisms based on matrices", *ASME Journal of Applied Mechanics*, 77, pp. 215-221, June 1955.
3. G. Dahlquist, A. Bjorck and N. Anderson, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
4. D. Whitney, C. Lozinsky and J. Rourke, "Industrial robot forward calibration method and results", *ASME Journal of Dynamic Systems, Measurement and Control*, 108, pp. 1-8, March 1986.
5. J. Jarvis, "Microsurveying: towards robot accuracy", Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, USA, pp. 1660-1665, March-April, 1987.
6. M. Driels and U. Pathre, "Vision based automatic theodolite for robot calibration", *IEEE Journal of Robotics and Automation*, 7, (3), pp. 351-360, June 1991.
7. K. Lau, R. Hocken and L. Haynes, "Robot performance measurements using automatic laser tracking techniques", *Robotics and Computer Integrated Manufacturing*, 2, pp. 227-236, 1985.
8. H. Stone and A. Sanderson, "A prototype arm signature identification system", Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, USA, pp. 175-182, March-April 1987.
9. I. Foulloy and R. Kelley, "Improving the precision of a robot", Proceedings, 1984 IEEE International Conference on Robotics and Automation, pp. 62-67, March 1984.
10. S. Hayati, "Robot arm geometric link parameter estimation", Proceedings 22nd IEEE Conference on Decision and Control, San Antonio, Tex., pp. 1477-1483, December 1983.
11. B. Mooring, "The effect of joint axis misalignment on robot positioning accuracy", Proceedings, ASME 1983 International Computers in Engineering Conference, Chicago, Ill., pp. 151-155, 1983.
12. C. Wu, "The kinematic error model for the design of robot manipulators", Proceedings, American Control Conference, San Francisco, Calif., pp. 497-502, 1983.
13. L. Everett, T. Hsu, "The theory of kinematic parameter identification for industrial robots", *ASME Journal of Dynamic Systems, Measurement and Control*, 110, pp. 96-99, 1988.
14. *IMSL Math/Library Users Manual*, Version 1.1, IMSL Inc., Houston, Tex., 1989.
15. R. Paul, *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, Cambridge, Mass., 1982.