S20-63
11P.
188140

Inverse Kinematic-Based Robot Control

W.A. Wolovich and K.F. Flueckiger
Brown University
Providence, RI 02912

B1720316

1. Introduction

A fundamental problem which must be resolved in virtually all non-trivial robotic operations is the well-known *inverse kinematic question.* More specifically, most of the tasks which robots are called upon to perform are specified in *Cartesian (x,y,z) space,* such as simple tracking along one or more straight line paths or following a specified surface with compliant force sensors and/or visual feedback. In all cases, control is actually implemented through coordinated motion of the various links which comprise the manipulator; i.e. in *link space.* As a consequence, the control computer of every "sophisticated" anthropomorphic robot must contain provisions for solving the inverse kinematic problem which, in the case of "simple", non-redundant position control, involves the determination of the first three link angles, θ_1, θ_2, and θ_3, which produce a desired wrist origin position p_{xw}, p_{yw}, and p_{zw} at the end of link 3 relative to some fixed base frame, as further explained in [1].

It is well known that the forward kinematic equations (the functional dependence of p_{xw}, p_{yw} and p_{zw} on the θ_i's) usually can be obtained in a relatively straightforward manner using (say) the Denavit-Hartenberg transformation matrices in order to obtain functions G_x, G_y, and G_z in the relation:

$$\begin{bmatrix} p_{xw} \\ p_{yw} \\ p_{zw} \end{bmatrix} \cancel{\triangleq X =} \begin{bmatrix} G_x(\theta_1,\theta_2,\theta_3) \\ G_y(\theta_1,\theta_2,\theta_3) \\ G_z(\theta_1,\theta_2,\theta_3) \end{bmatrix} \triangleq G(\theta) \tag{1}$$

P sub xw, P sub yw and P sub zw

~~However, the converse~~ determination of a particular

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \tag{2}$$

which "solves" (1) for a given or desired X is not nearly as straightforward, although analytical inverse kinematic solutions do exist for virtually all current industrial manipulators. It might be noted, however, that these analytical inverse kinematic solutions are usually non-unique and sequential, and further require the evaluation of some rather complex Atan2 functions---see Summary Sheet 3.4.57 of [1], for example, which presents one such solution for the PUMA 560 industrial manipulator.

We should also note that this problem becomes significantly more complex when the orientation question is addressed simultaneously; i.e. when a desired end effector or tool orientation is specified in addition to its position. Certainly, any technique which can "simplify" solutions to the inverse kinematic question in robotics can have a significant impact not only on the computational requirements involved with robot control, but also on the diversity of tasks the manipulator can perform. *The primary purpose of this paper will be to thoroughly evaluate, extend, and demonstrate a new computational technique for solving the complete configuration (position and orientation) inverse kinematic problem for a variety of multi-link manipulators.* Research

~~In the next section, we will~~ outline ~~this~~ a new inverse kinematic solution and demonstrate its potential via some recent computer simulations. ~~We will~~ also compare it to current inverse kinematic methods and outline some ~~of the~~ remaining problems which will be addressed in order to render it fully operational. ~~In Section 3 we will discuss a~~ Also number of practical consequences of this technique beyond its obvious use in solving the inverse kinematic question. ~~In particular, we will show how~~ a number ~~of diverse but well known robotic problems may be handled by~~ suitable modifications to and/or extensions of this new inverse kinematic result. They

2. A Complete Inverse Kinematic Solution

To motivate the more general, six degree of freedom solution to the inverse kinematic problem associated with the

PAGE 164 INTENTIONALLY BLANK

overall configuration of the end effector, we will first present a solution to the three degree of freedom inverse kinematic problem associated with only the position of (say) the wrist origin associated with the end of the third link of a six link manipulator. The particular solution given here follows directly from that given in [2] with J^T replaced by J^{-1}, as suggested in [2] and later implemented in [3], where J denotes the well known *Jacobian matrix* of the manipulator. More specifically, the time differentiation of (1) directly implies that

$$\dot{X} = \begin{bmatrix} \frac{\partial G_x}{\partial\theta_1} & \frac{\partial G_x}{\partial\theta_2} & \frac{\partial G_x}{\partial\theta_3} \\ \frac{\partial G_y}{\partial\theta_1} & \frac{\partial G_y}{\partial\theta_2} & \frac{\partial G_y}{\partial\theta_3} \\ \frac{\partial G_z}{\partial\theta_1} & \frac{\partial G_z}{\partial\theta_2} & \frac{\partial G_z}{\partial\theta_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J\theta\,, \tag{3}$$

with the Jacobian J being a matrix of partial derivatives, as specified via equation (3).

In light of the preceding, now consider the closed loop dynamical system depicted in Figure 1, which is "driven" by some desired wrist origin position in Cartesian space, namely

$$X_d = \begin{bmatrix} p_{xwd} \\ p_{ywd} \\ p_{zwd} \end{bmatrix} \tag{4}$$

It can be noted that in Figure 1, K might be a (3x3) arbitrary, diagonal, time-invariant gain matrix, $\dot{\theta}_s$ would be a time varying 3-vector system output which represents the derivative of the desired link angle displacement which, when integrated, yields the 3-vector output representative of the link angle displacement, θ_s , and $G(\cdot)$ represents the forward kinematic operator defined by equation (1).

We might next define the equations which describe the dynamical behavior of the Figure 1 system, namely

$$\dot{\theta}_s = J^{-1}(\theta_s)K(X_d - X_s)\,, \tag{5}$$

and

$$X_s = G(\theta_s)\,. \tag{6}$$

Clearly, the premultiplication of (5) by $J(\theta_s)$ and the subsequent substitution of $\dot{X}_s$ for $J(\theta_s)\dot{\theta}_s$, in light of (3), then implies that

$$\dot{X}_s = K(X_d - X_s)\,, \tag{7}$$

or that X_s has a dynamical system representation as depicted in Figure 2. The reader will immediately recognize the system of Figure 2 as a parallel combination of three relatively simple, decoupled, first order, linear, time invariant systems with arbitrarily adjustable (via the elements of K) stability properties. In particular, if X_d represents a step input of magnitude X_d (actually a 3-vector step input), applied at time t_0, then it is easy to show that for zero initial conditions on X_s,

$$X_s(t) = \left[I - e^{-K(t-t_0)}\right]X_d\,, \tag{8}$$

or that for K positive definite, $X_s(t)$ will track the desired Cartesian position $X_d(t) = X_d$ with an (arbitrarily fast) exponentially decaying error! In light of (6), it therefore follows that $\theta_s(t)$ can be made to approach the desired θ_d which corresponds to $X_d = G(\theta_d)$ arbitrarily fast as well.

The reader might next note that in order to make this inverse kinematic procedure applicable to more general forms of robotic motion, it has to be "extended" to include inverse orientation information as well; i.e. solutions for θ_4, θ_5, and θ_6 of a general six link manipulator. However, the extension of the Figure 1 system to include orientation as well as position is a non-trivial task, since (i) there is no 3-vector representation for orientation and (ii) even if there were, the Figure 1 system would then require an analytical expression for the inverse of a corresponding (6x6) Jacobian, a formidable computational task. In light of these observations, we will now present, for the first time, a complete dynamical system solution to the inverse kinematic problem for both position and orientation.

To begin, we first note that the orientation of (say) the tool frame relative to the fixed base frame can be, and often is, specified by an appropriate (3x3) orientation coordinate transformation matrix, often called a *rotation matrix*, of the form (using the notation in [1]):

$$R_0^t = \begin{bmatrix} a, n, s \end{bmatrix} = \begin{bmatrix} a_x & n_x & s_x \\ a_y & n_y & s_y \\ a_z & n_z & s_z \end{bmatrix}, \tag{9}$$

where the orthogonal unit vectors a, n, and s represent the ***approach***, ***normal***, and ***sliding*** vectors associated with the orientation of the tool frame relative to some fixed base frame. Furthermore, since s can be obtained via the vector cross-product relationship:

$$a \times n = s, \tag{10}$$

as described in [1], knowledge of a and n alone will uniquely specify orientation of the end effector.

We next note that if

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{11}$$

represents the angular velocity of the tool frame, then it is not difficult to show, in light of Figure 3, that ω can be represented by the sum of its "translational component" relative to the motion of a, namely the cross product $a \times \dot{a}$, where $\dot{a} = \frac{da}{dt}$, and its "rotational component" relative to the motion of a, namely the scalar velocity dot product $\dot{n} \cdot s$ multiplied by a; i.e.

$$\omega = a \times \dot{a} + (\dot{n} \cdot s)a\,. \tag{12}$$

Furthermore, it now follows by expanding (12) in light of (9) that ω is also given by the following matrix-vector product:

$$\omega = \begin{bmatrix} s_x a_x & s_y a_x & s_z a_x & 0 & -a_z & a_y \\ s_x a_y & s_y a_y & s_z a_y & a_z & 0 & -a_x \\ s_x a_z & s_y a_z & s_z a_z & -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} \dot{n}_x \\ \dot{n}_y \\ \dot{n}_z \\ \dot{a}_x \\ \dot{a}_y \\ \dot{a}_z \end{bmatrix} \triangleq F(\theta) \begin{bmatrix} \dot{n} \\ \dot{a} \end{bmatrix} \tag{13}$$

The results which now follow build on the material presented in Section 4.3 of [1] which pertains to so-called *spherical wrist manipulators*. In such cases, (4.3.2) of [1] establishes the fact that

$$\dot{X} \triangleq \begin{bmatrix} \dot{p}_{xw} \\ \dot{p}_{yw} \\ \dot{p}_{zw} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{p}_w \\ \omega \end{bmatrix} = J\dot{\theta} = \begin{bmatrix} J_P & 0 \\ J_{\bar{R}} & J_R \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix}, \tag{14}$$

or that the (6×6) Jacobian matrix associated with spherical wrist manipulators can be "triangularized", with J_P a (3×3) "positional" Jacobian associated with the velocity of the wrist origin relative to motion of the first three links, and J_R a (3×3) "orientation" Jacobian associated with the angular velocity of the end effector frame relative to the motion of the final three links.

If we now "invert" (14), it follows that

$$\dot{\theta} = J^{-1}\begin{bmatrix}\dot{p}_w \\ \omega\end{bmatrix} = \begin{bmatrix} J_P^{-1} & 0 \\ -J_R^{-1}J_{\bar{R}}J_P^{-1} & J_R^{-1}\end{bmatrix}\begin{bmatrix}\dot{p}_w \\ \omega\end{bmatrix}, \tag{15}$$

or, in light of (13), that

$$\dot{\theta} = \begin{bmatrix} J_P^{-1} & 0 \\ -J_R^{-1}J_{\bar{R}}J_P^{-1} & J_R^{-1}\end{bmatrix}\begin{bmatrix} I_3 & 0 \\ 0 & F(\theta)\end{bmatrix}\begin{bmatrix}\dot{p}_w \\ \dot{n} \\ \dot{a}\end{bmatrix} \triangleq J_I\begin{bmatrix}\dot{p}_w \\ \dot{n} \\ \dot{a}\end{bmatrix}, \tag{16}$$

with the (6x9) "inverse" Jacobian, J_I, given by the product of the (6x6) triangular inverse Jacobian defined by (15) and the (6x9) "block diagonal" matrix consisting of an upper left (3x3) identity matrix I_3, and a lower right (3x6) $F(\theta)$, as defined via (13).

We next note that the 9-vector "configuration"

$$\underline{X} \triangleq \begin{bmatrix}p_w \\ n \\ a\end{bmatrix} = \underline{G}(\theta) \tag{17}$$

for a known $\underline{G}(\cdot)$, with corresponding

$$\dot{\underline{X}} = \begin{bmatrix}\dot{p}_w \\ \dot{n} \\ \dot{a}\end{bmatrix}. \tag{18}$$

As defined, $\underline{X}$ completely specifies both the (wrist origin) position† and the (end effector) orientation of any given manipulator.

Now consider the dynamical system depicted in Figure 4, which we claim "solves" the inverse kinematic problem associated with the complete configuration of six link, spherical wrist manipulators. In particular, the dynamical equations associated with Figure 4 are

$$\dot{\theta}_s = J_I(\theta_s)K\left[\underline{X}_d - \underline{X}_s\right], \tag{19}$$

with K a diagonal (9×9) gain matrix, and

$$\underline{X}_s = \underline{G}(\theta_s)\,. \tag{20}$$

Since $\dot{\theta}_s$ is also equal to $J_I(\theta_s)\dot{\underline{X}}_s$, in light of (16) and (18), with $\dot{\underline{X}}_s$ arbitrary, (19) implies that

$$\dot{\underline{X}}_s = K\left[\underline{X}_d - \underline{X}_s\right], \tag{21}$$

or that the 9-vector $\underline{X}_s$ is analogous to the 3-vector X_s of (7). This in turn implies that $\underline{X}_s$ will track the desired Cartesian configuration $\underline{X}_d$ with an (arbitrarily fast) exponentially decaying error! As we noted earlier, it therefore follows that $\theta_s(t)$ can be made to approach the desired θ_d which corresponds to $\underline{X}_d = \underline{G}(\theta_d)$ arbitrarily fast as well. In other words, *the Figure 4 dynamical system solves for the first time the complete inverse kinematic problem for virtually any six link, spherical wrist manipulator.*

Figures 5 and 6 depict actual simulated runs of the Figure 4 system for the PUMA 560 industrial manipulator, as mathematically described in [1], when the (end effector) position vector (p_x, p_y, p_z) goes from (3.5, 2.5, 2.9) at $t_0 = 0$ to (1.5, 2.0, 4.4) at $t_f = 5$ along a LSPB (Linear Segment with Parabolic Blend) trajectory‡ while the orientation of the end effector frame undergoes a simultaneous smooth transition for $(n_x, n_y, n_z, a_x, a_y, a_z)$ from (0, 0, -1, 0, 1, 0) to (-1, 0, 0, 0, 0, -1) over the same 5 second time interval. Only four of the nine elements of $\underline{X}$ are explicitly depicted, and in both cases,

(†)The term *link space* rather than *joint space* will be used here for reasons which are delineated in Section 1.4 of [1].

† Of course, for spherical wrist manipulators, knowledge of the wrist origin position and a, the approach vector, directly implies knowledge of the

the initial conditions on θ_s were appropriately set to insure that $\underline{X}_s(0) = \underline{X}_d(0)$. It might be noted that in the Figure 5 runs all nine of the nonzero, diagonal elements of K were set equal to 10, while these same nine elements were increased to 100 for the Figure 6 runs. The reader will note that a rather small error exists between the desired and simulated dynamical system configuration parameters depicted in the K=10 case. Moreover, this small error is essentially eliminated by increasing the (elements of the) diagonal gain matrix K to 100, as depicted in Figure 6; i.e. the desired and simulated configuration parameters are so close that they are virtually undistinguishable in this latter case! This, in turn, implies corresponding dynamical system link output displacement values which are "very close" to those which would mathematically solve the inverse kinematic question for the given, desired $\underline{X}_d = \begin{bmatrix} p_{wd} \\ n_d \\ a_d \end{bmatrix}$, especially in the K=100 case. In summary, therefore, *Figures 5 and 6 clearly illustrate the employment of the Figure 4 dynamical system as a viable alternative technique for solving the inverse kinematic question for a large class of multi-link manipulators.*

A number of observations are now in order relative to this dynamical system inverse kinematic solution. First of all, we note that $\dot{\theta}_s$ is also obtained as an output of our dynamical system solution *without explicit knowledge or use of* $\dot{\underline{X}}$! This could prove most useful in the implementation of a variety of control schemes which require desired link velocities as well as positions; e.g. in relatively simple PID controllers, where D denotes the (time) derivative of the link positional drive signal.

We next note that because of the spherical wrist assumption, we actually can determine an analytical expression for the (6×9) "inverse" Jacobian, $J_I(\theta)$, as defined by equation (16). For example, such an analytical expression is essentially given in Example 4.3.23 of [1] for the PUMA 560 industrial manipulator. Certain earlier reports and texts have implied that analytical expressions for J^{-1} in the six link case are virtually impossible to obtain. In [1] we show that this is not necessarily the case for spherical wrist manipulators, and here we exploit this observation to extend a three-dimensional inverse kinematic positional result to the more general and important, six-dimensional configuration case.

We further note that the particular inverse kinematic (position and velocity) solutions we obtain via the dynamical system of Figure 4 will be *unique*, and will depend on the initial conditions associated with the system. Different initial conditions can be used to produce all of the solution sets associated with a given manipulator, if desired, or only the particular one "best suited" to a specific task, such as (say) an arm right, elbow above trajectory for the PUMA 560 (see Figure 3.4.56 of [1]).

We finally observe, again in light of Figures 5 and 6, that there is no need to sequentially solve a set of rather complex and time-consuming Atan2 functions associated with a given robot to obtain the inverse kinematic link displacements associated with a desired Cartesian configuration. Although the computational savings associated with the direct employment of the Figure 4 dynamical system, rather than the explicit solution of a sequential set of Atan2 functions, has yet to be completely determined, there is reason to believe that such savings can be rather significant.

3. Practical Consequences to be Investigated

There are numerous practical consequences associated with the new computational inverse kinematic procedure which has just been outlined, and the primary purpose of this section will be to delineate some of them. To begin, we might again note the obvious, namely that the procedure can be directly utilized *to produce desired link positional and velocity drive signals to the link motors* which then might be controlled by any "standard procedure", such as a unity feedback PID compensator, without the explicit evaluation of any analytical Atan2 functions. Of course, in such cases and in the others which we will outline in this section, it is important to realize that a flexible control computer must be employed in order to physically realize (say) the Figure 4 feedback system. In light of this observation, it is of interest to note that a truly significant amount of robotics development effort within LEMS at Brown University over the past year has focused on the development of one such flexible control computer for robotic applications, namely SIERA (System for Implementing and Evaluating Robotic Algorithms).

SIERA is a unique multiprocessor system composed of two subsystems--a tightly-coupled real-time servo system and a loosely coupled multiprocessor network (the "Armstrong system"), as depicted in Figure 7. A shared memory

end effector position as well, as is shown in [1].

‡ Both references [1] and [6] describe such LSPB trajectories.

interface allows communication between these two subsystems. The architecture is flexible enough to accommodate a variety of robots and sensors, since all robot dependent hardware is restricted to the robot interface board. Thus, we have been able to control the Unimation Puma 560 and the IBM 7565 robots that are currently in our laboratory. A detailed description of the SIERA hardware can be found in [4].

The SIERA operating system provides a flexible development system for research in robotic algorithms, without making the system too complex to be used for instructional purposes. This is accomplished by defining three different programming levels: i) the user level, which is analogous to a commercial system such as Unimation's VAL robot command language, ii) the researcher level, which fulfills the main objective of SIERA by allowing any type of robotic algorithm to be added to the system, and iii) the expert level, which is used to add a new robot or to enhance the operating system. It should be noted that the operating system is also generally applicable since all (low-level) robot tasks are handled by interface routines written by an expert level programmer. Further details of the operating system and the programming levels can be found in [5].

Another potential use for our inverse kinematic procedure which has yet to be fully exploited is in *the automatic avoidance of degenerate configurations*, such as those associated with Jacobian singularities. To be more specific, it is well known that certain desired Cartesian trajectories may imply corresponding link trajectories for which $|J(\theta)|$, the determinant of the Jacobian, approaches zero. In such cases, excessive link velocities are required to produce seemingly well-behaved Cartesian motion. We feel that one way of automatically avoiding such degenerate configurations could be to physically restrict the magnitude that $|J(\theta_s)|$ can decrease to in either the Figure 1 or the Figure 4 system. Although such a procedure will not yield the desired Cartesian trajectories, hopefully it will yield "acceptable" Cartesian trajectories which are "close to" the specified ones. Some preliminary computer simulations bounding $|J(\theta_s)|$ have produced rather encouraging results, and one of the primary objectives of our continuing research will be to thoroughly investigate this and other automatic degenerate configuration avoidance techniques.

Another potential use of our inverse kinematic procedure is that associated with *redundant manipulators*; i.e. manipulators which have more degrees of freedom than are necessary to achieve (say) desired end effector orientations. To be more specific, it is well known that the inverse kinematic problem associated with redundant manipulators can have an infinite number of solutions, and the problem then becomes one of appropriately selecting the "best" solution from this infinite set. It might be noted that one way of obtaining a variety of different link solutions, (say) in light of Figure 1, is to employ "different right inverse" Jacobians instead of the square $J^{-1}(\theta_s)$ depicted. Our investigations are continuing to determine how a "best right inverse" Jacobian might be selected and utilized in our computational inverse kinematic procedure in order to automatically yield a correspondingly "best" inverse kinematic solution for redundant manipulators.

Another potentially important application of our computational inverse kinematic procedure concerns *its employment in more sophisticated control strategies* where knowledge of $\ddot{\theta}_s(t)$, as well as $\dot{\theta}_s(t)$ and $\theta_s(t)$, would be used. One such example is that associated with the inverse dynamic, feedforward compensation procedure outlined in Section 8.5 of [1]. We have already conducted some preliminary simulations of an "extended" version of the Figure 1 and Figure 4 dynamical systems ("extended" by the addition of another parallel bank of integrators as well as appropriate feedback gain matrices) in order to produce $\ddot{\theta}_s$ as well as $\dot{\theta}_s$ and θ_s. One such "extended" system is depicted in Figure 8 in its simplest (positional) form. The mathematical equations associated with such a dynamical system can directly be shown to imply an analogous linear, time-invariant, second order differential equation relationship between input $X_d(t)$ and output $X_s(t)$, namely

$$\ddot{X}_s(t) + A\dot{X}_s(t) + KX_s(t) = X_d(t) , \tag{22}$$

which can then be used to establish convergence relations between $\theta_s(t)$ and its derivatives and the desired $\theta_d(t)$ and its derivatives. Results in this area are still under development. In particular, we are currently working on a more complete mathematically understanding of the Figure 8 system, including the implications regarding the $\theta_s(t)$, $\theta_s(t)$, and $\theta_s(t)$ thus obtained, when compared to the desired values of $\theta_d(t)$ and its derivatives in both the simple (positional) case depicted and the full six degree of freedom configuration case. Here again, our initial simulations have been encouraging and we are actively continuing these investigations.

4. Summary

We have now outlined a new computational procedure for solving the inverse kinematic question for a large class of multi-link manipulators. Furthermore, we have mathematically established the "equivalence" between this computational procedure and the behavior of relatively simple first and second order, linear, time-invariant dynamical systems. We have indicated a number of potential practical consequences associated with the employment of this technique in robotic applications, namely:
(i) its use in directly obtaining unique values for the inverse kinematic positions, velocities, and accelerations,
(ii) its potential for automatically avoiding degenerate configurations,
(iii) its ability to produce the "best" inverse kinematic solutions for redundant manipulators, and
(iv) its employment in more sophisticated motion control strategies.

We have expended a considerable amount of time and effort within LEMS in constructing a general purpose, flexible robot control system (SIERA) which can be used to thoroughly implement, test, and evaluate all aspects of our robotics research program, and we have two industrial manipulators (a PUMA 560 anthropomorphic robot and an IBM RS/1 Cartesian robot) to employ in our studies. Our investigations are well underway, and we are very optimistic that significant new techniques for robot control and manipulation will result as a consequence of these investigations.

References

[1] Wolovich, W. A., *Robotics: Basic Analysis and Design*, Holt, Rinehart and Winston, 1986.

[2] Elliott, H. and Wolovich, W. A., *A Computational Technique for Inverse Kinematics*, Proceedings of the 1984 Conf. on Decision and Control, Las Vegas, NV Dec. 1984.

[3] Vaccaro, R. J. and Hill, S. D., "A Joint-Space Command Generator for Cartesian Control of Robotic Manipulators," Department of Electrical Engineering, University of Rhode Island, 1986, (to appear).

[4] Kazanzides, P., Wasti, H., Weinstein, R., and Wolovich, W. A., "SIERA: System for Implementing and Evaluating Robotic Algorithms," Brown University Technical Report LEMS-23, June, 1986. Also to be presented at the 1987 IEEE International Conference on Robotics and Automation in Raleigh, NC March 30 - April 3, 1987.

[5] Kazanzides, P., Wasti, H., and Wolovich, W. A., "SIERA: A Multiprocessor System for Robotics," Proceedings of the SPIE Symposium on Advances in Intelligent Robotic Systems, Cambridge, Mass., Oct. 1986.

[6] Craig, John J., *Introduction to Robotics: Mechanics & Control*, Addison-Wesley Publishing Company, 1986.

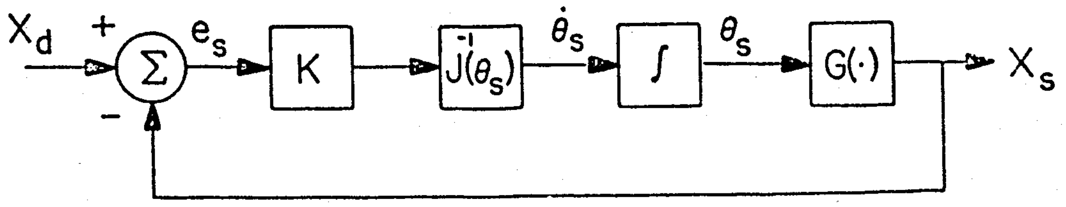

Figure 1

A Positional Inverse Kinematic Solution

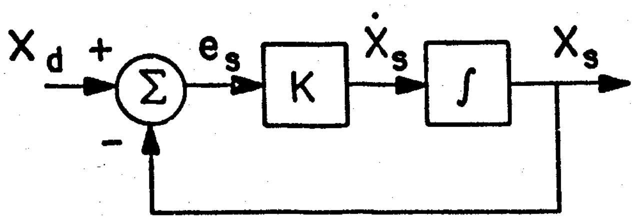

Figure 2

A Dynamical System Representation for X_s

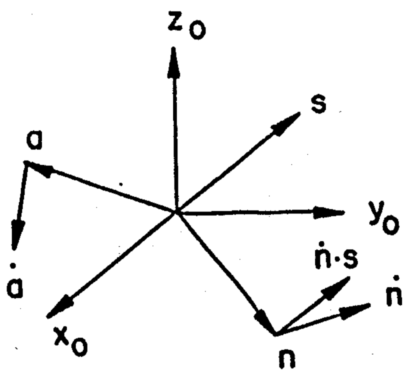

Figure 3

Rotation of the Tool Frame Relative to the Base Frame

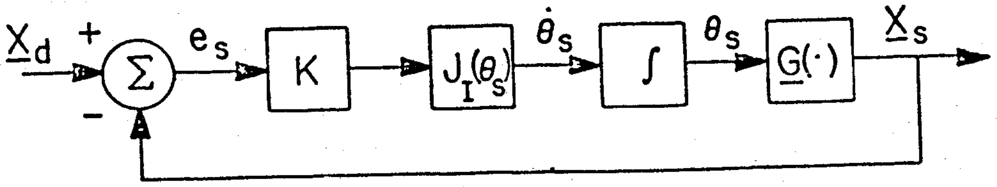

Figure 4

A Configuration Inverse Kinematic Solution

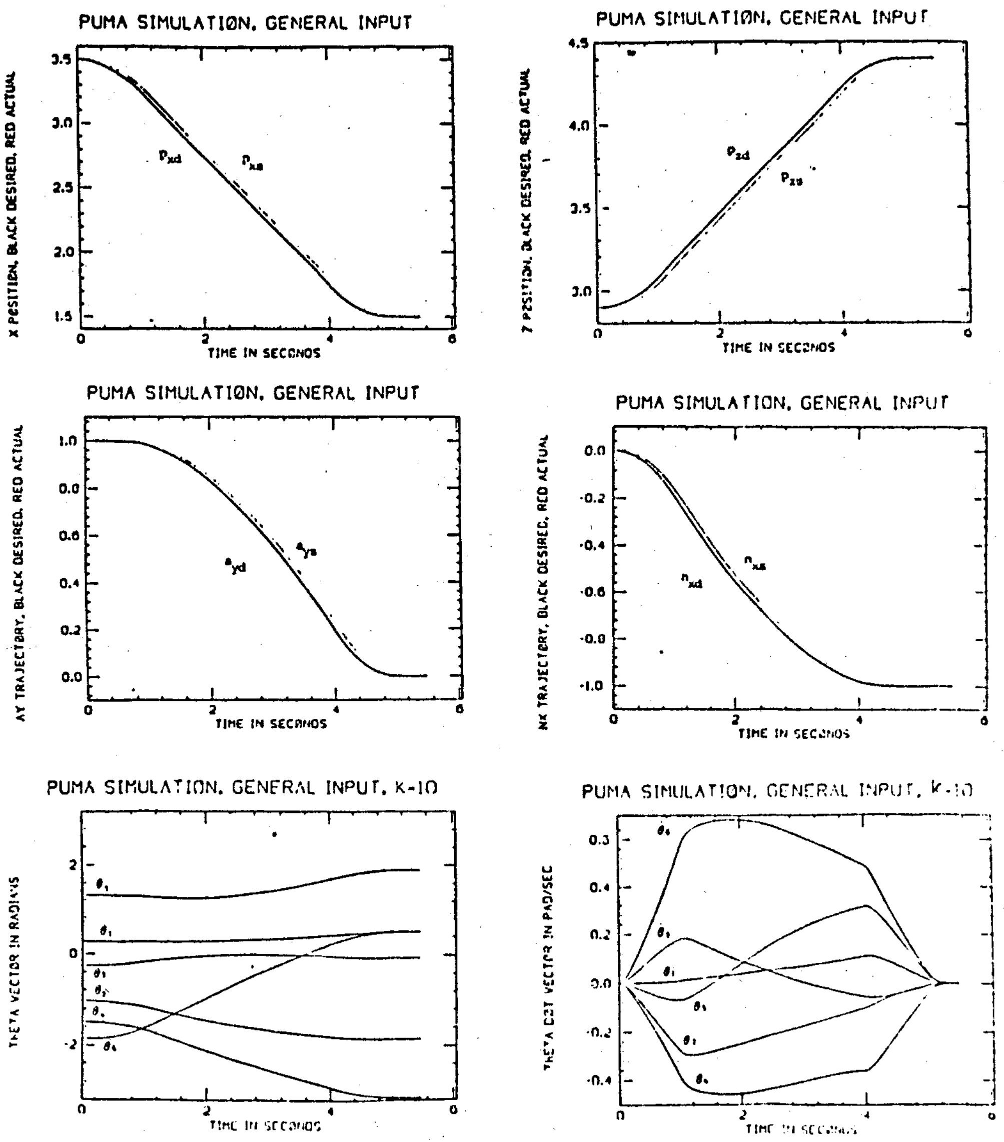

Figure 5
Puma Simulation with K=10

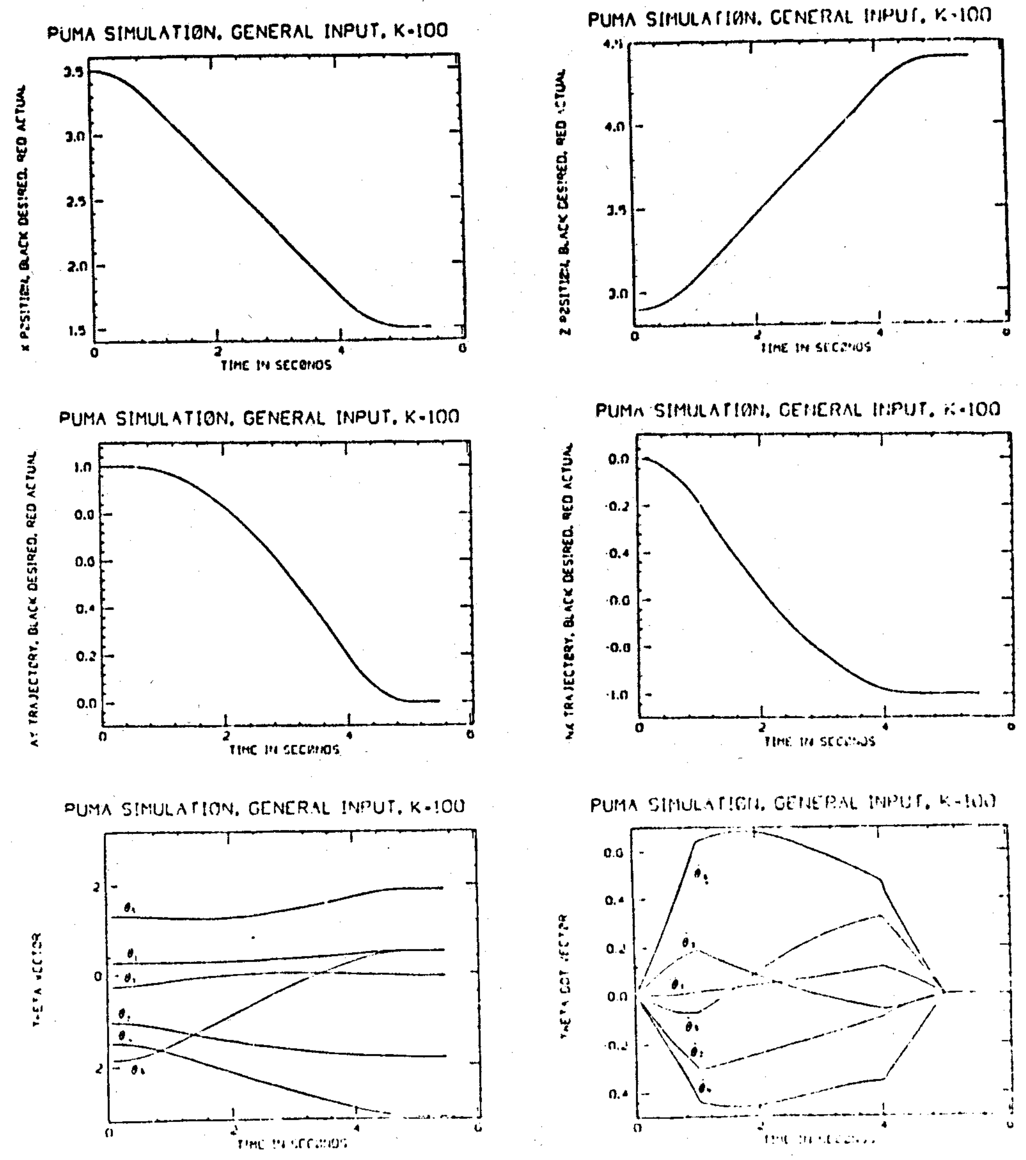

Figure 6

Puma Simulation with K=100

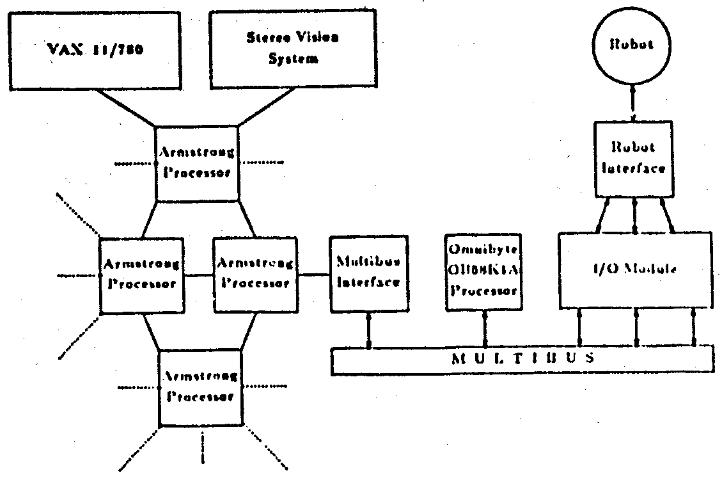

Figure 1. Overview of SIERA.

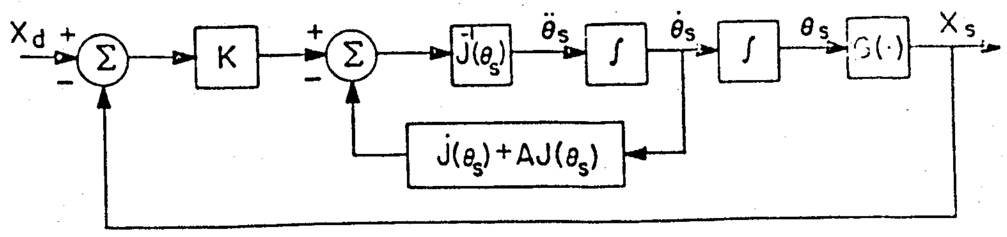

Figure 8
An "Extended" Positional Inverse Kinematic Solution