COMPLETE PARAMETER IDENTIFICATION OF A ROBOT FROM PARTIAL POSE INFORMATION

Ambarish Goswami
Arthur Quaid
Michael Peshkin

Department of Mechanical Engineering
Northwestern University
Evanston IL 60208

Abstract

The absolute accuracy of a robot depends to a large extent on the accuracy with which its kinematic parameters are known. Many methods have been explored for inferring the kinematic parameters of a robot from measurements taken as it moves. Some require an external global positioning system, usually optical or sonic. We have used instead a simple radial-distance linear transducer (LVDT) which measures the distance from a fixed point in the workspace to the robot's endpoint. This incomplete pose information is accumulated as the robot endpoint is moved within one or more spherical "shells" centered about the fixed point. Optimal values for all of the independent kinematic parameters of the robot can then be found.

Here we discuss the motivation, theory, implementation, and performance of this particularly easy calibration and parameter identification method. We also address a recent disagreement in the literature about the class of measurements needed to fully identify a robot's kinematic parameters.

1.0 INTRODUCTION

It is important to distinguish between the absolute accuracy and the repeatability of a robotic manipulator. Repeatability of a robot is the precision with which its endpoint achieves a particular pose[1] under repeated commands of the same set of joint angles. Gear backlash, and sensor and servo precision are some of the factors affecting robot repeatability.

Absolute accuracy represents the closeness with which the robot's actual pose matches the pose predicted by its controller. A robot may have high repeatability while having low absolute accuracy. Given the joint angles, the controller of a robot computes its endpoint pose with respect to a coordinate frame attached to its base. For this it needs an accurate description of the robot which involves many physical parameters such as link lengths and joint offsets. These parameters make up the kinematic model of the robot and the absolute accuracy of the robot depends on the accuracy of this model.

A high repeatability is of prime importance for a variety of robot applications such as pick and place, spray painting, and welding. In these operations a robot is guided through the required endpoint motions (with the help of a teach-pendant) and the corresponding joint-angles are recorded. During actual operation, the robot "plays back" the recorded joint angles.

Tasks involving off-line programming, on the other hand, crucially depend on the absolute accuracy, in addition to the repeatability, of the robot. Accuracy issues arise for us in the context of employing a robotic system as a precision positioning device in a medical operating room. A robot may potentially improve the overall accuracy, and hence, the quality of an orthopedic surgery by guiding the surgeon to different preoperatively-selected poses. The poses are selected on the basis of the patient's relevant geometrical features (such as bone dimensions) presurgically obtained through CT scans. Absolute accuracy is very important for the effective implementation of these off-line manipulation strategies. See [25] and [14] for a description of our robot-assisted knee surgery project.

For various practical reasons, the actual values of some of the kinematic parameters of a robot may not be available. This may happen due to manufacturing errors, unaccounted for deviations such as link and joint compliance, and time-dependent effects such as gear wear and component damage. Therefore, the nominal kinematic model of the controller, in actuality, is an inaccurate representation of the robot. This adversely affects the absolute accuracy of the robot.

A practical approach to address this problem is to re-evaluate the parameters of a robot by using a calibration scheme. Robot calibration typically involves externally measuring the position and orientation of a robot's endpoint at different commanded locations in its workspace and comparing these measured values to those predicted by its controller. The difference between the actual (measured) values and the predicted values is the error. The existing kinematic model is then replaced by a new "optimal" model which minimizes the aggregate error.

Several robot calibration techniques have been reported in the literature. These approaches differ in the hardware employed for external measurements, the type of collected data (position data, orientation data, etc.) and in the nature of the optimization algorithm that generates the model. Among the most popular measurement systems are theodolites, lasers, sonic digitizers, proximity sensors, and three cable measuring systems. (See [15] for a description of common measurement devices.) The selection of a particular system is based upon criteria such as cost, ease of use, set-up time, accuracy of measurements, and volume of calibrated workspace.

2.0 SCOPE AND CONTRIBUTION

Robot calibration, in its broadest scope, addresses static errors (such as those caused by changes in link dimensions, gear wear, elastic bending of links, etc.) as well as dynamic errors (such as those caused by vibration) [20]. The static errors of a manipulator may result from geometric errors as well as from non-geometric errors. Geometric errors are related to link dimensions and joint transducer characteristics. The non-geometric errors are caused by effects such as gear train compliance, motor bearing wobble, and gear backlash. In this paper we compensate the static geometrical errors in a robot by re-evaluating its kinematic parameters in a numerical optimization scheme. With a suitable model, our approach can be extended to include non-geometric parameters. Since about 90% of the RMS values of the endpoint positioning error are caused by errors in the static geometric parameters [13],, the

[1] Pose is defined as the position and attitude of one coordinate frame w.r.t. another. Position includes a 3-vector of translational parameters and attitude includes a 3-vector of angular parameters.

1050-4729/93 $3.00

additional improvement in robot accuracy resulting from the compensation of non-geometric parameter errors will be relatively small.

The measuring device we use is known as a *telescopic ball-bar system* manufactured by Automated Precision Inc. It is relatively inexpensive, easy to use, and highly accurate [15, 24]. The heart of the system is a simple linear transducer (an LVDT). The ball-bar precisely measures the distance of the robot endpoint from a fixed location. The system set-up is shown in Fig. 1.

The ball-bar has a magnetic chuck permanently mounted at one end, and a removable high precision steel ball mounted at the opposite end. This removable ball end allows the attachments of extension bars. Extension bars permit the nominal length of the ball-bar to be easily increased in order to reach more of the robot's workspace if desired. Additional magnetic chucks and precision balls mate with the ball-bar ends to form spherical joints. In our application, the ball end of the ball-bar pivots around one of three stationary magnetic chucks mounted in the workspace while the magnetic chuck end mates with one of the three precision balls connected to the robot's moving endpoint.

Figure 1. The calibration system with the ball-bar connected between two fixtures mounted on the robot endpoint and the robot table.

The calibration system must read robot joint positions and ball-bar lengths for various robot poses within the workspace of the robot and the reach of the ball-bar. Then, using forward kinematics with the nominal model parameters, the expected position of the robot endpoint can be calculated from the joint positions. This position gives us the expected location of the robot end of the ball-bar. The distance between the two ends is then computed to get the expected ball-bar length. The difference between this length and the actual length read from the ball-bar is the error due to incorrect robot parameters. We compute new model parameters which minimize this error.

The maximum number of independent parameters, N, in a robot is given by the equation [4, 5, 6, 11][2]

$$N = 5n_r + 3n_p + 6 \ , \qquad (1)$$

where n_r is the number of revolute joints and n_p is the number of prismatic joints in the robot. Regardless of the total number of physical parameters in the kinematic description of the robot, N is the maximum number of parameters that can be identified by collecting data at the robot's endpoint[3] alone.

Robot endpoint data consists of its position (three measurements) and its attitude (three measurements) with respect to base coordinate frame. In order to identify all of the independent parameters of a robot, one should be able to obtain the position and attitude data of the robot endpoint thereby requiring the use of a sophisticated measuring device. We show in this paper that a simple measuring device (capable of collecting only partial pose information) used with a specialized data collection scheme is equally capable of identifying all of the independent robot parameters. This represents an economic and easy road to robot calibration.

Choice of a kinematic model is important for calibration; we discuss it in the next section. Section 4.0 discusses the optimization scheme adopted in this work. We use the Levenberg-Marquardt algorithm for its robustness and ability to handle singular systems. In Section 5.0 we detail the data collection procedure. The results are discussed in Section 6.0. Here we show that all the independent parameters of the robot are identified.

3.0 THE KINEMATIC MODEL

The endpoint pose of a robotic manipulator relative to a base coordinate frame may be expressed in terms of nonlinear mathematical functions involving the link parameters and the joint parameters, which are constants (these parameters constitute the kinematic model), and the joint variables (e.g., angles for revolute joints and displacements for prismatic joints). Many kinematic models have been suggested. See [12] for an overview of various kinematic models in use. One has to pay particular attention while selecting a kinematic model for calibration purposes. For instance, a model which is compact and efficient for forward and inverse kinematics may give rise to numerical instability during parameter estimation. See [4] and [5] for the desirable properties of a kinematic model.

For calibration purposes, the two most important properties of a kinematic model are *completeness* and *proportionality* [4]. In a *complete* kinematic model a change in any one of the physical features (such as one of the link length components) is reflected by one and only one model parameter. In an incomplete model the unmodeled errors tend to be compensated by physically unrelated parameters. This may result in an optimized parameter set that does not reflect the actual values of the robot's physical features. Worse, the robot may not achieve the best possible accuracy.

Proportionality of a kinematic model requires that a small change in the physical features of a robot is reflected by a small

[2]The original equation given in [4, 5, 6, 11], where joint encoders are not modeled, is of the form $N = 4n_r + 2n_p + 6$. Each joint encoder contributes one independent parameter.

[3]The total number of parameters required to model a robot depends on the particular kinematic model used.

change in the model parameters. Proportionality corresponds to the continuity of the model parameters and ensures the numerical stability of the optimization process.

In this work we adopt the Sheth-Uicker kinematic model [1, 21, 22]. This model is complete, proportional, and easy to assign to a manipulator. In this model three translational and three angular parameters (*ZYX*-ordered Euler angles) are used to describe the position and attitude of the distal coordinate frame relative to the proximal coordinate frame of a link. Zeigert and Datseris [26] points out the necessity of using a six-parameter kinematic model.

Assuming linear joint transducer characteristics we adopt a two parameter model for manipulator joints. Following [1], [11], and [22] we model a manipulator joint as,

$$\theta = \theta_0 + k\,\theta_i \ , \tag{2}$$

where θ_0 is called the joint transducer zero-point offset, and k is the joint transducer output constant. An ideal joint should have unity k implying that an increment θ_i of the joint encoder actually represents an equal increment of the joint angle.

A PUMA 560 manipulator, which we use in our experiment, is an open-loop six degree-of-freedom (DOF) kinematic chain composed of a base and an end-effector (which we design) connected by five intermediate links and six joints. Allowing six parameters per link and two parameters per joint, a total of 54 parameters (for seven links and six joints) are required to completely describe the manipulator[4]. The resulting transformation matrix T relating the endpoint to the global coordinate frame may then be expressed as a sequential multiplication of the 13 local homogeneous transformation matrices (one for each of the seven links and the six joints), as follows:

$$T = T_1^L\, T_1^J\, T_2^L\, T_2^J T_3^L \ldots\ldots \; T_6^L\, T_6^J\, T_7^L \ , \tag{3}$$

where T_i^L and T_i^J represent the link transformation matrices and the joint transformation matrices, respectively, of the i^{th} link or joint. The global position and attitude of the endpoint can be easily extracted from the matrix T above [9]. We use the position components to calculate the expected distance of the endpoint from the fixed end of the ball-bar.

4.0 THE OPTIMIZATION SCHEME

In practice, the computed distance of the robot's endpoint from the fixed end of the ball-bar will be different from its measured distance. This difference, known as the *residual*, is computed at each data collection point. The aggregate sum-of-squares of the residuals over n data collection points, ϕ, is calculated as

$$\phi = \sum_{i=1}^{n} (d_i^a - d_i)^2 \ , \tag{4}$$

where d_i^a and d_i are the measured distance and the computed distance, respectively, at the i^{th} data collection point. The quantity ϕ is used as an objective function, to be minimized by parameter estimation to ensure the best possible manipulator accuracy.

[4]Note that according to Equation 1, only 36 of its parameters may be identified. Our model, therefore contains 18 extra parameters, which are called *redundant parameters* [10].

The mathematical formulation and implementation of various parameter estimation schemes are detailed in [10] Here, a brief exposition of the implemented method is presented. The multivariable objective function ϕ depends on all 54 parameters, and may be expressed as the second-order expansion of a Taylor series as

$$\phi(p + \Delta p) \cong \phi(p) + g^T \Delta p + \frac{1}{2!} \Delta p^T H \Delta p \ , \tag{5}$$

where p is the 54-vector of parameter initial guesses and Δp is an small perturbation of p. Vector g is the gradient of the objective function with respect to p. Matrix H is the local Hessian matrix of the objective function ϕ. The gradient vector represents the direction of maximum rate of change of the objective function surface and ideally has zero magnitude at the minimum. The Hessian matrix is a 54^{th}-order quadratic matrix which contains information about the convexity of the objective function. Since our kinematic model contains redundant parameters (a total of 54 parameters as opposed to 36 independent parameters) the Hessian matrix is singular. In a non-redundant kinematic model, singularities may occur if the collected data fail to excite all the parameters.

The well known *steepest descent method* (also called *gradient search method*) selects a search direction precisely opposite to that of the gradient vector. The search technique continues iteratively until the nearest local minimum of the objective function is reached[5]. Alternatively, under the *Gauss-Newton* algorithm, an iteration step is taken according to

$$\Delta p = -H^{-1} g \ , \tag{6}$$

which minimizes the objective function in a small neighborhood of the current objective function value. The elements of the update vector Δp are added to the corresponding elements of the parameter vector to give an improved set of parameters. This process continues until one or more pre-defined convergence criteria are met [18].

Although the steepest descent method is guaranteed to reach a local minimum or a saddle point, the greatest disadvantage with this method is that it often requires an excessively large number of iterations for convergence. The Gauss-Newton method, on the other hand, fails whenever the Hessian matrix is singular. It was observed by Fletcher [8] that, even in the case of positive definite Hessians, convergence may not be assured.

A robust algorithm to solve singular systems was developed independently by Levenberg [16] and Marquardt [17] and is known as the Levenberg-Marquardt algorithm [19] [18]. This algorithm has been successfully used in the parametric synthesis of kinematic linkages by Tull and Lewis [23] and Chen and Chan [2]. In recent years, it has become popular with the robotics community, especially in robot parameter estimation applications [1, 12].

According to the Levenberg-Marquardt algorithm, (6) is modified as

$$\Delta p = -(H + \lambda I)^{-1} g \ , \tag{7}$$

where λ is an adjustable scalar and I is an identity matrix. A sufficiently large λ *is* always available which makes the inversion of $H + \lambda I$ possible. This method represents a useful

[5]We assume that our initial guess, based on the nominal design parameters of the robot, is sufficiently close to the current parameters and that the optimization process terminates at the global minimum.

combination of the steepest descent and the Gauss-Newton techniques.

The search procedure commences with a large λ and an initial guess for the model parameters. The behavior of the algorithm resembles the steepest descent method at this time. For every successful iteration, as reflected by a decrease in the objective function, the λ is adjusted to a lower value. As the search progresses, the successive steps increasingly resemble Gauss-Newton steps.

In order to avoid calculating the second order derivatives (in $\boldsymbol{H}$), the gradient and the Hessian matrix may be expressed as [10]

$$\boldsymbol{g} = -2\boldsymbol{J}^T\boldsymbol{f} \qquad \text{and} \qquad \boldsymbol{H} = 2\boldsymbol{J}^T\boldsymbol{J} \ , \qquad (8)$$

where $\boldsymbol{f}$ is a vector of residuals and $\boldsymbol{J}$ is a Jacobian matrix[6]. A typical element J_{ij} of the Jacobian matrix is expressed as

$$J_{ij} = \frac{\partial f_i}{\partial p_j}, \qquad (9)$$

where f_i is the i^{th} element of the residual vector $\boldsymbol{f}$ and p_j is the j^{th} element of the parameter vector $\boldsymbol{p}$. In practice, enough calibration data should be collected (more than the minimum number necessary to solve (7), see [22]) to populate the portion of the workspace where greatest accuracy is desired. The Jacobian matrix therefore contains many more rows (equal to the number of data points) than columns (equal to the number of parameters).

Equation 7 may now be rewritten as

$$\Delta\boldsymbol{p} = (\boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I})^{-1}(\boldsymbol{J}^T\boldsymbol{f}) \qquad (10)$$

Equation 10 is called the *normal equation* of this least-squares optimization. We use this equation in error minimization.

5.0 DATA COLLECTION PROCEDURE

Recall from Section 2.0 that our data collection procedure involves measuring the distance of the robot endpoint from suitable positions in its workspace. Since we can measure only one out of six pose information of the robot endpoint at a given posture (with the ball-bar), we have to devise a data collection method in order to access the complete pose information. This we do by comparing our data collection "kinematics" with that of a Stewart platform. In a Stewart platform, six linear actuators are connected between three points on a base and three points on a platform such that at any configuration the actuator lengths completely identify the pose of the platform relative to the base [7]. We use three magnetic chucks mounted on a fixture on the robot table and imagine the fixture as the Stewart platform base. Another fixture with three precision balls mounted to the robot endpoint assumes the role of the platform.

Ideally we should use simultaneously use six ball-bars in order to completely identify the robot endpoint pose *at every posture*. However we find it equally useful to "serialize" the procedure by commanding the robot to travel on a spherical shell. while only one ball-bar is connected between its end point and the table. We repeat this procedure six times while interchanging the connections between the three balls and three chucks in six appropriate combinations. In this way we perform a complete parameter identification with a single ball-bar at the cost of extra time.

We use the fixture shown in Fig. 1 to mount the three precision balls to the robot endpoint. The fixture must be designed such that the three balls are not collinear and are not mounted on the axis of the final robot joint. Other design concerns are low overall weight to minimize gravity loading, and small size to minimize possibilities of collisions. The fixture used to mount the three magnetic chucks to the workspace is shown at the bottom of Fig. 1. The three magnetic chucks must not be collinear, and the fixture should be mounted in a location such that the spherical shells traced out by the ball-bar lie where the highest robot accuracy is desired.

For data collection to be automated, the ball-bar must remain connected to the robot. Unfortunately, the limited reach of the ball-bar substantially restricts the robot endpoint's positional freedom. Furthermore, collisions between the ball-bar and the robot-mounted fixture must be avoided. The net effect of these restrictions is that the path planning cannot be done on-line using Unimation's VAL robot controller user interface.

Path planning must then be done off-line on an external computer. We use a custom program for path planning utilizing an inverse kinematics routine based on the uncalibrated robot model. We pre-define positions where we wish to collect data. These positions are located at the intersections of selected latitude and longitude lines on the hemisphere reachable by the ball-bar at zero deflection (see Fig. 2). Moving the robot between these data points without exceeding the ball-bar range requires the use of intermediate path points ("via"s). The off-line path planning must also avoid robot joint limits and singularities. The final output of this planning is a list of robot joint angle 6-vectors, each corresponding to a data collection pose or a "via".

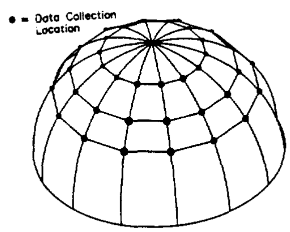

Figure 2. The figure shows a spherical shell on which the robot travels while measurement takes place on each indicated data collection point.

During on-line data collection, the external program transmits this list of joint vectors to the robot controller. The program commands the robot to sequentially move through the "via"s, stopping at the data collection points for two seconds (to allow any dynamic oscillations to subside), before reading the ball-bar length and the current robot joint angles (which may differ slightly from the commanded joint angles). This information is then automatically recorded for later use in an off-line parameter identification program (see Section 5.0).

As mentioned, a minimum of six trajectories on six spherical shells must be used to collect data. Since collection of excess data is always beneficial, sometime we use all nine possible

[6]This Jacobian matrix $\boldsymbol{J}$ should not be confused with the more common 6x6 Jacobian matrix that maps joint displacements to endpoint displacements.

ball-bar connections. Also, we use ball-bar extensions to collect additional sets of data on shells with larger radii.

6.0 PARAMETER IDENTIFICATION

Duelen and Schroer [3] mentions that all of the kinematic parameters of a manipulator may be identified by measuring only the position of a suitable point on the end-effector. Only requirement is that the measurement point should be away from the axis of the robot's last joint. In our experiment, however, we were able to identify only 33 parameters while measuring the distance of a single end-effector ball from all 3 fixed chucks. This situation is equivalent to measuring the three position coordinates of the ball which, by virtue of the fixture design, is not on the robot's sixth joint axis. The three orientation parameters of the last link elude identification in this case. A recent paper supports our results [27]. If the suitable endpoint happens to lie on the sixth joint axis we fail to identify, additionally, the error in the sixth joint offset.

In order to evaluate the possible physical changes in any of the link geometric features or joint encoders characteristics, the collected data must "excite" all 36 independent parameters of the PUMA 560 robot. The number of excited (or identified) parameters is given by the number of linearly independent columns in the Hessian matrix $\boldsymbol{H}$ or the Jacobian matrix $\boldsymbol{J}$. Counting the number of non-zero singular values of $\boldsymbol{H}$ or $\boldsymbol{J}$ therefore constitutes a simple test for calculating the number of identified parameters [11, 27].

We ran several trials for identifying the robot parameters by choosing different combinations of the nine ball-bar connections. All the combinations along with the corresponding number of identified parameters are shown in Table I.

TABLE I: DIFFERENT BALL-BAR CONNECTIONS AND THE NUMBER OF IDENTIFIED PARAMETERS

Number of magnetic chucks on robot table \ Number of spherical balls on robot endpoint	1	2	3
1	30	32	33
2	32	34	35
3	33	35	36

The above results indicate that by using all nine combinations of ball-bar connections, we can identify the complete set of 36 parameters. Any six of these nine combinations, which is analogous to a 6-DOF Stewart platform, is sufficient for identifying all the independent parameters.

7.0 RESULTS AND DISCUSSION

In order to check the success of the parameter estimation process, we use the optimal parameter set computed from one set of nine hemispherical shells and use it to compute the accuracy of the robot in a different spherical shell. We first use data collected from a set of nine hemispherical shells, using the ball-bar without an extension bar, to compute a set of optimal parameters. Using these parameters with the collected data, we compute the RMS residual value to be 0.084 mm.

Fig. 3, which shows a distribution of the final residuals on the longitudes of the data collection sphere (see Fig. 2), demonstrates that the residuals do not vary significantly with spatial position and thus, they reflect random errors such as repeatability and gear backlash rather than the errors in the geometric parameters.

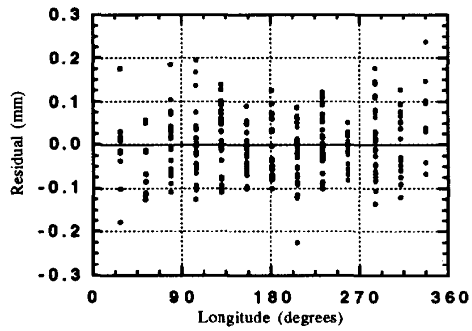

Figure 3. Plot of the distribution of the final residuals vs. longitude of the hemispherical calibration shell.

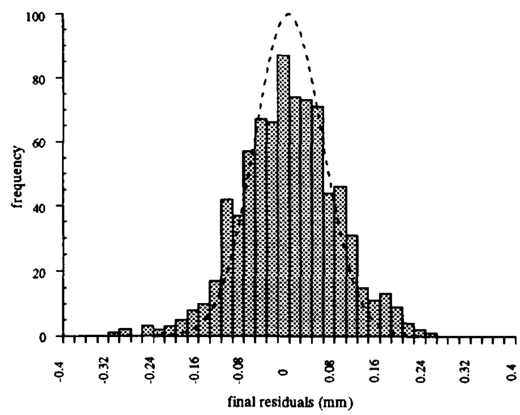

Figure 4. Histogram of the final residuals from 800 calibration points.

In Fig. 4 we plot a histogram of the final residuals. The Gaussian distribution curve which fits the histogram has a mean of 0.0 mm and a standard deviation of 0.084 mm has been superposed on the histogram. The final residuals are seen to be normally distributed about zero error.

The optimized parameters from one set of nine spheres are next used to compute the residuals for a different set of spheres using the ball-bar with a 15 cm extension bar. The initial RMS residual value using these parameters is 0.110 mm. This result implies that the optimized kinematic model correctly reflects the robot geometry, and is not simply a best fit of the collected data.

Finally, it is important to distinguish between the RMS error of the ball-bar and the RMS error of the robot endpoint position. We attempt to compute the positional error of the endpoint triangle for a given RMS error of the ball-bar. The Stewart platform analogy is again useful here. Imagine again that the robot endpoint triangle (with three balls) and the base triangle (with three chucks) are interconnected through six ball-bars in a Stewart platform geometry. The question we then ask is: for 0.084 mm RMS errors in the ball-bars, what is the RMS error in the positioning of the triangle?

The so-called Jacobian matrix of a Stewart platform transforms small changes in its leg lengths to small changes in the position of the platform [7]. For a nominal configuration of our imaginary Stewart platform (the robot endpoint plate situated directly over the base triangle with all the ball-bars of equal length) we compute how errors in each of its legs propagate to the top platform. Using an RMS error of 0.084 mm we find the positional error to be roughly four times the ball-bar error.

8.0 CONCLUSIONS

We have shown that incomplete pose information collected in accordance with a special data-collection scheme can be successfully used in identifying all the independent kinematic parameters of a robot. We used a radial-distance transducer, called the telescopic ball-bar system, to simply measure the distance of the robot endpoint from a fixed point in the workspace. Special fixtures were mounted on the robot endpoint and on the robot table in order to indirectly access the complete pose information of the robot. We identified all the independent parameters of a six dof robot. Our procedure represents an easy and economic way of calibrating a robot.

ACKNOWLEDGMENTS

We would like to thank James Bosnik of Drexel University for useful discussions, especially in regard to the idea of the robot endpoint fixture. Support for this research was provided by Northwestern University, Northwestern Memorial Hospital and the NSF PYI Grant DMC-8857854. Kornel Ehmann of Northwestern University drew our attention to the use of ball-bar for calibration purpose. Kam Lau of Automated Precision Inc. was instrumental in making the ball-bar device available to us at a subsidized price.

REFERENCES

1. Bosnik, J. R. *Static and Vibrational Kinematic Parameter Estimation for Calibration of Robotic Manipulators.* Ph. D. Thesis, Pennsylvania State University, University Park, PA (1986)

2. Chen, F. Y. and V.-L. Chan. *Dimensional Synthesis of Mechanisms for Function Generation Using Marquardt's Compromise.* ASME Journal of Engineering for Industry **96**(1):1312-137 (1974)

3. Duelen, G. and K. Schroer. *Robot Calibration—Method and Results.* Robotics and Computer-Integrated Manufacturing **Vol. 8**(No. 4):223-231 (1991)

4. Everett, L. J., M. Driels and B. w. Mooring. *Kinematic Modelling for Robot Calibration.* Proceedings of IEEE International Conference of Robotics and Automation. (183-189) IEEE Press (1987)

5. Everett, L. J. and T.-W. Hsu. *The Theory of Kinematic Parameter Identification for Industrial Robots.* ASME Journal of Dynamic Systems, Measurement, and Control **Vol. 110, March**:96-100 (1988)

6. Everett, L. J. and A. H. Suryohadiprojo. *A Study of Kinematic Models for Forward Calibration of Manipulators.* Proceedings of IEEE International Conference of Robotics and Automation. Philadelphia, PA (798-800) IEEE Press (1988)

7. Fichter, E. F. *A Stewart Platform Based Manipulator: General Theory and Practical Construction.* Int'l J. Robotics Research 5(2):157-182 (1986)

8. Fletcher, R. *Practical Methods of Optimization.* John Wiley and Sons, Inc.,. New York (1980)

9. Fu, K. S., R. C. Gonzalez and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence.* McGraw-Hill. New York, NY (1987)

10. Goswami, A. *Analysis of the Relationship between the Physical and the Mathematical Kinematic Parameters in Robotic Manipulator Parameter Estimation Algorithms.* M.S. Thesis, Drexel University, Philadelphia, PA (1988)

11. Goswami, A. and J. R. Bosnik. *On a Relationship between the Physical Features of Robotic Manipulators and the Kinematic Parameters Produced by Numerical Calibration.* ASME Journal of Mechanical Design (Accepted) (1992)

12. Hollerbach, J. M. *A Survey of Kinematic Calibration.* The Robotics Review 1. al. ed. MIT Press. Cambridge, MA (1989)

13. Judd, R. P. and A. B. Knasinski. *A Technique to Calibrate Industrial Robots with Experimental Verification.* Proceedings of IEEE International Conference of Robotics and Automation. (351-357) IEEE Press (1987)

14. Kinzle, T. C., S. D. Stulberg, M. Peshkin, A. Quaid and C.-H. Wu. *An Integrated CAD-Robotics System for Total Knee Replacement Surgery.* IEEE Conference on Systems, Man, and Cybernetics. Chicago IEEE Press (1992)

15. Lau, K., N. Dagalakis and D. Myers. *Testing.* International Encyclopedia of Robotics: Applications and Automation. Dorf ed. John Wiley and Sons. New York (1984)

16. Levenberg, K. *A Method for the Solution of Certain Non-Linear Problems in Least-Squares.* Quarterly of Applied Mathematics-Notes **2**(2):164-168 (1944)

17. Marquardt, D. W. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters.* Journal of the Society for Industrial and Applied Mathematics **11**(2):431-441 (1963)

18. Nash, J. C. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimization.* John Wiley and Sons, Inc. New York, NY (1979)

19. Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press. Cambridge, U.K. (1985)

20. Roth, Z., B. W. Mooring and B. Ravani. *An Overview of Robot Calibration.* IEEE Transactions of Robotics and Automation **RA-3**(5):377-385 (1987)

21. Sheth, P. N. and J. J. Uicker. *A Generalized Symbolic Notation for Mechanisms.* ASME Journal of Engineering for Industry **93**(1):102-112 (1971)

22. Sommer, H. J. and N. R. Miller. *A Technique for the Calibration of Instrumented Spatial Linkages Used for Biomechanical Kinematic Measurements.* Journal of Biomechanics **14**(2):91-98 (1981)

23. Tull, H. G. and D. W. Lewis. *Three Dimensional Kinematic Synthesis.* ASME Journal of Engineering for Industry **90**(3):481-484 (1968)

24. Vira, K. and K. Lau. *Design and Testing of an Extensible Ball Bar for Measuring the Positioning Accuracy and Repeatibility of Industrial Robots.* NAMRAC Conference. (583-590) (1986)

25. Wu, C. H., J. Papaiannou, T. C. Kienzle and S. D. Stulberg. *A CAD-Based Human Interface for Preoperative Planning of Total Knee Surgery.* IEEE Conference on Systems, Man, and Cybernetics. Chicago IEEE Press (1992)

26. Zeigert, J. and P. Datseris. *Basic Considerations for Robot Calibration.* Proceedings of IEEE International Conference of Robotics and Automation. Philadelphia, PA (932-938) IEEE Press (1988)

27. Zhuang, H., Z. S. Roth and F. Hamano. *Observability Issues in Kinematic Identification of Manipulators.* ASME Journal of Dynamic Systems, Measurement, and Control **Vol.114**(June):319-322 (1992)