A PROTOTYPE ARM SIGNATURE IDENTIFICATION SYSTEM

Henry W. Stone and Arthur C. Sanderson

Department of Electrical and Computer Engineering and Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213

The S-Model identification algorithm described in [6, 7] is a technique which can be used to accurately identify the actual kinematic parameters of serial link robotic manipulators. The actual kinematic parameters of a manipulator differ from the design parameters due to the presence of random manufacturing errors. The set of identified kinematic parameters is called the *arm signature*. Accurate arm signatures are needed to control and improve the end-effector positioning accuracy of robotic manipulators for a variety of important tasks. This paper describes the hardware and software implementation of a prototype arm signature identification system. This system uses an external ultrasonic range sensor to measure the Cartesian position of target points placed on the links of the robot. Algorithms to compensate the primary range measurements for spatial variations in air temperature and humidity are also incorporated. The relative Cartesian positioning *accuracy* of the sensor system is ± .02 cm. The general characteristics of our sensor design and the overall system design which exploits averaging over many sensor readings offer numerous advantages for arm signature identification. The prototype system has been applied in [6] to improve the kinematic performance of seven Puma 560 robots. For these robots relative positioning accuracy was improved by a factor of 10 on straight-line positioning tasks. Analysis and simulation of systematic errors confirms that the resolution of our sensor system should provide kinematic performance close to the limitations of the joint encoders. Our experimental studies show that sensor bias ultimately limits kinematic performance using this arm signature system. Experience with this prototype system has demonstrated that the S-Model identification algorithm is a practical and viable method for improving the kinematic performance of robotic manipulators.

1 Introduction

The S-Model identification algorithm described in [6, 7] is a technique which can be used to accurately identify the actual kinematic parameters of serial link robotic manipulators. The actual kinematic parameters of a manipulator differ from the design parameters due to the presence of random manufacturing errors. The set of identified kinematic parameters is called the *arm signature*. Accurate arm signatures are needed to control and improve the end-effector positioning accuracy of robotic manipulators for a variety of important tasks.

The S-Model is a mathematical model which can be used to describe the exact kinematic structure of any robotic manipulator with rigid links. In contrast to the Denavit-Hartenberg model, the S-Model is directly applicable to identification. In particular, the flexibility with which the link coordinate frames can assigned and fixed to each of the robot's links provides a mechanism for decoupling the entire identification problem into a set of simple identification problems. Linear regression techniques are applied to solve these simpler problems and the solutions are combined to estimate the $6 \cdot n$ S-Model parameters. For control, the S-Model parameters are easily mapped into an equivalent set of Denavit-Hartenberg parameters. The S-Model identification algorithm has been applied to identify the signatures and improve the performance of seven Puma 560 robots. The experimental results presented in [7] demonstrate the potential advantages and practicality of the S-Model identification algorithm and control approach.

This paper describes the hardware and software implementation of a prototype arm signature identification system. The primary hardware components of this system are a three-dimensional Cartesian position sensor system, sensor system control and interface hardware, a robot manipulator, a robot controller, and a host computer. The S-Model identification algorithm uses measurements of the Cartesian position of targets placed on the robot's links to estimate the kinematic parameters. In our system, we use an ultrasonic source (sparker) for each target and an array of four ultrasonic range detectors to estimate target positions. Target range measurements are collected by the Puma 560 Controller which is interfaced to the ultrasonic sensor system via an RS-232 serial link. The measurements are then uploaded to a Vax 11/780 for processing and analysis through a hardline interface. While our experiments have focused upon the performance of Puma 560 robots, our prototype system can be applied to identify the signatures of an y moderately sized robot.

The software which controls the Puma 560 during the data collection process and activates the various target sparkers is implemented in the Val II programming language. It is initially downloaded to the Puma 560 Controller from the Vax 11/780 which maintains current versions of all the systems software as well as performance data. The primary task of this program is to sequence the robot's joint configurations insuring that the proper target trajectories are generated. After being uploaded to the Vax, the range measurements are feed through a preprocessing algorithm, written in the C programming language, which estimates the true Cartesian positions of the targets. The Cartesian target positions are then applied to identify the robot's kinematic parameters. The signature identification algorithms are also implemented in the C programming language and run on the Vax. Finally, we utilize the identified arm signature to control the robot end-effector in a variety of standardized tasks. For convenience, the numerical algorithms which have been developed to invert the complex arm signature models are implemented in the C programming language. Signature control is thus performed offline.

This paper is organized as follows. In Section 2 we re-introduce the S-Model and review the S-Model identification algorithm. Then, in Section 3, we describe the organization of our prototype arm signature identification system. The ultrasonic sensor system is described separately in Section 4 along with the methods used to compensate the raw range measurements. Section 5 highlights the data collection process and parameter identification algorithms. In Section 6, simulation results are presented which provide insight into the statistical performance of the S-Model identification algorithm. Concluding remarks appear in Section 7

2 Kinematic Parameter Identification

Analytically, the S-Model identification algorithm [6, 7] is based upon the properties and structure of the S-Model. The S-Model is a completely general method for describing and characterizing kinematics of robotic manipulators. In the S-Model, the matrix

$$\mathbf{S}_n = \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \ldots \cdot \mathbf{B}_n \tag{1}$$

defines the position and orientation of a coordinate frame fixed relative to the last (n^{th}) link of a manipulator with respect to a coordinate frame fixed relative to the base link. The *general transformation matrices* $\mathbf{B}_i$ in (1) are (4x4) homogeneous transformation matrices. The $\mathbf{B}_i$ and $\mathbf{S}_n$ matrices in (1) are analogous to the $\mathbf{A}_i$ and $\mathbf{T}_n$ matrices of the Denavit-Hartenberg model [2]. The symbolic name $\mathcal{S}_i$ signifies the i^{th} link coordinate frame defined by the S-Model.[1] The transformation matrix $\mathbf{B}_i$ describes the relative transformation between the $\mathcal{S}_{i-1}$ and $\mathcal{S}_i$ coordinate frames (measured with respect to the $\mathcal{S}_{i-1}$ coordinate frame). In the S-Model, *six* parameters β_i, $\bar{\mathbf{d}}_i$, $\bar{\mathbf{a}}_i$, $\bar{\alpha}_i$, γ_i, and $\mathbf{b}_i$ define the transformation matrix

$$\mathbf{B}_i = Rot(z,\beta_i)Trans(0,0,\bar{\mathbf{d}}_i)Trans(\bar{\mathbf{a}}_i,0,0) \tag{2}$$

$$Rot(x,\bar{\alpha}_i)Rot(z,\gamma_i)Trans(0,0,\mathbf{b}_i) \quad .$$

The physical definitions of the S-Model parameters are illustrated in [7]. To specify the S-Model for an n degree-of-freedom manipulator requires $6 \cdot n$ parameters in contrast to the $4 \cdot n$ parameters for Denavit-Hartenberg model. Because each link transformation matrix is specified by six parameters rather than four, the S-Model convention which defines the allowable positions and orientations of the link coordinate frames is less restrictive than the Denavit-Hartenberg convention.

There are two fundamental distinctions between the Denavit-Hartenberg link coordinate frame $\mathcal{T}_i$ and the S-Model link coordinate frame $\mathcal{S}_i$. First, in contrast to the origin of $\mathcal{T}_i$, the location of the origin of $\mathcal{S}_i$ on the joint $i+1$ axis is arbitrary. Second, the direction of the X axis of $\mathcal{S}_i$ must only be orthogonal to the Z axis. The arbitrary location of the origin of $\mathcal{T}_i$ along the joint axis and the arbitrary orientation of the X axis of $\mathcal{S}_i$ provide an infinite number of link coordinate frames $\mathcal{S}_0$ through $\mathcal{S}_n$ which satisfy the S-Model convention. The S-Model identification algorithm exploits these properties are in order to decouple and simplify the kinematic parameter identification problem. Furthermore, the analytic structure of the S-Model provides a simple mechanism for computing the Denavit-Hartenberg parameters.

The objective of *S-Model identification* is to estimate the S-Model kinematic parameters from a set of $2n_r + n_p$ mechanical features inherent to the manipulator, where n_r is the number of revolute joints and n_p is the number of prismatic joints. (The number of degrees-of-freedom $n = n_r + n_p$). The two features of a revolute joint are a called the *center-of-rotation* and the *plane-of-rotation*, and the feature of prismatic joint is called the *line-of-translation*. These features are derived from basic geometric considerations of the joints. For expository purposes, the following discussion is limited to robots with revolute joints. The locus of a point rotating about an axis is a circle lying in a plane, called the *plane-of-rotation* and the normal to this plane is a vector which is parallel to the axis of rotation. The center of the circle is a point, called the *center-of-rotation* which lies on the axis of rotation. When joint i-1 of a manipulator is rotated, any point which is fixed relative to the i^{th} link defines a plane-of-rotation and a center-of-rotation, under the assumption that the positions of joints 1 through i-2 remain fixed. This plane-of-rotation and center-of-rotation are by definition associated with the $(i$-$1)^{th}$ joint and the i^{th} link.

The S-Model identification algorithm consists of four steps in which the overall identification problem is separated into a set of independent, less complex identification problems. The first step, *feature identification*, involves the explicit identification of the kinematic features from measurements of the Cartesian position of targets placed on the robots links. The robot is placed in an arbitrary starting configuration, called the *signature configuration*, and is then programmed to move through a systematic sequence of joint configurations. At each configuration the position of a target is measured. During this sequence of motions, each joint beginning with the last is incrementally indexed through its complete range of motion. The target point which is attached to the link following the joint thus generates a circle in space. Figure 1 illustrates the generation of a plane-of-rotation and a center-of-rotation for Joint 6 of a Puma 560.

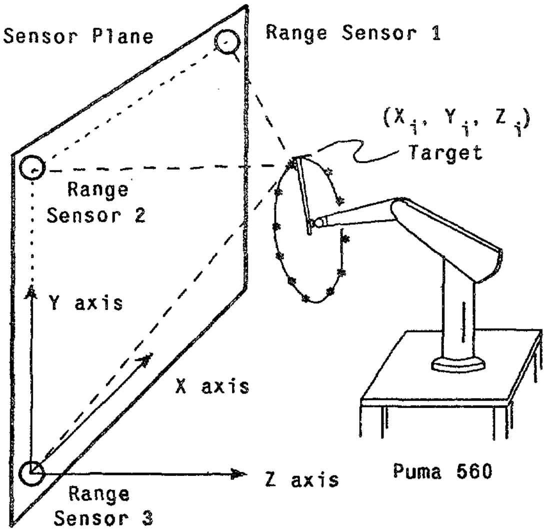

Figure 1: Generating a Plane-of-Rotation and Center-of-Rotatio for Joint 6 of a Puma 560 robot.

Linear least-squares algorithms are then applied to estimate the coefficients of the equations which define the set of generated planes and circles.

In the second step, *link coordinate frame specification*, the identified planes-of-rotation and centers-of-rotation are used to establish the positions and orientations of a set of S-Model link coordinate frames. These positions and orientations are defined with respect to the sensor system coordinate frame. This step amounts to computing the elements of the matrices

$$\bar{\mathbf{S}}_i = \mathbf{P}\mathbf{S}_i \qquad \text{for } i = 0, 1, \ldots, n \tag{3}$$

where $\mathbf{P}$ is a constant transformation representing the spatial relationship between the sensor system coordinate frame and a coordinate frame fixed relative to the base of the robot. Using Paul's terminology [4], the $\mathbf{a}$ vectors of the matrices $\bar{\mathbf{S}}_i$ are set equal to the unit normal vectors to the identified planes-of-rotation; the $\mathbf{p}$ vectors are set equal to the vectors which define the positions of the identified centers-of-rotation; and the $\mathbf{n}$ and $\mathbf{o}$ vectors are chosen arbitrarily[1].

In the third step, *S-Model parameter computation*, the elements of the link transformation matrices $\mathbf{B}_i$ are evaluated and the $6 \cdot n$ S-Model parameters are solved using the formulas derived in [6].

Finally, in the fourth step, the Denavit-Hartenberg parameters including a set of n encoder offsets are determined. The identified signature (Appendix I) can then be applied to control the manipulator.

3 System Organization

In order to demonstrate the feasibility and evaluate the performance of the identification and control algorithms, we have developed a complete prototype arm signature identification system. Figure 2 illustrates the configuration of the hardware

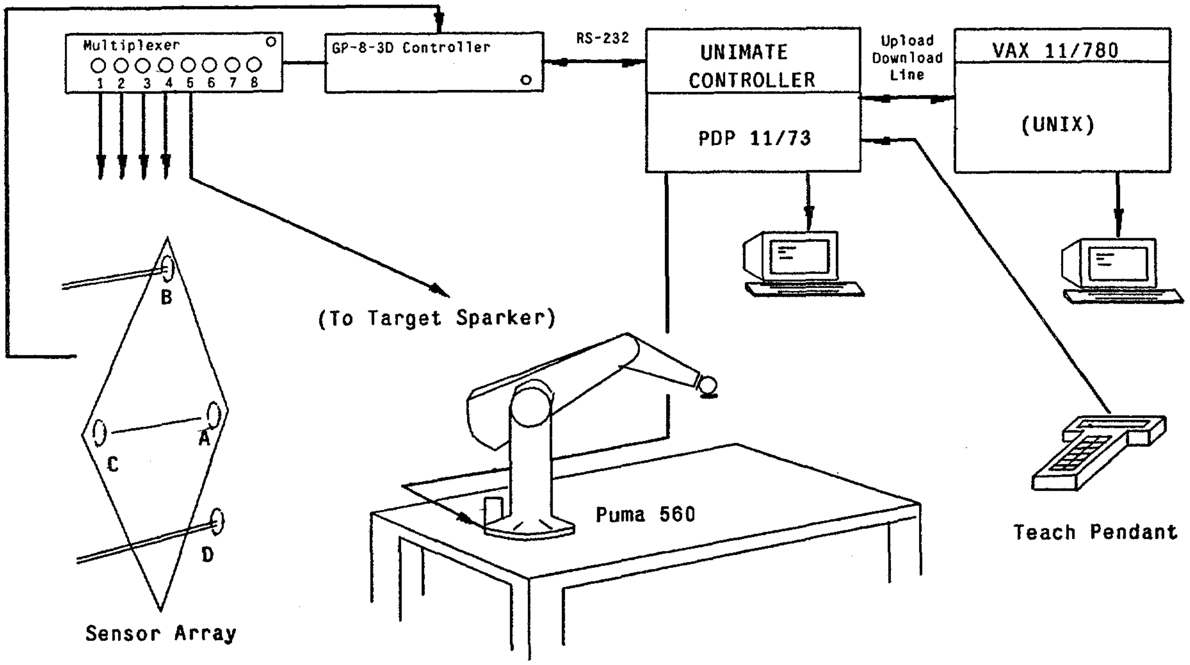

Figure 2: Hardware Configuration

components in this implementation which include a three-dimensional Cartesian position sensor system, sensor system control and interface hardware, a robot manipulator, a robot controller, and a host computer.

The sensor system measures the three-dimensional positions of targets attached to the robot arm. In our system, we use an ultrasonic source (sparker) for each target and an array of four ultrasonic range detectors to estimate target positions. Each detector measures time-of-flight independently from the target and computes an estimated range. Precautions are taken to insure that effects of environmental disturbances such as temperature variations and air currents are minimized. In addition, since the speed of sound is extremely sensitive to temperature, we have developed a special method for estimating the spatial variations in the speed of sound and apply these estimates to compensate the sensor's range measurements. Knowing the geometry of the detector configuration the three-dimensional coordinates of the point may be determined.

The sensor system is interfaced to the robot controller using the RS-232 serial port available on the GP-8-3D Sonic Digitizer controller [5]. In response to commands initiated from the robot controller, the Sonic Digitizer's controller coordinates the low level communications between the sensor's receiving and transmitting devices. It then transmits initial estimates for the target ranges to the robot controller.

Our laboratory experiments have focused upon the arm signature identification and kinematic control of Puma 560 robots. The Puma 560 [9] is six degree of freedom revolute joint manipulator and provides a good testbed. Our prototype system, however, is not limited to the analysis of Puma 560's and can be applied to identify the signature of any robot whose size is commensurate with the active volume of the sensor system.

The software which controls the Puma 560 during the collection of the raw target ranges is implemented on the Puma's Val II Controller and is written in the Val II programming language [10]. This program is initially downloaded from the VAX 11/780 which maintains current versions of all the system software and provides permanent storage for all of our experimental data. The software which implements the range compensation and target triangulation algorithms is currently written in the C programming language to run under the Unix[2] operating system on the Digital Equipment Corporation VAX 11/780 [3]. The range measurements are thus collected by the VAL II controller and then uploaded to the VAX 11/780. The software which implements the feature parameter estimation and computes the signature parameters is also written in the C programming language and runs on the VAX 11/780. The implementation of the range compensation, target triangulation, and signature identification algorithms on the VAX 11/780 was a matter of convenience. In practice, such software could reside in the robot controller. Once identified, the arm signature is stored for future use to control the manipulator.

4 Sensor System

4.1 Hardware Description

The GP-8-3D Sonic Digitizer [5] is an ultrasonic range sensing system which computes range measurements based upon the time of flight of an ultrasonic wave emitted from a source (i.e., sparker) to a set of receivers (i.e., microphones). The GP-8-3D Sonic Digitizer includes a controller, a multiplexer, four ultrasonic piezo-electric transducers (microphones), and eight emitters (sparkers). The controller, equipped with an RS-232 serial port, has been interfaced to the PDP 11/73 in the Puma 560 Controller to allow for remotely-controlled digitizing and data acquisition. Using the multiplexer, the range measurements to any one of eight different sparkers can be obtained.

Our sensor system design utilizes five sparkers and four microphones. The sparkers are numbered 1 through 5 and the microphones are labeled A through D. Sparker 5 represents the target sparker which is attached to each of the links of the manipulator during the identification process. The remaining sparkers and the microphones are mounted at fixed locations relative to one another on the *sensor array*. The locations of Microphones A, B, and C implicitly define a Cartesian coordinate

[1] To the extent that they are orthogonal each other and to the vector **a**.

[2] Unix is a trademark of AT&T Bell Laboratories, Murray Hill, NJ.

frame to which the Cartesian coordinates of the target sparker, obtained through triangulation, are referenced. In [6], this reference frame is called the *Sensor Coordinate Frame*. Its origin coincides with the acoustic zero point of Microphone C, its Y axis is directed along the line joining the acoustic zero points of Microphones C and B, and the Z axis is perpendicular to and out of the plane defined by the acoustic zero points of Microphones A, B, and C. The X axis is defined accordingly.

The GP-8-3D can measure slant ranges between 30 cm and 250 cm with a resolution of .01 cm. This, in combination with the aperture of the microphones and their relative placement, limits the sensor's active volume to a cube approximately 143.0 cm on a side. Although it has a resolution of .01 cm, the accuracy of the computed slant ranges varies significantly, depending upon the operating conditions (i.e., the environment). The speed at which sound travels in air is primarily a function of temperature, humidity, altitude, frequency, barometric pressure, and air flow. The speed of sound is particularly sensitive to variations in temperature (x cm/sec/ F at 70.0). In general, differences between the actual and nominal operating conditions produce a difference between the actual and nominal speed of sound and hence, reduce the accuracy of the computed slant ranges. The use of multiple sparkers and the sensor array design depicted in Figure 2 enable us to compensate the slant range measurements taken by the GP-8-3D for variations in the speed of sound. These procedures greatly increase the Cartesian positioning accuracy of the overall sensor system to ± .02 cm and are essential for arm signature identification.

4.2 Sensor Calibration and Compensation

The method we have developed for correcting the slant ranges obtained from the GP-8-3D (i.e., the raw slant ranges) compensates for both overall variations in the speed of sound and spatial variations in the speed of sound. Applying this method, we have achieved relative Cartesian positioning accuracies on the order of the resolution of the GP-8-3D. The resulting slant range compensation algorithms are based upon a simplified model of the operating environment. According to these model,

- The effect of air currents upon the speed of sound is negligible (A-1).
- All variables upon which the speed of sound depends, such as temperature and humidity, vary slowly relative to the time required to gather an *individual* target measurement (e.g., time constants on the order of minutes) (A-2).
- The temperature of the air within the active region of the sensor varies linearly with vertical height and is constant in the two horizontal directions (A-3).
- The temperature differential between the air at the top and bottom of the sensor's active region is less than 2 F (A-4).
- The average temperature of the air within the active volume of the sensor is between 65 F and 75 F (A-5).

In practice, robot manufacturers who want to identify the kinematic parameters of their manipulators as an integral part of the manufacturing process may enclose the sensor system in a moderately-controlled environmental chamber such as those currently used for robot testing and qualification. Use of the ultrasonic sensor system for accurate measurements requires attention to these issues.

We have determined experimentally that, in an environment which approximately satisfies the assumptions (A-1) - (A-5), the speed of sound varies linearly with vertical height. It is believed that this variation is caused primarily by the convection of warmer air although other factors may also have an effect. In our laboratory, temperature variations on the order of 1.0 - 1.5 deg occur across a height differential of 8 ft. To compensate for linear variations, two calibration rods are incorporated into the sensor array design. A calibration rod is simply an apparatus for securing two sparkers at a known and fixed distance apart. The upper calibration rod contains Sparkers 1 and 2, and the lower calibration rod contains Sparkers 3 and 4.

The calibration rods are positioned at strategic locations and provide a means for accurately determining the speed of sound at a particular instance in time at two different heights. Knowning the heights of the calibration rods with respect to the microphones we can identify the parameters of the linear gradient model. The speed of sound at each of the microphones is found through interpolation as is the speed of sound at the target. The latter, however, requires an additional step since an initial estimate of the target's height is required. In turn, this estimate depends upon the Cartesian position of the target. An initial estimate of the Cartesian position of the target is thus obtained by triangulating with the uncompensated range measurements.

Since the speed of sound varies linearly in only one direction, the variation in the speed of sound in any direction is also linear. The estimated speeds of sound at each microphone and at the target, along with the original time-of-flight measurements are used to compute the final estimates of the target ranges. Finally, the triangulation algorithms are again applied to determine the target's Cartesian positions. This method of compensating the GP-8-3D's slant range measurements requires that all five sparkers be sparked at least once to determine the target's location. In practice, 10-20 sets of slant range measurements are obtained and statistically averaged to reduce the effects of random measurement noise inherent to the GP-8-3D.

Each identified arm signature is based upon $\sum_{i=1}^{6} 7 \cdot N_{avg} \cdot N_i$ raw slant range measurements, where N_i denotes the number of distinct positions of joint *i* at which the position of target *i* is measured and N_{avg} denotes the number of slant measurements averaged at each target position. We have identified signatures using as few as five measurements per circle and as many as 100. When $N_i = 100$ for $i = 1, \ldots, 6$ and $N_{avg} = 10$, 42,000 slant range measurements are required. These measurements can be obtained in approximately 1 1/2 hours using the hardware in Figure 2. This includes the time required to manually switch the target sparker from one target point fixture to the next.

5 Model Identification

5.1 Generating Features

The S-Model Identification algorithm leads to an efficient and well-defined method for collecting measurements. We sequentially attach a sparker to each link of the Puma robot to implement the target points. The sparker attached to link *i* implements Target *i*. The fixtures used to attach the sparker to a link are simple both in design and construction. The two requirements for a fixture are that it be rigid and that the distance between the sparker and the Joint *i* axis be approximately equal to R_i, where R_i depends upon the active volume of the sensor system. We call R_i the nominal radius of target point *i*. In general, increasing the nominal radii increases the accuracy of the identified planes-of-rotation.

We begin the measurement process by positioning the manipulator in the desired signature configuration using a teach pendant. In practice, the selection of a signature configuration may depend upon the characteristics of the sensor system. In our case, we must insure that the lines of sight between the target sparker and the three Microphones A, B, and C remain unobstructed. Requirements such as this may also affect the design of the mounting fixtures and the selection of the target point's positions with respect to the links. The angular positions of the joints which define the signature configuration, denoted by the vector $\vec{q}^{\,s} = [q_1^s, q_2^s, \ldots, q_6^s]^T$, are measured and recorded.

The position of each joint of the manipulator is independently and incrementally varied such that the six target points generate circular trajectories in space. At each fixed configuration of the manipulator, 10-20 sets of seven slant ranges are measured and recorded to determine the target's position. The measured target positions represent sample points on the generated circular trajectories. Six such sets of points are obtained, each set corresponding to the trajectory of a distinct target point. The measured target positions will be used to identify 12 kinematic features; six planes-of-rotation and six centers-of-rotation, which are then applied to construct the set of S-Model link coordinate frame transformation matrices $\bar{\mathbf{S}}_i$ for $i=1,\ldots,6$. To identify the manipulator's kinematic parameters, the matrices $\bar{\mathbf{S}}_i$ must represent the position and orientation of a set of S-Model link coordinate frames when the manipulator is in the specified signature configuration. Consequently, while taking measurements of the *i* th target's position, joints *1 - i-1* must remain in their respective signature configuration positions (i.e., $q_1 = q_1^s,\ q_2 = q_2^s,\ \ldots\ ,\ q_{i-1} = q_{i-1}^s$). *The positions of joints i+ 1 - 6* have no effect upon the position of the *i* th target point since it is assumed to be rigidly attached to the *i* th link. Joints *i+ 1 - 6* are positioned so as not to limit the motion of joint *i* or obstruct the lines of sight between the target and Microphones A, B, and C.

We have established a systematic method for indexing the configuration of the manipulator to insure that the previous conditions are satisfied. Figure 3 presents a flow chart which summarizes the target measurement procedure in which measurements of the position of Target Point *6* are first obtained, followed successively by those of Target Points *5*, *4*, *3*, 2, and *1*.

5.2 Identifying Features

Identifying a plane-of-rotation and a center-of-rotation is conceptually straightforward. When joint *i* is rotated, Target *i* traces a circle in space. The coefficients of the equation of the plane in which the circle lies can be estimated from a curve fit of *m* measured Cartesian positions of the target along the circle corresponding to *m* different positions of joint *i*. The three-dimensional coordinates of the center-of-rotation can be extracted from an estimate of the parameters of the equation for the traced circle.

In [6], two methods for fitting the measured target positions to a plane are presented. In both the objective is to minimize the sum of the perpendicular distances between the measurements and the estimated plane. The solution to this minimization problem yields the coefficients which define an estimated plane-of-rotation. The first method [1], is based on an error measure divided by its average gradient and uses an eigenvalue solution. The second method approximates the solution of the desired minimization problem by the repeated application of a linear least-squares regression. The linear least-squares solution is easier to implement in software while the eigenvalue solution is analytically exact. We present a brief overview of the average gradient constraint algorithm.

The normal form for the equation of plane is

$$\mathrm{A}x + \mathrm{B}y + \mathrm{C}z + \mathrm{D} = 0\ , \tag{4}$$

where the coefficients A, B, C, and D are the parameters to be identified. The measure of error between the measured target position and the plane is the function

$$\xi(x,y,z) = \mathrm{A}x + \mathrm{B}y + \mathrm{C}z + \mathrm{D} = \varphi^T\Theta, \tag{5}$$

where $\varphi = [x\ y\ z\ 1]^T$ and $\Theta = [\mathrm{A\ B\ C\ D}]^T$. Using the *m* measured target positions, the aggregate error is

$$\Xi \equiv \sum_{j=1}^{m} \xi_j^2 = \sum_{j=1}^{m} (\Theta^T \varphi_j \varphi_j^T \Theta) = \Theta^T \Phi \Theta\ , \tag{6}$$

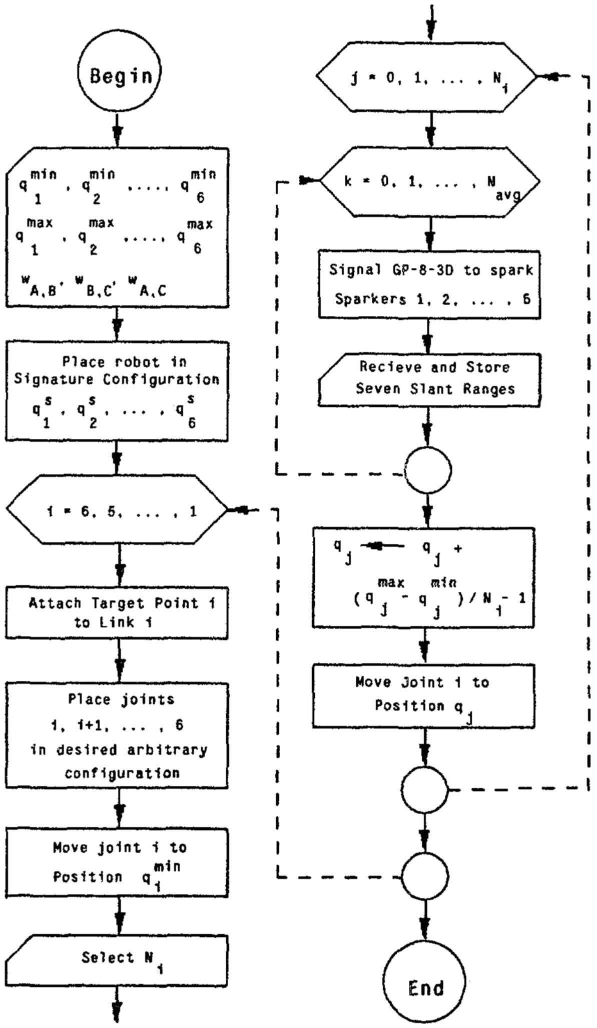
Begin
$q_1^{min}, q_2^{min}, \ldots, q_6^{min}$
$q_1^{max}, q_2^{max}, \ldots, q_6^{max}$
$^wA,B, {}^wB,C, {}^wA,C$
Place robot in Signature Configuration
$q_1^s, q_2^s, \ldots, q_6^s$
i = 6, 5, ... , 1
Attach Target Point i to Link i
Place joints i, i+1, ... , 6 in desired arbitrary configuration
Move joint i to Position q_i^{min}
Select N_i
j = 0, 1, ... , N_i
k = 0, 1, ... , N_{avg}
Signal GP-8-3D to spark Sparkers 1, 2, ... , 6
Recieve and Store Seven Slant Ranges
$q_j \leftarrow q_j + (q_j^{max} - q_j^{min}) / N_i - 1$
Move Joint i to Position q_j
End

Figure 3: Flow Chart of the Algorithm for Collecting Target Point Measurements

where $\Phi = \sum_{j=1}^{m} (\varphi_j \varphi_j^T)$. Minimizing the sum of the squares of the perpendicular distances between the measured target points and the plane is equivalent to minimizing Ξ under the constraint that the average magnitude of the gradient of the error function at the measured target points, $\vec{\mathbf{p}}_{i,j}$, for $j=1,\ldots,m$, is unity. The squared magnitude of the gradient of the error is $\Theta^T \Psi \Theta$ where Ψ is the (4x4) diagonal matrix Diag$[1\ 1\ 1\ 0]$. Since Ψ is constant, the constraint requiring that the mean-squared gradient of ξ equal unity is equivalent to setting

$$\Theta^T \Psi \Theta = 1\ . \tag{7}$$

The parameter vector Θ that minimizes (6) subject to the constraint (7) is given by the solution to the generalized eigenvalue problem [8]

$$\Phi\Theta = \lambda\Psi\Theta\ , \tag{8}$$

where λ is a scalar. Using (6) and (8), the aggregate error $\Xi = \lambda m$. Therefore, the desired set of parameters are contained in the eigenvector corresponding to the smallest eigenvalue. A method for solving (8) which takes into account the singularity of Ψ, is presented in [1].

An estimate for the Cartesian coordinates of the center-of-rotation, with respect to the sensor frame, is obtained by fitting the measured target positions $\vec{p}_{ij}$, for $j = 1, \ldots, m$, to the equation of a circle. Our objective in this estimation problem is to minimize the sum of the perpendicular distances between the measured and the estimated circle. Consider the standard form for the equation of a circle

$$(x - \mathbf{g})^2 + (y - \mathbf{h})^2 = \mathbf{r}^2 , \tag{9}$$

where **g** and **h** are the X and Y coordinates of the center, respectively, and r is the radius. While a circle can lie in three dimensions, it only spans two dimensions. In the form (9), the Z coordinate of the center is thus implicitly zero. To apply (9) or a variation thereof, we must project the measured target positions which will, in general, span three dimensions to an appropriate two-dimensional subspace. Naturally, the required subspace is the plane-of-rotation and a transformation of coordinates will be necessary.

Let the new coordinate frame be denoted by $\mathbb{C}$ and let the homogeneous transformation matrix **R** represent the transformation between the sensor frame and $\mathbb{C}$. By design, the $\vec{a}$ vector component of **R** is the unit normal vector to the estimated plane-of-rotation and the unit direction vectors $\vec{n}$ and $\vec{o}$ are chosen arbitrarily to form an orthogonal right-handed coordinate system. The target positions are transformed to $\mathbb{C}$ according to $\vec{\mathbf{p}}^*_j = \mathbf{R}\vec{\mathbf{p}}_j$ for $j = 1, \ldots, m$. The Z coordinate of the center-of-rotation in frame $\mathbb{C}$ becomes

$$z^*_c = \frac{\mathbf{D}}{(\mathbf{A}^2 + \mathbf{B}^2 + \mathbf{C}^2)^{1/2}} . \tag{10}$$

Since the X-Y plane of $\mathbb{C}$ is parallel to the estimated plane-of-rotation, the X and Y coordinates of $\vec{p}^*_j$ constitute the projection of the target positions.

The equation of a circle (9) can be rewritten as

$$w \equiv x^2 + y^2 = \mathbf{A}x + \mathbf{B}y + \mathbf{C} = \varphi^T\Theta , \tag{11}$$

where $\varphi = [x \ y \ 1]^T$ and $\Theta = [\mathbf{A} \ \mathbf{B} \ \mathbf{C}]^T$. The output of (11) is the squared distance between a point on the circle and the origin. A simple regression of w on x, y, and 1 corresponds to the minimization of

$$\Xi \equiv \sum_{j=1}^{m} \xi_j = \sum_{j=1}^{m} (w - w_j)^2$$
$$= \sum_{j=1}^{m} (\varphi^T\Theta - w_j)^2 , \tag{12}$$

the sum of the squared errors in the output, w. The solution is

$$\Theta = [\Phi^T\Phi]^{-1}\Phi\mathbf{W} , \tag{13}$$

where $\Phi = [\varphi_1 \ \varphi_2 \ , \ldots , \ \varphi_m]^T$ and $\mathbf{W} = [w_1 \ w_2 \ , \ldots , \ w_m]^T$. In general, minimizing (12) is not the same as minimizing the sum of the squared perpendicular distances between the measurements and the circle, unless the origin of the coordinate frame $\mathbb{C}$ and the circle are coincident. Thus, we repeatedly apply the linear solution (13) followed by a transformation of coordinates. At the i th iteration, we translate the X and Y components of the originally projected measurements $\vec{p}^*_j$ by $x^i_j = x^*_j - \mathbf{g}_{i-1}$ and $y^i_j = y^*_j - \mathbf{h}_{i-1}$ where $\mathbf{g}_{i-1}$ and $\mathbf{h}_{i-1}$ are the estimated coordinates of the center-of-rotation computed during the i-1 th iteration. By repeatedly translating the original data during each iteration, the origin of the circle in the translated frame approaches zero (i.e., approaches the origin of the frame). The solution (13) using x^i_j and y^i_j thus approaches the solution to the desired minimization problem. The vector defining the center-of-rotation for joint i in coordinate frame $\mathbb{C}$ is $[\mathbf{g} \ \mathbf{h} \ \mathbf{z}_c]^T$. The center-of-rotation in sensor coordinates is $\vec{p}^c_i = \mathbf{R}[\mathbf{g} \ \mathbf{h} \ \mathbf{z}_c]^T$.

5.3 Feature Accuracy

The accuracy of the angular orientation of an identified plane-of-rotation is a function of the accuracy of the target measurements, the number of measurements N_i and the nominal radius of the Target. By assuming that the residual errors between an identified plane-of-rotation and the actual target measurements are independent with zero mean and variance σ^2_n, the estimates of the plane parameters will be unbiased and expressions for the estimate variances can be derived [6]. Based upon the physical interpretation of these parameters, we can then also derive an expression of the variance in the orientation of an estimated plane-of-rotation. The respective standard deviations translate into confidence intervals which can be compared with manufacturing tolerances to determine whether an identified signature is likely to be more accurate than the robot's design model. If the standard deviation in the orientational accuracy of the planes-of-rotation are significantly greater than the orientational accuracy with which the robot can be manufactured, then it is highly probable that the identified signature will actually degrade the performance of the robot.

Taking into account the limited range of most manipulator joints, a conservative approximation for the variance in the orientational accuracy of an identified plane-of-rotation is given by

$$\sigma^2_\delta \approx \frac{\sigma^2_n}{\mathrm{R}^2_i\, n(.00244X - .344)} , \tag{14}$$

where X represents the joint range of motion in degrees, R_i is the nominal Target radius, and n is the number of measurements. In practice, σ^2_n equals the sensor system noise variance.

The accuracy of the position of an identified center-of-rotation is a function of the accuracy of the target positions measured relative to the joint axis. Since the origin of an S-Model link coordinate frame is permitted to lie anywhere on the joint axis, errors in the identified position of a center-of-rotation in the direction of the joint axis, have no effect upon the accuracy of the identified kinematic parameters. Unfortunately, since the errors in measuring w, are dependent upon x and y, an analytical derivation of the variances for the estimates of the parameters A and B is impossible.

Monte-Carlo simulation techniques have been applied [6] to empirically determine the relationship between the variance in the estimated coordinates of the center-of-rotation, the variance in the measurements, the range of motion of the joint, and the number of measurements. The accuracy of the estimated center-of-rotation is given by the variance

$$\sigma^2_c \approx \frac{\sigma^2_r}{n(.00229X - .313)} , \tag{15}$$

where σ^2_r is the variance of the radial residuals and X is the range of motion of the joint measured in degrees. The accuracy of an estimated center-of-rotation can be increased by either increasing the accuracy of the sensor system or increasing the number of measurements.

Table 1 lists the standard deviations of the angular orientation of the identified planes-of-rotation and the positional standard deviations of the identified centers-of-rotation for a Puma 560 robot with

Table 1: Orientational and Positional Accuracy of the Identified Planes-of-Rotation and Centers-of-Rotation

Joint	Feature Estimate Accuracy	
	Plane-of-Rotation σ_δ (deg)	Center-of-Rotation σ_c(cm)
1	.002937	.00176
2	.007397	.00335
3	.008514	.00250
4	.003204	.00356
5	.004149	.00184
6	.003699	.00176

the identified arm signature, listed in Appendix I. For Joint 6, the orientation of the identified plane-of-rotation is accurate to within $\pm.0074\,deg$. The variance in orientation of an identified plane-of-rotation is significantly less than the machining and assembly variances with which commercially available robots are manufactured. The orientational standard deviation in manufacturing may be on the order of hundredths of a degree or more. This is also true for the positional variance of the identified *centers-of-rotation. For instance, the positional standard* deviation of the identified Joint 6 center-of-rotation is ±0.00176 cm. The significance of (14) and (15) is that they provide approximate guidelines for the selection of the identifier parameters N_i and R_i, and the requirements for sensor system accuracy.

6 Simulation Results

The relationships (14) and (15) are useful for determining and/or indicating whether or not an identified signature model is likely to be a more accurate kinematic description of the robot than is its design model. However, they do not provide us with an understanding of how measurements errors or feature estimate errors propagate into end-effector positioning and orienting errors. The nonlinear relationships which define the inverse kinematics of most robots, such as those for the Puma 560, preclude the derivation of an analytical relationship between feature estimate errors and end-effector positioning and orienting errors. In [6], an alternative approach is taken in which Monte-Carlo simulation techniques are applied to derive empirical models to relate end-effector positioning accuracy to sensor system accuracy. To complement hardware experimentation, the simulator developed was used to evaluate the kinematic performance of the Puma 560 robot with signature-based control.

The S-Model identification algorithm has been applied, in simulation, to identify the kinematic parameters of a Puma 560 robot in the presence of slant range errors and sensor system calibration errors. Values for the input parameters to the simulator were chosen to coincide with our software implementation of the S-Model identification algorithm and the hardware of our prototype system. The simulator uses the Puma 560 design model to compute the actual locations of the target points with respect to the sensor coordinate frame. Zero mean gaussian noise is then added to each of the true slant ranges. The standard deviations of this noise varies as a function of the range. The relationship between the noise standard deviation, σ, and the range, X, was determined experimentally. The S-Model identification algorithm is applied to determine the robot's arm signature 500 times using corrupted slant range measurements. The identified signatures are then used to control the robot to perform a variety of tasks. In [6], 34 performance indices are defined to measure the performance of the robot in these tasks.

Based on these simulations, we find that the standard deviations of all 34 performance indices are approximately inversely proportional to the square root of N_i. By increasing the number of measurements used to identify a robot's arm signature, substantial increases in end-effector positioning accuracy can be achieved. This Our simulations confirm that feature estimate accuracy has a direct effect upon the performance of a signature-based controller.

In the simulation experiments conducted to analyze the effect of target radius variations on performance, all six target radii were set equal to one another and then varied simultaneously. Under these conditions, the standard deviations of all 34 performance indices tend to be inversely proportional to the square of the target radius. In contrast to the effects due to N_i, these results are not consistent with the analytical relationships in Section 5. Intuitively, one might have expected the relationship to be inversely proportional to R and not to R^2. Since R only effects the estimates of the planes-of-rotation, orientational errors in the identification of an arm signature are thus especially critical. The best strategy for increasing signature accuracy is to first increase the target radii followed by increases in the number of measurements.

Our simulation studies have revealed another important feature of the S-Model identification algorithm and control approach. We find that in the presence of zero-mean measurement noise, the S-Model identification algorithm, on the average, identifies the true kinematic parameters of the robot and thus, on the average, provides perfect end-effector positioning accuracy. The average expected performance is independent of the measurement noise standard deviation. In the design model control approach, the *input errors are the manufacturing errors. With this type of* control, the average expected performance of the robot depends both upon the mean and standard deviation of the manufacturing errors. Consequently, the average expected performance will be less than perfect even if the manufacturing errors are zero-mean.

7 Conclusions

A practical prototype arm signature identification system has been implemented and applied to improve the positioning accuracy of seven Puma 560 robots. This system incorporates an ultrasonic range sensor to measure the three-dimensional Cartesian positions of target sparkers placed on the robot's links. The S-Model identification algorithm [6, 7] uses the target position measurements to first identify a set of robot kinematic features and then combines the information contained therein to identify the robot's Denavit-Hartenberg kinematic parameters. This algorithm leads to an efficient method for collecting measurements. Using our prototype system, thousands of measurements can be obtained in less than one hour. The use of our prototype system to increase the performance of actual robots demonstrates that the S-Model identification algorithm is both feasible and practical.

An advantage of the S-Model identification algorithm is that it contains internal features which make it possible to predict whether or not a particular identified arm signature is likely to be more accurate then the robot's original design model. Monte-Carlo simulations techniques have also been developed and applied to obtain a more general understanding of the effects of sensor errors and various identifier parameters on kinematic performance. From this understanding and the predictors of feature estimate accuracy, engineering guidelines to aid in the design and implementation of future arm signature identification systems have been generated.

A key issue in the design of an arm signature identification system is the selection and/or design of a sensor system. With the accuracy of our ultrasonic sensor system we have been able to significantly improve the performance of Puma 560 robots. However, the improvements are not as large as one might expect based upon the feature estimate accuracy predictions listed in Table 1. The disparity is believed to be caused by the presence of small biases in the individual range measurements, and this appears to be the limiting factor in the performance of our prototype system. In order to obtain additional improvements in manipulator kinematic performance, alternative sensor systems may be necessary.

I. Identified Model

In Table I-1, we list as an example the pseudo Denavit-Hartenberg parameters of a Puma 560 identified by the S-Model identification algorithm.

Table I-1: Identified Arm Signature Parameters

Link	Variable	θ_{offset} (deg)	d (cm)	a (cm)	α (deg)
1	θ_1	110.308	-20.678	-0.005	-90.066
2	θ_2	79.977	11,210.213	7.607	-0.217
3	θ_3	-79.483	-11,194.978	-2.039	90.519
4	θ_4	-0.296	43.278	-0.001	-90.007
5	θ_5	-0.497	-0.018	0.002	89.990
6	θ_6	-89.081	10.300	0.000	0.000

References

[1] Agin, G. J.
Fitting Ellipses and General Second-Order Curves.
Technical Report CMU-RI-TR-81-5, Robotics Institute,
Carnegie-Mellon University, Pittsburgh, PA, July, 1981.

[2] Denavit, J. and Hartenberg, R. S.
A Kinematic Notation for Lower-Pair Mechanisms Based
on Matrices.
Journal of Applied Mechanics 77(2):215-221, June, 1955.

[3] Kernighan, B. W. and Ritchie, D. M.
The C Programming Language.
Prentice-Hall, Englewood Cliffs, NJ, 1978.

[4] Paul, R. P.
*Robot Manipulators: Mathematics, Programming, and
Control.*
The MIT Press, Cambridge, MA, 1981.

[5] *GP-8-3D Sonic Digitizer Operator's Manual*
Science Accessories Corporation, Southport, CN, 1985.

[6] Stone, H.W.
*Kinematic Modeling, Identification, and Control of Robotic
Manipulators.*
PhD thesis, Carnegie-Mellon University, Pittsburgh, PA,
November, 1986.

[7] Stone, H.W., Sanderson, A.C., and Neuman, C.P.
Arm Signature Identification.
In *the Proceedings of the IEEE International Conference on
Robotics and Automation*, pages 41-48. April, 1986.
San Fransico.

[8] Strang, G.
Linear Algebra and Its Applications.
Academic Press, New York, 1980.

[9] *Unimate Puma Mark II 500 Series Equipment Manual
398U1*
Unimation Incorporated, Danbury, CN, 1985.

[10] *Val II Programming Language*
Unimation Incorporated, Danbury, CN, 1985.