NASA
Technical
Paper
2807

February 1988

A Generalized Method for Automatic Downhand and Wirefeed Control of a Welding Robot and Positioner

Ken Fernandez
and George E. Cook

NASA

A Generalized Method for Automatic Downhand and Wirefeed Control of a Welding Robot and Positioner

Ken Fernandez
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

George E. Cook
Vanderbilt University
Nashville, Tennessee

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DEFINITION OF SYMBOLS

Symbol	Definition
A_{Di}	(i)th 4x4 Link Transform of Mechanism D
[J]	Mechanism Jacobian matrix
[K]	Positive Definite diagonal scaling matrix
$[M^T]$	Transpose of matrix M
$[M^{-1}]$	Inverse of matrix M
$T_D(j)$	Product A_{Di} for all i in the (j)th time interval
$\underline{g}$	Gravity gradient vector
$\underline{v}$	Four dimensional position vector
$\underline{a} \cdot \underline{b}$	Scalar or dot product of two vectors
$\underline{a} \times \underline{b}$	Vector or cross product of two vectors
$\| \underline{a} \|$	Norm or length of a vector
$\underline{a} \,\|\, \underline{b}$	Parallel condition between vectors $\underline{a}$ and $\underline{b}$
$\underline{\theta}$	Vector composed of mechanism joint angles
θ_i	(i)th Mechanism joint angle
$\underline{\theta}^*$	Ideal mechanism joint solution
$\underline{\theta}_j$	(j)th Estimate of ideal mechanism joint solution

UNUSUAL TERMS

downhand welding - Welding with the part maintained in an orientation such that the weld puddle lies in a horizontal plane.

GTAW - Gas tungsten arc welding is performed with a tungsten electrode in an inert gas atmosphere. This process is also call TIG or tungsten inert gas welding.

off-line programming - A method of specifying the motion path of a robot in an external computer.

wire feed - When welding with a consummable electrode, the electrode is usually supplied to the welding process in the form of a wire fed into the arc directly in front of the electrode as it moves along the weld seam.

A GENERALIZED METHOD FOR AUTOMATIC DOWNHAND AND WIREFEED CONTROL OF A WELDING ROBOT AND POSITIONER

INTRODUCTION

Automated arc welding imposes several requirements on robot/positioner controls that are unique to many other applications. The most important of these are: (a) the desirability to weld in such a position that the effect of gravity on the molten weld pool is to aid the metal fusion process rather than to impede it, (b) the need to maintain constant travel speed of the torch relative to the part to insure a constant heat input to the weld, (c) for Gas Tungsten Arc Welding (GTAW) processes, the need to maintain proper weld wire orientation, and (d) the need to accurately track the weld joint to insure proper fusion at all points along the weld. The latter requirement stems from the need to adapt to variations between the joint and the pre-programmed path of the manipulator due to thermal distortion, dimensional variations in the position of the weld joint on the welded assembly, dimensional variations in the weld assembly, variations in the weld fixturing, and variations in the equipment (i.e., the manipulator and the positioner). Cook [1], Richardson, Farson, Jones, and Rogers [2], and others have addressed the joint tracking problem from the standpoint of both optical and electrical arc sensing.

The purpose of the algorithm described in this paper is to address the other three requirements of an automated arc welding system, i.e., welding in the downhand position, maintaining weld travel speed, and in GTAW proper weld wire orientation. A transformation algorithm is derived that permits programming the weld path teach points with the assembly in any convenient position. It is only necessary to position the torch at each programmed position such that it is at the desired orientation relative to the weld joint. The transformation algorithm computes the required kinematic transformation

matrix for the manipulator, the manipulator velocity, the position vector for the positioner axes, and the velocities of the positioner axes to insure that the weld is performed in the downhand position and that the weld travel speed is constant with wire feed orientation controlled. The algorithm is applicable to any arc welding robot/positioner system. It is only necessary to be able to download the kinematic transformation matrix, positioner axes values, and velocity information to the robotic controller.

The weld process requirements are discussed first. This is followed by a derivation of the algorithms, results of a numerical simulation coupled to the graphics simulation, and a discussion of an implementation of the algorithms in hardware. Applications are described for welding of the Space Shuttle Main Engine.

DOWNHAND WELDING

For most arc welding processes the highest deposition rate can be obtained when welding is performed in the flat (or downhand) position. The downhand position is defined as that position in which welding is performed from the upper side of the joint and the face of the weld is approximately horizontal [3]. The highest deposition rate is achievable in the flat position because gravity acts to keep the molten metal in the joint. This permits the use of higher welding current, with a resulting larger molten weld pool. Studies of flat position welding versus welding in other positions have proven that a heavy weldment can be made in approximately one-half the time if all of the welds are made in the flat position [4].

Another reason for positioning the workpiece for downhand welding is to improve weld quality. The higher deposition rates achievable with downhand welding mean fewer passes are required to complete a multi-pass weld. This in turn can be expected to reduce the amount of distortion that may be encountered. Also, the desired weld bead contour is much more easily maintained in the flat position than in other positions. For example, a fillet weld is defined as a weld of approximately triangular cross section joining two

2

surfaces approximately at right angles to each other [5]. In the downhand position, each surface would be at an angle of approximately 45-degrees with respect to the horizontal. The face of the weld, corresponding to the hypotenuse of the triangular cross section, would be horizontal. In the flat position, gravity would act to maintain this desired weld bead profile. On the other hand, if one of the surfaces being welded lies in the horizontal plane then the other surface would be in the vertical plane and the face of the weld would be undesirably at an angle of 45-degrees with respect to the horizontal. This position of welding is termed the horizontal position [5]. The effect of gravity is now to cause the molten weld pool to sag toward the horizontal surface. This may result in an unacceptable weld profile and possible under-cutting in the vertical surface.

POSITIONING OF WELD ASSEMBLY

Two types of positioner controls are typically used for automatic, robotic arc welding. With one type, the weld assembly is indexed to a programmed weld position, and the weld is performed by moving the manipulator only. When the weld is completed the assembly is indexed to the next programmed weld position, and the next weld is performed. This scenario is repeated for all welds (of relatively straight contour) on the assembly. For circular welds the assembly is indexed to a programmed welding position and then rotated while the weld is performed. Even though time is wasted during indexing intervals (when welding is not performed), this type positioner control is adequate for assemblies containing relatively short, straight weld contours or circular contours. The programmed welding positions may be chosen such that the welding is performed in the downhand position. Programming is relatively easy with a teach pendant. Constant weld travel speed (and hence constant heat input) is easily maintained because the weld contours are assumed to be straight or circular and because the positioner and the manipulator are not in motion simultaneously.

With the other type of positioner control the motion of the positioner is coordinated with that of the manipulator such that both can be in motion simultaneously. This type of control is particularly important for assemblies containing irregular joint contours. Typically, the control algorithms (used with commercially available robots) permit manipulator motion from one programmed point $p(i)$ to a second programmed point $p(i+1)$ along a straight line (relative to robot base coordinates) at a constant velocity $[p(i+1)-p(i)]/T$. The velocities of the positioner axes are automatically set to be completed in the same time interval T. This capability within itself does not guarantee either downhand welding or constant weld travel speed. However, the former can be obtained by using the teach pendant to position the weld assembly (for each point to be programmed) such that the weld is in the flat position at that point. At each such point the manipulator is located (via the teach pendant) to provide correct position and orientation of the weld torch. This is a tedious task and does not ensure that the welding travel speed will be constant. Indeed, the welding travel speed will not be constant on any but the simplest of weld contours. This is due to the relative motion (between the assembly and the torch) that is not taken into account in establishing the manipulator travel speed. GTA welding further complicates the robot programming problem since filler wire must be fed into the arc plasma in a controlled manner. For most applications the filler wire is fed directly in front of the moving plasma. As stated in the introduction, these problems (i.e., maintenance of downhand position, constant travel speed measured with respect to the weld joint and proper weld wire orientation) are overcome with the transformation algorithm discussed in this paper.

OVERVIEW OF ALGORITHM DEVELOPMENT

The algorithms described use an iterative method to determine the robot and positioner joint angles needed to traverse a weld path program, maintain the orientation of the part so that welding in the downhand position, and maintain weld wire orientation. This

4

functionality is achieved in three steps by the development of three algorithms each of which achieved more control of the robot/positioner than its predecessors. The first algorithm maintains torch to part position/velocity control using the eight degrees of freedom of the mechanism formed by combining the positioner and robot. The additional constraint of minimizing the weighted sum-of-squares of joint angle displacement is used to eliminate the redundancy inherent in the combined eight DOF mechanism. The second algorithm achieves torch position/velocity while simultaneously controlling the orientation of the part so that welding is performed in the downhand position. The second algorithm differs from those previously published [6,7,8] in that now robot and positioner are treated as a single mechanism. The third algorithm is similar to the second with the added constraint of maintaining weld wire feed orientation.

DEFINITIONS

We begin our description of the algorithm by stating several definitions. The homogeneous transformation is commonly used in the robotics literature [9,10] and consists of a 4 by 4 matrix, $^{j-1}A_j$ or more simply A_j, as shown in equations (1) and (2).

$$A_j = [\; \underline{n} \;\; \underline{o} \;\; \underline{a} \;\; \underline{p} \;] \tag{1}$$

$$A_j = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The upper left 3 by 3 partition of A_j is a rotational transformation of a vector space, j, whose orthogonal basis are the vectors $\underline{n}$, $\underline{o}$, and $\underline{a}$, with respect to the j-1 frame. Vector $\underline{p}$ defines the displacement of the origin of frame j with respect to frame j-1. In defining the motion of a robot reference, frames are defined fixed within each link [11] and the position and orientation of an N link robot's hand reference frame is defined in the base coordinate frame by T_R in equation (3).

5

$$[T_R(i)] \quad = \quad A_{R1}A_{R2}A_{R3}\cdots A_{RN} \tag{3}$$

Each link transformation, A_j, is a function of the (j)th joint angle for a revolute joint or the (j)th extension for a prismatic (sliding) joint. A similar transformation may be defined for the robot's M DOF positioner given in equation (4).

$$[T_P(i)] \quad = \quad A_{P1}A_{P2}A_{P3}\cdots A_{PM} \tag{4}$$

ALGORITHM I: TORCH POSITION CONTROL

The position of a tool center point (in our case a welding torch) may be defined by a series of transformations which includes T_R. Figure 1 depicts a transform graph of our robot workcell. The origin of our workcell is designated by point O in the figure. Z_R, a fixed transformation, defines the position of the robot's base coordinate frame with respect to the workcell frame. $[T_R(i)]$ is the robot transformation as defined in equation (3) with the variable, i, indicating the (i)th programmed point in our welding path. Finally, the homogeneous fixed transformation, E, is the end-effector or torch transformation. In defining E in the same manner as equation (1) the vector, $\underline{a}$, is chosen to be along the torch plasma flow and the vector, $\underline{p}$, is the point in the torch plasma where contact with the seam is required.

In a similar manner the position of the part weld path may be defined by a series of transformations: Z_P is a fixed homogeneous transformation which translates the part positioner's base reference frame to the workcell frame; $[T_P(i)]$, the positioner transformation matrix defined as in equation (4); [G], the part frame to positioner mounting frame transformation; and [P(i)], the transformation which relates the desired position of the weld torch reference frame (at the (i)th program point) to the part reference frame. The transform, [P(i)], may be determined either "on-line" by using a teach

6

pendant to move the robot through the desired torch positions (not necessarily maintaining the downhand position), or [P(i)] may also be determined by some "off-line" technique (e.g., interaction with a CAD/CAM system).

The condition that the welding torch is located at the desired path program point, represented by the matrix [P(i)], is met when equation (5) is satisfied.

$$Z_R[T_R(i)]E = Z_P[T_P(i)]G[P(i)] \tag{5}$$

Re-arranging equation (5) to form equation (6) and referring to figure (1) we note that the desired torch position, [P(i)], is equal to the actual torch position when there is no error.

$$P(i) = [G^{-1}][T_P(i)^{-1}][Z_p^{-1}][Z_r][T_r(i)]E \tag{6}$$

Expressing equation (6) in terms of our link transformations we get equation (7). (Note: For this example the robot has six DOF's and the positioner has two DOF's.)

$$P(i) = [G^{-1}][A_{2p}^{-1}][A_{1p}^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r} A_{4r} A_{5r} A_{6r} E \tag{7}$$

An error in torch position may be expressed by defining the transformation, [Err], as the transformation between the desired torch position and the actual position. Equations (8) and (9) represent this more general condition.

$$P(i) = [Act][Err] \tag{8}$$

$$[Act] = [G^{-1}][A_{2p}^{-1}][A_{1p}^{-1}][Z_p^{-1}]Z_rA_{1r}A_{2r}A_{3r}A_{4r}A_{5r}A_{6r}E \qquad (9)$$

Figure (2) depicts our modified workcell transform graph in which the torch position error is represented. This error may be determined easily from the command, [P(i)], and the [Act] matrices as shown in equation (10). We may also represent the error, [Err], as a translation and three rotations about the x,y and z axes shown in equation (11).

$$[Err] = [Act^{-1}][P(i)] \qquad (10)$$

$$[Err] = [Trans(dx,dy,dz)][Rotx(d\phi_x)][Roty(d\phi_y)][Rotz(d\phi_z)] \qquad (11)$$

For very small rotational errors, $d\phi_x$, $d\phi_y$ and $d\phi_z$, the order of rotation becomes less distinct. This may be shown by expanding equation (11) in terms of sines and cosines and using the small angle approximations: cosine($d\phi$) is equal to one minus $d\phi$; and sine($d\phi$) is equal to $d\phi$. Terms involving higher orders of $d\phi$ are then eliminated resulting in the small angle approximation of [Err] in equation (12).

$$[Err] = \begin{bmatrix} 1 & -d\phi_z & d\phi_y & dx \\ d\phi_z & 1 & -d\phi_x & dy \\ -d\phi_y & d\phi_x & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (12)$$

Equation (12) may be re-written from the matrix form into a vector form as shown in equation (13). A vector, $\underline{v}$, which is transformed by the matrix, [Err], is equivalent to the same vector affected by a displacement, $\underline{v}_{err}$, and rotation about ϕ_{err} with $\underline{v}_{err}$ and ϕ_{err} defined in equations (14).

8

$$[Err]\underline{v} = [I]\underline{v} + \underline{v}_{err} + \underline{\phi}_{err} \times \underline{v} \tag{13}$$

$$\underline{v}_{err} = \begin{bmatrix} dx \\ dy \\ dz \\ 1 \end{bmatrix} \quad ; \quad \underline{\phi}_{err} = \begin{bmatrix} d\phi_x \\ d\phi_y \\ d\phi_z \\ 0 \end{bmatrix} \tag{14}$$

An examination of the matrix form in equation (10) and the vector form in equation (13) reveals that as the error displacements and rotations vanish [Err] becomes the identity matrix and figure (2) becomes equivalent to figure (1).

A six degree-of-freedom error function, $\underline{f6}_{p\text{-err}}$, may be formed by concatenating $\underline{v}_{err}$ and $\underline{\phi}_{err}$. Since control of the torch motion will be related to the path, [P(i)], defined in the part frame, $\underline{f6}_{p\text{-err}}$ is shown in equation (15) transformed to the part reference frame by [Act], the homogeneous transform of the actual torch position. Matrix [Q] in equation (16) removes the 4th, scalar component, from the normalized homogeneous vector reducing it to a 3D vector. The resulting 6D vector is given in equation (17). The independent variable is an eight-dimensional vector, $\underline{\theta}$, in equation (18). The components of $\underline{\theta}$ are defined in terms of the positioner and robot angles in equation (19). This definition is consistent with the definition of the robot/positioner combination as a single serial kinematic chain defined from the part to the end-effector.

$$\underline{f6}_{p\text{-err}} = \begin{bmatrix} [Q][Act]\underline{v}_{err} \\ [Q][Act]\underline{\phi}_{err} \end{bmatrix} \tag{15}$$

where,

$$[Q] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{16}$$

9

$$\underline{f6}_{p-err} = [dx,dy,dz,d\phi_x,d\phi_y,d\phi_z]^T = \underline{f6}_{p-err}(\underline{\theta}) \tag{17}$$

$$\underline{\theta} = [\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6,\theta_7,\theta_8]^T \tag{18}$$

$$\underline{\theta} = [\theta_{2P},\theta_{1P},\theta_{1R},\theta_{2R},\theta_{3R},\theta_{4R},\theta_{5R},\theta_{6R}]^T \tag{19}$$

The vector error function has the property that the norm of $\underline{f6}_{p-err}$ vanishes as the torch position and orientation errors vanish (equation 20). This condition is also expressed in equation (21) when the function approaches the vector, $\underline{0}$, a vector having six zero components.

$$[(\underline{f6}_{p-err})^T][\underline{f6}_{p-err}] \to 0, \text{ as } [Err] \to [I] \tag{20}$$

$$\underline{f6}_{p-err}(\underline{\theta}) \to \underline{0} \qquad (\text{where } \underline{0} = [0,0,0,0,0,0]^T) \tag{21}$$

Newton's method has been used by Whitney [13] to solve equation (21) for $\underline{\theta}$. The iterative equation implementing the Newton's method is shown in equation (22). The predicted solution vector, $\underline{\theta}_{j+1}$, is equal to the previous solution vector, $\underline{\theta}_j$, modified by a correction term as shown. One factor in the correction term of equation (22) is the Jacobian matrix which is defined in equation (23). (Equation (A-3) of Appendix A defines the Jacobian matrix in terms of the robot and positioner transformation matrices.) Since the Jacobian matrix, J, has eight columns corresponding to the eight joint angles of our combined mechanism and J has only six rows for our differential translations and rotations, J is non-square and the pseudo-inverse of J must be used (see equation 24). The scalar factor, h, in equation (24) is the iteration step size. The use of the pseudo-inverse results in a solution of the under-determined system of equations for the eight component angles of $\underline{\theta}$ that satisfy the torch-to-part position requirement and minimize the sum of the squares of the eight joint displacements. A derivation of equation (24) is presented in Appendix B. In solving for $\underline{\theta}$, equation (24) is repeated until the weighted vector norm in equation (25) is

satisfied. The elements of the diagonal weighting matrix, K, and epsilon may be chosen to increase or decrease the tolerance to position and orientation errors in the final solution.

$$\underline{\theta}_{j+1} = \underline{\theta}_j - ([\frac{\partial}{\partial \underline{\theta}} \underline{f6}_{p-err}(\underline{\theta}_j)]^{-1})\underline{f6}_{p-err}(\underline{\theta}_j) \tag{22}$$

$$[J] = [\frac{\partial}{\partial \underline{\theta}} \underline{f6}_{p-err}(\underline{\theta}_j)] \tag{23}$$

$$\underline{\theta}_{j+1} = \underline{\theta}_j + h*[J^T(JJ^T)^{-1}]\underline{f6}_{p-err}(\underline{\theta}_j) \tag{24}$$

$$[\underline{f6}_{p-err}(\underline{\theta})^T][K][\underline{f6}_{p-err}(\underline{\theta})] < \text{epsilon} \tag{25}$$

ALGORITHM II: POSITION CONTROL WITH DOWNHAND TORCH ALIGNMENT

In this section we will derive the control needed to maintain the proper weld orientation. To accomplish this we will first examine the robot's path program to determine the desired local vertical in the part frame, and secondly we will develop additional constraint equations to augment the forcing function defined in equation (15) achieving the correct torch-to-part position while maintaining proper part orientation for downhand welding.

The robot's path was defined above as a series of homogeneous transformations, where [P(i)] denotes the desired position of the torch defined in part coordinates for the (i)th program step. The first three columns of [P(i)] are the basis vectors of a reference frame tied to the torch and the fourth vector is the desired position of the torch frame's origin (see equation 26).

$$[P(i)] = [\underline{n}(i)\ \underline{o}(i)\ \underline{a}(i)\ \underline{p}(i)] \tag{26}$$

Figure (3) depicts a segment of the path sequence defined by [P(i)] in the part reference frame. In figure (3) $\underline{a}(i)$ and $\underline{p}(i)$ are the third and fourth columns of [P(i)]. The program points have been chosen so that the secant, $\underline{s}(i)$, between adjacent points, $\underline{p}(i-1)$

and $\underline{p}(i)$, approximates the weld path tangent with sufficient accuracy. A unit vector, $\underline{n}_s(i)$, along the vector $\underline{s}(i)$ is defined in equation (27). In a majority of weld programs the vector, $\underline{a}(i)$, directed along the torch plasma flow, the path secant, $\underline{s}(i)$, and the desired gravity gradient, $\underline{n}_g(i)$, will be co-planar. Additionally, $\underline{n}_g(i)$ will be perpendicular to the path tangent which is approximated in this case by the secant vector, $\underline{s}(i)$. Figure (4) depicts this relationship and equation (28) defines the vector $\underline{n}_g(i)$ which is to be aligned with the gravity gradient vector in the workcell. $\underline{n}_s(i)$ is the unit vector along $\underline{s}(i)$.

$$\underline{n}_s(i) \ = \ \frac{\underline{p}(i) \ - \ \underline{p}(i-1)}{\| \ \underline{p}(i) \ - \ \underline{p}(i-1) \ \|} \tag{27}$$

$$\underline{n}_g(i) \ = \ \frac{\underline{a}(i) - [\underline{n}_s(i) \cdot \underline{a}(i)] \underline{n}_s(i)}{\| \ \underline{a}(i) - [\underline{n}_s(i) \cdot \underline{a}(i)] \underline{n}_s(i) \ \|} \tag{28}$$

Downhand position control is achieved when the part is positioned so that $\underline{n}_g(i)$ and the Earth's gravity gradient, $\underline{g}$, are aligned at the (i)th program point. A rotation vector (equation 29) equal to the cross product of $\underline{n}_g$ and $\underline{g}$ is used as a rotational forcing function that will tend to align $\underline{n}_g$ with the local gravity gradient, $\underline{g}$. We note that as in the case of the positional forcing function defined in equation (15), this new gravity alignment function vanishes as the error in alignment diminishes (equation 30).

$$\underline{f}_g \ = \ \underline{n}_g(i) \ \times \ \underline{g} \tag{29}$$

$$\| \ \underline{f}_g \ \| \ -> \ 0 \ as \ \underline{n}_g \ -> \ \underline{n}_g \ \| \ \underline{g} \tag{30}$$

The coordinate frame for our robot's workcell in this study was chosen with the x axis vertical, resulting in the gravity gradient, $\underline{g}$, being equal to $(-1 \ 0 \ 0 \ 0)^T$. Since $\underline{g}$ is fixed with respect to workcell coordinates the vector forcing function defined in equation (29) will lie in the y-z or horizontal plane, therefore the vertical component of the vector forcing

function may be eliminated reducing the forcing function to a 2D vector as shown in equation (31) with [R] defined in equation (32).

$$\underline{f2}_{g\text{-err}} = [R](([Z_p][A_{1p}][A_{2p}][G]\underline{n}_g(i)) \times \underline{g}) \qquad (31)$$

where,

$$[R] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (32)$$

The 2D vector error function, $\underline{f2}_{g\text{-err}}$, also has the property that its norm vanishes as $\underline{n}_g(i)$ and $\underline{g}$ are aligned as shown in equation (33).

$$[(\underline{f2}_{g\text{-err}})^T][\underline{f2}_{g\text{-err}}] \rightarrow 0 \quad \text{as } \underline{n}_g(i) \times \underline{g} \rightarrow 0 \qquad (33)$$

A combined error (equation 34) function, $\underline{f8}_{t\text{-err}}$, is formed by augmenting the two previously defined error functions. This function approaches the zero vector, $\underline{0}$, as all constraints are met (equation 35).

$$\underline{f8}_{t\text{-err}} = \begin{bmatrix} \underline{f6}_{p\text{-err}} \\ \underline{f2}_{g\text{-err}} \end{bmatrix} \qquad (34)$$

$$\underline{f8}_{t\text{-err}} \rightarrow \underline{0} \qquad \text{(where } \underline{0} = [0,0,0,0,0,0,0,0]^T \qquad (35)$$

Once again Newton's method is used to iteratively solve equation (35) for the eight joint angles required to control torch position and weld seam orientation. This algorithm is given by equation (36). The evaluation of this equation is repeated until the error measurement in equation (37) falls below the acceptable tolerance, epsilon. Elements in the diagonal matrix [K] are chosen to vary the relative tolerance on torch to weld seam positioning accuracy and downhand orientation of the part. If these diagonal elements are

13

denoted by k_{ii}, then: torch-to-seam position tolerance is affected by k_{ii} for $i=1,2,3$; torch-to-seam orientation tolerance is specified in k terms 4,5 and 6; while terms 7 and 8 control downhand part orientation tolerance.

$$\underline{\Theta}_{i+1} = \underline{\Theta}_i + h*[J^{-1}]\underline{f8}_{t-err} \qquad (36)$$

$$[(\underline{f8}_{t-err})^T][K][\underline{f8}_{t-err}] < epsilon \qquad (37)$$

The derivation of the Jacobian matrix, [J], which is needed for the implementation of the Newton's method specified in equation (36) is found in Appendix A (equation A-23) while the inverse of this Jacobian is given in Appendix C.

ALGORITHM III: WIRE FEED

This section develops the control needed to adjust the wire feed mechanism so that wire is fed in front of the moving torch. In our work to this point we specified the torch's position and orientation with-respect-to the part. This specification established displacements in x, y, z and orientations of pitch yaw and roll requiring six degrees-of-freedom (DOF). The additional constraint equations developed to maintain the downhand orientation required two more DOF's. Since our total mechanism included the two DOF positioner and six DOF robot, our system is totally determined. If further constraints on torch motion are needed we must either add an additional DOF's e.g., an auxiliary joint, or we may relax a previously imposed constraint. The later approach is used here. Positioning of the torch within the weld seam requires that control be maintained in only five dimensions: x, y, z, pitch and yaw about the torch tip. The sixth DOF, roll about the torch axis, does not affect the welding process and will be used to control wire feed orientation.

Wire feed control is necessary to maintain the position of the weld wire's entry into the plasma from the front of the torch as it moves along the seam. The method used is

14

similar to that of the previous section. We will define a constraint equation that will force our solution to include proper wire feed orientation. To begin our definition of this forcing function, $fl_{w\text{-err}}$, we must expand the definitions of the path program vectors presented in figure (3). Referring to figure (5) we observe a close-up detail of the path. As in the previous definitions, vectors $s(i)$ and $s(i+1)$, are path secant vectors with the corresponding unit vectors $n_s(i)$ and $n_s(i+1)$, respectively. The vector $n_g(i)$ is the desired vertical vector used to correct for downhand orientation at the (i)th program point. Also shown in figure (5) is $n_w(i)$, a unit vector along the direction of the wirefeed and pointing counter to the wire feed direction. This vector is fixed with respect to the torch and the end-effector reference frame, but it is designated $n_w(i)$ in figure (5) since it varies within the part frame and is associated with the (i)th path-program point. The end-effector's approach vector, $a(i)$, is shown co-linear to the axis of rotation of the robot's last joint, θ_{6R}. This relationship is necessary to insure that a rotation of θ_{6R} will affect wire-feed orientation without displacing the point at which welding occurs.

The algorithm proceeds by first solving the inverse kinematic equations iteratively using equations (36) and (37) resulting in an arm position that achieves welding in the downhand position. The robot's last joint angle is then modified to achieve proper wire feed orientation. Referring to figure (5), proper wire feed orientation occurs when $n_w(i)$, $n_s(i)$ and $a(i)$ are co-planar and the angle formed by the positive directed segments of $n_w(i)$ and $a(i)$ includes $n_s(i)$.

A scalar forcing function, $fl_{w\text{-err}}$, defined as in equation (38) vanishes when the optimum wirefeed position is achieved. The sign of this function is such that it may be used iteratively to drive θ_{6R} to the optimum position when the magnitude of θ_{6R} is defined by the right-hand-rule about $a(i)$.

$$fl_{w\text{-err}} = [n_w(i) \times n_s(i)] \cdot a(i) \qquad (38)$$

This function may be re-written in terms of the kinematic matrices as shown in equation (39).

$$f1_{w-err} = [([G^{-1}][T_P(i)^{-1}][Z_P^{-1}][Z_R][T_R(i)]E\underline{n}_w) \times \underline{n}_s(i)] \cdot \underline{a}(i)$$

(39)

In evaluating equation (39) the transformation angles which include the two positioner angles, θ_{1P} and θ_{2P}, and the first five robot angles, θ_{1R} to θ_{5R}, are fixed at the result derived from the solution of equation (36) and (37) in the previous section on downhand control. The robot's last degree-of-freedom, θ_{6R}, is derived by repetitive evaluation of equation (40) until the condition in equation (41) is satisfied. The constant, h, in equation (40) adjusts the iteration step size.

$$\theta_{6R}(j+1) = \theta_{6R}(j) + h*fl_{w-err}$$

(40)

$$k * (fl_{w-err})^2 < epsilon$$

(41)

SIMULTANEOUS TOTAL SOLUTION

An alternative to the separate solution of wire feed orientation and downhand path control is a method using a concurrent form of the iterative equations. The concurrent solution may be written by combining the equations (36), (37), (40) and (41) into a single pair of iterative and constraint equations. A generalized nine dimensional vector forcing function, $\underline{f9}_{t-err}$, is defined in equation (42) as the concatenation of our $\underline{f6}_{p-err}$, $\underline{f2}_{g-err}$ and fl_{w-err} functions.

$$\underline{f9}_{t-err} = \begin{bmatrix} \underline{f6}_{p-err} \\ \underline{f2}_{g-err} \\ fl_{w-err} \end{bmatrix}$$

(42)

Since the solution of equation (36) was based on small angle approximations and the angular motions required to maintain proper wirefeed may be quite large, the two sets of iterative equations (36) and (40) must be de-coupled by defining an artificial state variable, θ_{6R}'. Angle θ_{6R}' is the angle required to maintain wirefeed control assuming the other constraints of position and downhand were met. The resulting iterative equation given in (43) and is evaluated subject to equation (44) being met.

$$\underline{\theta}_{j+1} = \underline{\theta}_j + h * \begin{bmatrix} J^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{f\theta}_{t-err}(\underline{\theta}) \end{bmatrix} \qquad (43)$$

$$[\underline{f\theta}_{t-err}(\underline{\theta})^T][K'][\underline{f\theta}_{t-err}(\underline{\theta})] < epsilon \qquad (44)$$

$$\underline{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9] \qquad (45)$$

$$\underline{\theta} = [\theta_{2P}, \theta_{1P}, \theta_{1R}, \theta_{2R}, \theta_{3R}, \theta_{4R}, \theta_{5R}, \theta_{6R}, \theta_{6R}'] \qquad (46)$$

The vector, $\underline{\theta}$, in equations (43) through (46) has nine components, θ_1 through θ_8, as defined previously and θ_9 is equal to the artificial state variable, θ_{6R}'. Matrix $[J]$ is defined in equation (A-23). The vector function $\underline{f\theta}_{t-err}$ is from equation (42), and finally, the matrix $[K']$ is a one dimensional extension of $[K]$, the 8x8 matrix from equation (37). The added diagonal term K_{99} controls the tolerance of the wirefeed orientation error. The results of a computer graphic simulation of this algorithm are now presented.

GRAPHIC SIMULATION RESULTS

The workcell layout, the six DOF robot, welding torch, the two DOF part positioner and the part were modeled using ROBOSIM [12], a NASA and Vanderbilt University developed computer graphic modeling package. Each mechanism is described using a procedure oriented instruction set. The completed model may be displayed on a variety of graphic displays including Tektronix 4014, a CAD system via IGES (Initial Graphics Exchange Standard), Evans & Sutherland PS330, GTI Poly 2000 or Silicon Graphics IRIS

(other interfaces are planned). Figure (6) shows the workcell containing the robot, positioner and part. A close-up detail of the part is shown in Figure (7). The part for this example is a corrugated plate with the programmed weld seam as shown in Figure (7).

Figures (8a-8h) display the robot at several points along the path without downhand control in effect (K_{77} and K_{88} equal to zero in equation (37) with fixed positioner angles, Θ_{1P} and Θ_{2P}). The assumed torch orientation is perpendicular to the surface tangent. Figures (9a-9h) show the same path points with downhand control (algorithm II) enabled. In Figures (9a-9h) we observe that the tangent to the weld path at the torch tip is horizontal, i.e., welding is occurring in the downhand position. An appreciation for the complex motion that must be performed by the torch can be gained by examining figure (10). Figure (10) depicts the torch trajectory with respect to the workcell reference frame during the downhand welding operation. This path bears little resemblance to the sinusoidal motion that would be observed in the part's reference frame. The velocity profile of the torch tip also differs considerably when observed from the two reference frames. Figure (11) compares the torch's velocity, observed in the workcell's reference frame (dashed line), to the constant torch velocity observed in the part's frame (solid line). We observe that the torch velocity must vary widely in workcell coordinates in order to maintain constant (.10) units/sec seam velocity.

Since algorithm III controls wire feed in addition to controlling the orientation of the part for downhand welding, a new part will be used to demonstrate this added feature. Figure (12) depicts this new part which is similar to the old part with a path that weaves in a sinusoidal manner as it travels over the surface. Algorithm III is demonstrated in Figures (13a-13h) which show an animation of the robot's motion from the side at each time point. The front view clearly illustrates the control of downhand orientation while the corresponding top views in Figures (14a-14h) allow us to observe that the weld wire's orientation is controlled. For clarity, the wire frame views in figures 14a through 14h have

18

been modified by rendering invisible, details that would obscure. Removed are the cell floor; positioner links zero and one; and robot links four and five. In the first frame of the animation sequence (14a) the weld wire orientation is not controlled since control is based on an approximation of the path tangent using a backward difference (equation 39). Subsequent frames, however, clearly show that the weld wire's orientation is maintained.

HARDWARE IMPLEMENTATION

For those applications employing a CAD/CAM system to develop the path programs for the robotic arc welding cell, the transformation algorithm derived in this paper may be appropriately resident in the CAD/CAM system. It is only necessary for the weld assembly itself to be in the CAD/CAM display. The user specifies the position and orientation of the welding torch at each of the program points. The position and orientation is specified relative to the assembly. The kinematic transformation matrix, positioner axis commands, and velocity commands are downloaded from the CAD/CAM system to the robotic controller.

For those applications employing a teach pendant to develop the path programs for the robot, the transformation algorithm would be resident in another computer. For each programmed point, the kinematic transformation matrix and the positioner axes values would be uploaded to this computer. The output of the transformation algorithm would then be downloaded to the robotic controller for welding.

DISCUSSION AND CONCLUSIONS

A transformation algorithm has been presented that permits the development of path programs for arc welding applications without the necessity of manually positioning the assembly to the downhand position or maintaining wire feed orientation for each point to be programmed. The algorithm automatically generates the required kinematic transformation algorithms, positioner axes command values and velocities to insure

19

downhand welding at constant weld travel speed with wire feed orientation controlled. The algorithm is compatible with CAD/CAM development of the path programs or with development of path programs via a teach pendant. As of this writing, the algorithm is being used to develop downhand adjusted path programs for a robot used in Space Shuttle Main Engine (SSME) weld process development [6-8] at the Marshall Space Flight Center. It is estimated that complex weld contours can be programmed in 75% less time by use of the transformation algorithm. On assemblies, such as the SSME, which contains over 2000 welds. This represents a very substantial savings in costs and time.

Since use of the transformation algorithm in developing path programs does not require that the weld assembly be positioned to the downhand position during the programming stage, it does not guarantee that the down position will be achievable for all welds programmed. It is hence important that this be established prior to welding. This may be done by means of a no-weld dryrun. However, a less time consuming method to check the path programs prior to welding is via computer simulation.

APPENDIX A: FORMATION OF THE JACOBIAN MATRIX

Use of the Jacobian in manipulator control was first performed by Whitney [13] in his specification of resolved motion rate control (RMRC). RMRC has been applied to numerous man-in-the-loop manipulator control systems including the Space Shuttle's Remote Manipulator System (RMS). The Jacobian is a matrix which relates joint angle rates to torch displacement and Euler rotation rates. The definition of the j-th column of the Jacobian matrix is given in equation (A-1) below.

$$
[J] \;=\;
\begin{bmatrix}
\ldots \dfrac{\partial x}{\partial \theta_j} \ldots \\[2mm]
\ldots \dfrac{\partial y}{\partial \theta_j} \ldots \\[2mm]
\ldots \dfrac{\partial z}{\partial \theta_j} \ldots \\[2mm]
\ldots \dfrac{\partial \phi_x}{\partial \theta_j} \ldots \\[2mm]
\ldots \dfrac{\partial \phi_y}{\partial \theta_j} \ldots \\[2mm]
\ldots \dfrac{\partial \phi_z}{\partial \theta_j} \ldots
\end{bmatrix}
$$

element

differential of

```
..∂x ..        torch-x in part frame
..∂θⱼ..        w.r.t. the j-th joint

..∂y ..        torch-y in part frame
..∂θⱼ..        w.r.t. the j-th joint

..∂z ..        torch-z in part frame
..∂θⱼ..        w.r.t. the j-th joint

..∂φₓ..        torch-x-rot. in part frame
..∂θⱼ..        w.r.t. the j-th joint

..∂φᵧ..        torch-y-rot. in part frame
..∂θⱼ..        w.r.t. the j-th joint

..∂φ𝓏..        torch-z-rot. in part frame
..∂θⱼ..        w.r.t. the j-th joint
```

(A-1)

```
Joint angle definitions:
    for j=1, θⱼ equals positioner angle second from base
    for j=2, θⱼ equals positioner angle next to base
    for j=3, θⱼ equals robot angle next to base
    for j=4, θⱼ equals robot angle second from base
    for j=5, θⱼ equals robot angle third from base
    for j=6, θⱼ equals robot angle fourth from base
    for j=7, θⱼ equals robot angle fifth from base
    for j=8, θⱼ equals robot angle sixth from base
```

The physical interpretation of the Jacobian can be shown with the aid of figure (A-1). A wheel with axis of rotation $\underline{w}$ and a point (R) on its circumference with the vector $\underline{r}$ from the axis of rotation to the point (R). Both vector $\underline{w}$ and $\underline{r}$ are 3D vectors for this

21

example rather than the 4D homogeneous vectors used elsewhere in this paper. The instantaneous displacement of (R) due to a differential rotation of the wheel is given by the cross product of $\underline{w}$ and $\underline{r}$. The instantaneous rotation of a reference frame located at (R) and fixed to the wheel would be $\underline{w}$ since the wheel is a single rigid body and as such all points in that body rotate about a fixed axis by definition. The Jacobian of this simple single degree-of-freedom mechanism is given in equation (A-2).

$$[J] = \begin{bmatrix} \underline{w} \times \underline{r} \\ \underline{w} \end{bmatrix} \qquad (A-2)$$

The Jacobian can be similarly defined for the eight degree-of-freedom (DOF) mechanism formed by combining the two DOF positioner and the six DOF robot. Equations (A-3) contain this definition

$$[J] = \begin{bmatrix} \underline{v}_1 & \underline{v}_2 & \underline{v}_3 & \underline{v}_4 & \underline{v}_5 & \underline{v}_6 & \underline{v}_7 & \underline{v}_8 \\ \underline{w}_1 & \underline{w}_2 & \underline{w}_3 & \underline{w}_4 & \underline{w}_5 & \underline{w}_6 & \underline{w}_7 & \underline{w}_8 \end{bmatrix} \qquad (A-3)$$

The end-effector displacements in part frame given by equations (A-4) to (A-11):

$$\underline{v}_1 = -[Q]([G^{-1}][A_{2p}^{-1}](\underline{w} \times ([A_{1p}^{-1}][Z_p^{-1}]Z_r T_r(i) E\underline{r}))) \qquad (A-4)$$

$$\underline{v}_2 = -[Q]([G^{-1}][A_{2p}^{-1}][A_{1p}^{-1}](\underline{w} \times ([Z_p^{-1}]Z_r T_r(i) E\underline{r}))) \qquad (A-5)$$

$$\underline{v}_3 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r(\underline{w} \times (T_r(i) E\underline{r}))) \qquad (A-6)$$

$$\underline{v}_4 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r}(\underline{w} \times (A_{2r}A_{3r}A_{4r}A_{5r}A_{6r} E\underline{r}))) \qquad (A-7)$$

$$\underline{v}_5 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r}(\underline{w} \times (A_{3r} A_{4r} A_{5r} A_{6r} E\underline{r})))$$

(A-8)

$$\underline{v}_6 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r}(\underline{w} \times (A_{4r} A_{5r} A_{6r} E\underline{r})))$$

(A-9)

$$\underline{v}_7 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r} A_{4r}(\underline{w} \times (A_{5r} A_{6r} E\underline{r})))$$

(A-10)

$$\underline{v}_8 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r} A_{4r} A_{5r}(\underline{w} \times (A_{6r} E\underline{r})))$$

(A-11)

The end-effector rotations defined in part frame given by:

$$\underline{w}_1 = -[Q]([G^{-1}][A_{2p}^{-1}]\underline{w})$$

(A-12)

$$\underline{w}_2 = -[Q]([G^{-1}][A_{2p}^{-1}][A_{1p}^{-1}]\underline{w})$$

(A-13)

$$\underline{w}_3 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r\underline{w})$$

(A-14)

$$\underline{w}_4 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r}\underline{w})$$

(A-15)

$$\underline{w}_5 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r}\underline{w})$$

(A-16)

$$\underline{w}_6 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r}\underline{w})$$

(A-17)

$$\underline{w}_7 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r} A_{4r}\underline{w})$$

(A-18)

$$\underline{w}_8 = [Q]([G^{-1}][T_p(i)^{-1}][Z_p^{-1}]Z_r A_{1r} A_{2r} A_{3r} A_{4r} A_{5r}\underline{w})$$

(A-19)

$$\underline{r} = (0\ 0\ 0\ 1)^T$$

(A-20)

$$\underline{w} = (0\ 0\ 1\ 0)^T$$

(A-21)

$$[Q] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(A-22)

The Jacobian matrix defined in equation (A-3) has six rows and eight columns. The system of equations represented is under-determined and if equation (A-3) is to be used to control torch to part position and orientation, two additional constraints must be added. As stated in the text these additional constraints will require the part to be oriented so that welding occurs in the downhand position. These added constraints can be met by rotation of the part about a vector lying in the workcell's horizontal plane. Equations (A-23) to (A-27) describe a Jacobian augmented with additional rotational vectors, $\underline{d}_1$ and $\underline{d}_2$. These terms relate part rotations defined in workcell coordinates to joint angle rotations. The vectors $\underline{d}_1$ and $\underline{d}_2$ have been reduced to 2D vectors by the [R] operator which removed the x-component and the scale factor (fourth component). This was done since the x-component is vertical in the workcell reference frame definition in the example used. We further note that only columns one and two have non-zero $\underline{d}$ entries since only the two positioner angles are effective in orienting the part with respect to the gravity field.

$$[J] \;=\; \left[\begin{array}{cccccccc} \underline{v}_1 & \underline{v}_2 & \underline{v}_3 & \underline{v}_4 & \underline{v}_5 & \underline{v}_6 & \underline{v}_7 & \underline{v}_8 \\ \underline{w}_1 & \underline{w}_2 & \underline{w}_3 & \underline{w}_4 & \underline{w}_5 & \underline{w}_6 & \underline{w}_7 & \underline{w}_8 \\ \hline \underline{d}_1 & \underline{d}_2 & \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} \end{array} \right] \tag{A-23}$$

Where $\underline{d}_1$ and $\underline{d}_2$ are given by:

$$\underline{d}_1 \;=\; [R][Z_p][A_{1p}]\underline{w} \tag{A-24}$$

$$\underline{d}_2 \;=\; [R][Z_p]\underline{w} \tag{A-25}$$

$$[R] \;=\; \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] \tag{A-26}$$

$$\underline{0} \;=\; [\; 0 \; 0 \;]^T \tag{A-27}$$

APPENDIX B: TORCH CONTROL WITH MINIMIZED JOINT DISPLACEMENTS

The six degree-of-freedom position error function, $\underline{f6}_{p\text{-err}}$, which was introduced in equation (15) vanishes as the error transformation of equation (12) becomes the identity matrix. This relationship is graphically depicted in figure (B-1). The ordinate of figure (B-1) is the norm squared of the function $\underline{f6}_{p\text{-err}}$ corresponding to equation (21) while the abscissa is a one-dimensional representation of the function's domain. The point, $\underline{\theta}^*$, represents the solution of joint angles needed to maintain proper torch-to-part position. The point $\underline{\theta}_j$ represents the current estimate of the joint solution, $\underline{\theta}^*$; and $\underline{\theta}_{j+1}$ represents the next estimate of the joint solution. The derivation of an iterative equation using Newton's Method begins with a truncated Taylor's series approximation of the function, $\underline{f6}_{p\text{-err}}$, as given in equation (B-1).

$$\underline{f6}_{p\text{-err}}(\underline{\theta} + \underline{\delta\theta}) = \underline{f6}_{p\text{-err}}(\underline{\theta}) + \left[\frac{\partial \underline{f6}_{p\text{-err}}}{\partial \underline{\theta}} \right] \underline{\delta\theta} + \text{H.O.T.}$$

$$(B-1)$$

The vector, $\underline{\delta\theta}$, represents the change in $\underline{\theta}$ between estimations, $\underline{\theta}_j$ and $\underline{\theta}_{j+1}$, as given in equation (B-2).

$$\underline{\delta\theta} = \underline{\theta}_{j+1} - \underline{\theta}_j \qquad (B-2)$$

Substituting (B-2) into (B-1) to solve for the $\underline{\theta}_{j+1}$ estimation we get equation (B-3).

$$\underline{\theta}_{j+1} = \underline{\theta}_j + \left[\frac{\partial \underline{f6}_{p\text{-err}}}{\partial \underline{\theta}} \right]^{-1} \left[\underline{f6}_{p\text{-err}}(\underline{\theta} + \underline{\delta\theta}) - \underline{f6}_{p\text{-err}}(\underline{\theta}) \right]$$

$$(B-3)$$

For our solution $\underline{f6}_{p\text{-err}}(\underline{\theta} + \underline{\delta\theta})$ is the zero vector, $\underline{0}$, and the definition of the Jacobian, [J], we may re-write this as equation (B-4).

$$\underline{\theta}_{j+1} = \underline{\theta}_j - [J^{-1}]\left[\underline{f6}_{p\text{-err}}(\underline{\theta}_j) \right] \qquad (B-4)$$

The matrix, [J], must be square to have an explicit inverse, equation (B-4) applies only for a six degree-of-freedom mechanism. In this case, however, the Jacobian has six rows and eight columns and cannot be inverted. Although [J] is non-square the relationship between [J] and variations in $\underline{\theta}$ and $\underline{f6}_{p\text{-err}}$ is still valid as expressed in equation (B-5).

$$\underline{\delta f6}_{p\text{-err}} = [J]\underline{\delta\theta} \qquad (B-5)$$

The system of equations defined in (B-5) is under-determined and an infinite number of solutions exist. One method [14] to overcome this problem is to impose the added constraint that the weighted sum-of-squares of joint angle displacements, $\underline{\delta\theta}$, will be minimized. This constraint measure is given in equation (B-6).

$$C = \tfrac{1}{2}(\underline{\delta\theta}^T W \underline{\delta\theta}) \qquad (B-6)$$

The matrix, W, is a diagonal weighting matrix to allow the the specification of factors which will increase the motion of preferred joint angles. A combined cost function is formed in (B-7) using a LaGrange multiplier, λ, to append equations (B-5) and (B-6).

$$S = \tfrac{1}{2}(\underline{\delta\theta}^T W \underline{\delta\theta}) - [\lambda^T]\left[[J][\underline{\delta\theta}] - \underline{f6}_{p\text{-err}} \right] \qquad (B-7)$$

The derivative of S with respect to $\underline{\delta\theta}$ will be zero at the minimum condition yielding equation (B-8).

$$\frac{\partial S}{\partial\underline{\theta}} = W\underline{\delta\theta} - J^T\lambda = 0 \qquad (B-8)$$

Solving (B-8) for $\underline{\delta\theta}$ we get equation (B-9).

$$\underline{\delta\theta} = [W^{-1}][J^T]\lambda \tag{B-9}$$

Multiplying (B-9) by [J] and substituting (B-5) into the result we get (B-10).

$$[J]\underline{\delta\theta} = [J][W^{-1}][J^T]\lambda = \underline{f6}_{p-err} \tag{B-10}$$

Solving (B-10) for the LaGrange multiplier, λ, and substituting into equation (B-8) we get equation (B-11).

$$\underline{\delta\theta} = [W^{-1}][J^T]\Big[[J][W^{-1}][J^T]\Big]^{-1}[\ \underline{f6}_{p-err}\ ] \tag{B-11}$$

Using our definition of $\underline{\delta\theta}$ and $\underline{\delta f6}_{p-err}$ as given in figure (B-1) and equation (B-3) and (B-12) we get the resulting iterative equation in (B-13).

$$\underline{\delta f6}_{p-err} = \underline{0} - \underline{f6}_{p-err} \tag{B-12}$$

$$\underline{\theta}_{j+1} = \underline{\theta}_j - \Big[W^{-1}J^T(JW^{-1}J^T)^{-1}\Big]\underline{f6}_{p-err} \tag{B-13}$$

$$\underline{f6}_{p-err}^T[K]\underline{f6}_{p-err} < epsilon \tag{B-14}$$

$$\underline{\theta}_{j+1} = \underline{\theta}_j - J^T(JJ^T)^{-1}\underline{f6}_{p-err} \tag{B-15}$$

Equation (B-13) is executed repeatedly until the weighted norm in equation (B-14) is less than the tolerance epsilon. For the case that equal weighing of joint displacement is desired joint angle weighting matrix, W, may be set to identity yielding equation (B-15).

APPENDIX C: COMPUTATION OF J^{-1}

The eight by eight Jacobian matrix defined in Appendix A, equation (A-23) has the sparse form shown in equation (C-1). The inverse of the Jacobian is given in equation (C-2). We note that the inverse of the eight by eight matrix may be written in terms of the inverses of a six by six and a two by two matrix a result that significantly reduces the computation time.

$$[J] = \begin{bmatrix} A & B \\ \hline C & 0 \end{bmatrix} \tag{C-1}$$

$$[J^{-1}] = \begin{bmatrix} 0 & [C^{-1}] \\ \hline [B^{-1}] & -[B^{-1}]A[C^{-1}] \end{bmatrix} \tag{C-2}$$

Where:

$$[A] = \begin{bmatrix} \underline{v}_1 & \underline{v}_2 \\ \underline{w}_1 & \underline{w}_2 \end{bmatrix} \tag{C-3}$$

$$[B] = \begin{bmatrix} \underline{v}_3 & \underline{v}_4 & \underline{v}_5 & \underline{v}_6 & \underline{v}_7 & \underline{v}_8 \\ \underline{w}_3 & \underline{w}_4 & \underline{w}_5 & \underline{w}_6 & \underline{w}_7 & \underline{w}_8 \end{bmatrix} \tag{C-4}$$

$$[C] = \begin{bmatrix} \underline{d}_1 & \underline{d}_2 \end{bmatrix} \tag{C-5}$$

The matrix, B, in equations (C-1) and (C-2) is similar to the Jacobian of the six DOF manipulator alone, while the two by two sub-matrix, C, is a sub-matrix within the positioner's Jacobian. The inverse of the eight by eight Jacobian (C-2) exists, if both the inverses of B and C exist. B^{-1} exists when the robot's Jacobian exists, and C^{-1} exists, for this example problem, when θ_{1P} is not zero.

REFERENCES

[1] Cook, George E., "Robotic Arc Welding: Research in Sensory Feedback Control," IEEE Trans. on Industrial Electronics, Vol. IE-30, No. 3, pp 252-268, August, 1983.

[2] Richardson, R.W., Farson, D.F., Jones, C.S. and Rogers, P.F., "A Vision Based Adaptive Welding System for Aerospace," Proc. of 1985 American Welding Society Conv., pp 110-111, April 29-May 3, 1985.

[3] Welding Handbook, Vol. 1, "Fundamentals of Welding," 7th Edition, Editor: Charlotte Weisman, 1976, American Welding Society.

[4] Cary, Howard B., Modern Welding Technology, 1979, Prentice-Hall, Inc.

[5] Metals Handbook, Vol. 6, "Welding, Brazing, and Soldering," 9th Edition, Editor: Joseph R. Davis, 1983, American Society for Metals.

[6] Fernandez, K.R. and Cook G.E., "Automatic Control of Downhand Position in Robotic Welding Operations," Proceedings of the IEEE Industry Applications Society Conference, IASCON85, Toronto, Ontario, Canada, October 1985, pp. 1634-1644.

[7] Fernandez, K.R. and Cook, G.E.,"Computer Graphic Simulation of an Algorithm for Controlling Downhand Position in Robotic Welding Operations," Proceedings of the SME Conference on Robotic Solutions in Aerospace Manufacturing, Orlando, FL, March 3-6, 1986, 20 pages.

[8] Fernandez, K.R. and Cook, G.E.,"Use of Computer Graphic Simulation Techniques for Robot Control System Development," Proceedings of the IEEE 18th Southeastern Symposium on System Theory, Knoxville, Tennessee, April 7-8, 1986, pp.433-441.

[9]   Duda, R.O. and Hart, P.E., <u>Pattern Classification and Scene Analysis</u>, 1973, John Wiley & Sons, Inc.

[10]   Paul, R.P., <u>Robot Manipulators: Mathematics, Programming, and Control</u>, 1981, The MIT Press.

[11]   Denavit, J., and Hartenberg, R.S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," <u>Journal of Applied Mechanics</u>, ASME Transactions, June 1955, pp 215-221.

[12]   <u>Exploiting Robots in Arc Welded Fabrication</u>, The Welding Institute, Cambridge, England, 1984, Chapter, "Robotic Arc Welding of Pipe and Piping Components," Cook, G.E., Fernandez, K.R., and Levik, P., 14 pages.

[13]   Whitney, D.E.,"Resolved Motion Rate Control of Manipulators and Human Prostheses," <u>IEEE Transactions on Man-Machine Systems</u>, Vol. MMS-10, No. 2, June 1969, pp.47-53.

[14]   Starr, G. and Salisbury, K., "Remote Manipulator Performance Measures and Display Design," Final Report NASA Grant NGR-05-020-345, Design Division, Mechanical Engineering Department, Stanford University, Stanford, CA, February 1976.

$T_P(i)$

G

z_P

O

z_R

$P(i)$

E

$T_R(i)$
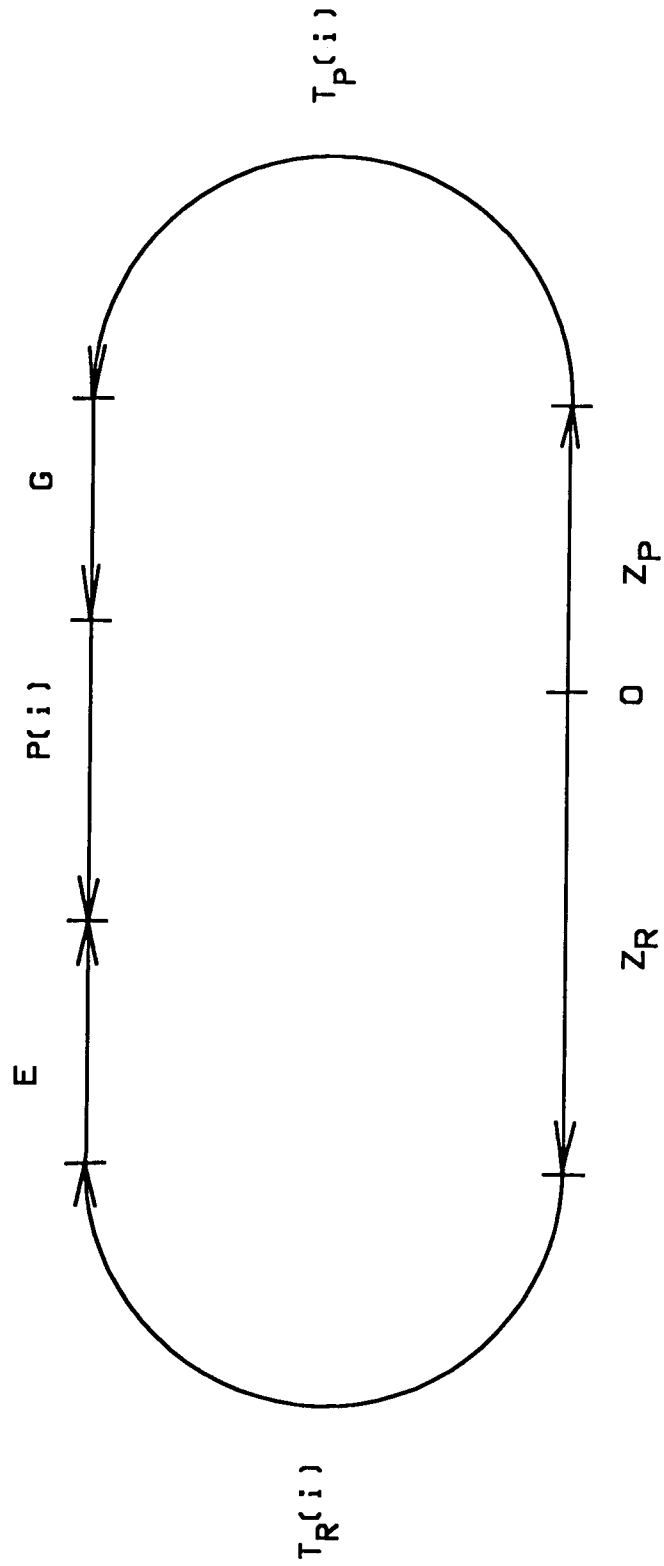
Figure 1.   Workcell transform graph.
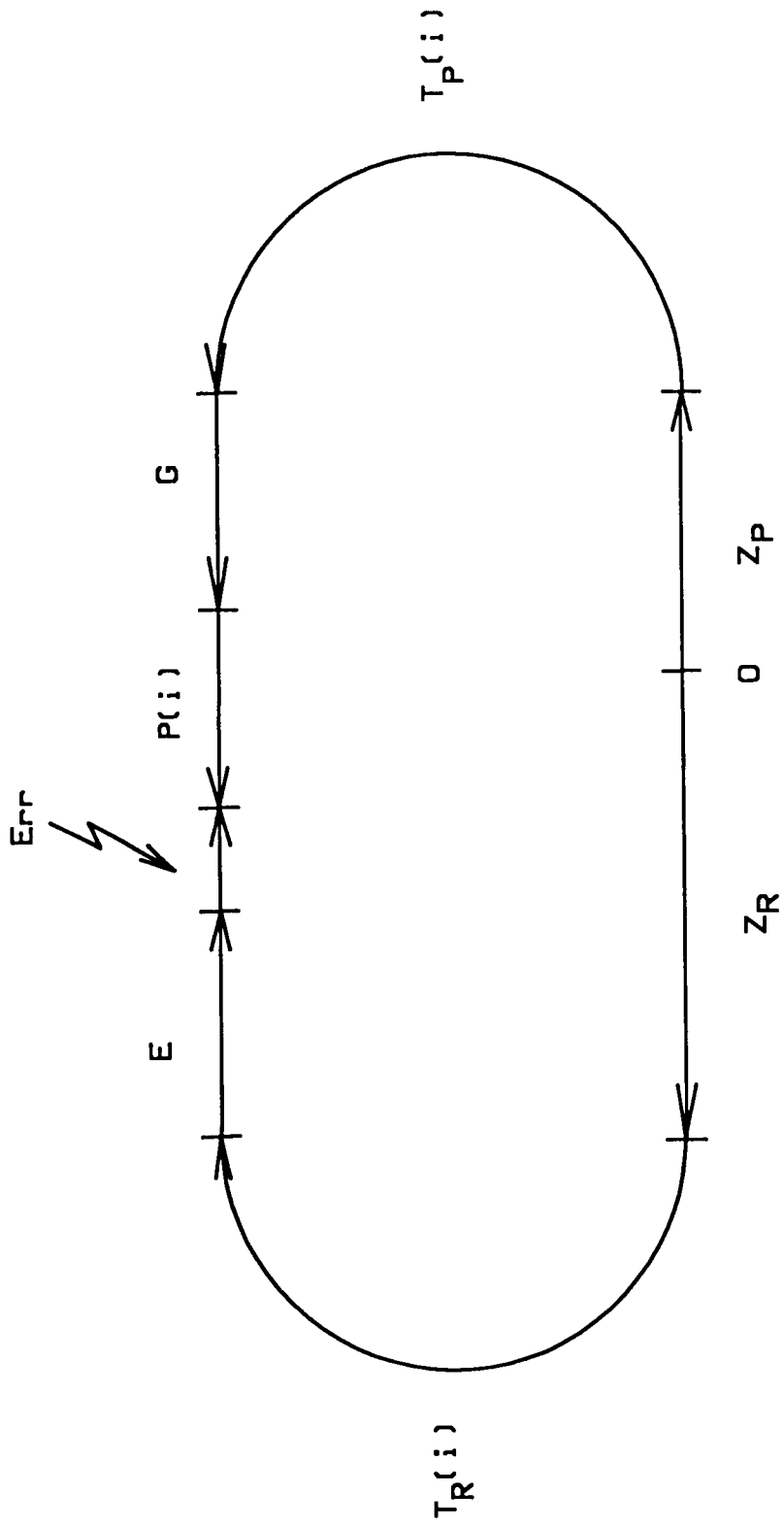
Figure 2.   Transform graph with torch position error.
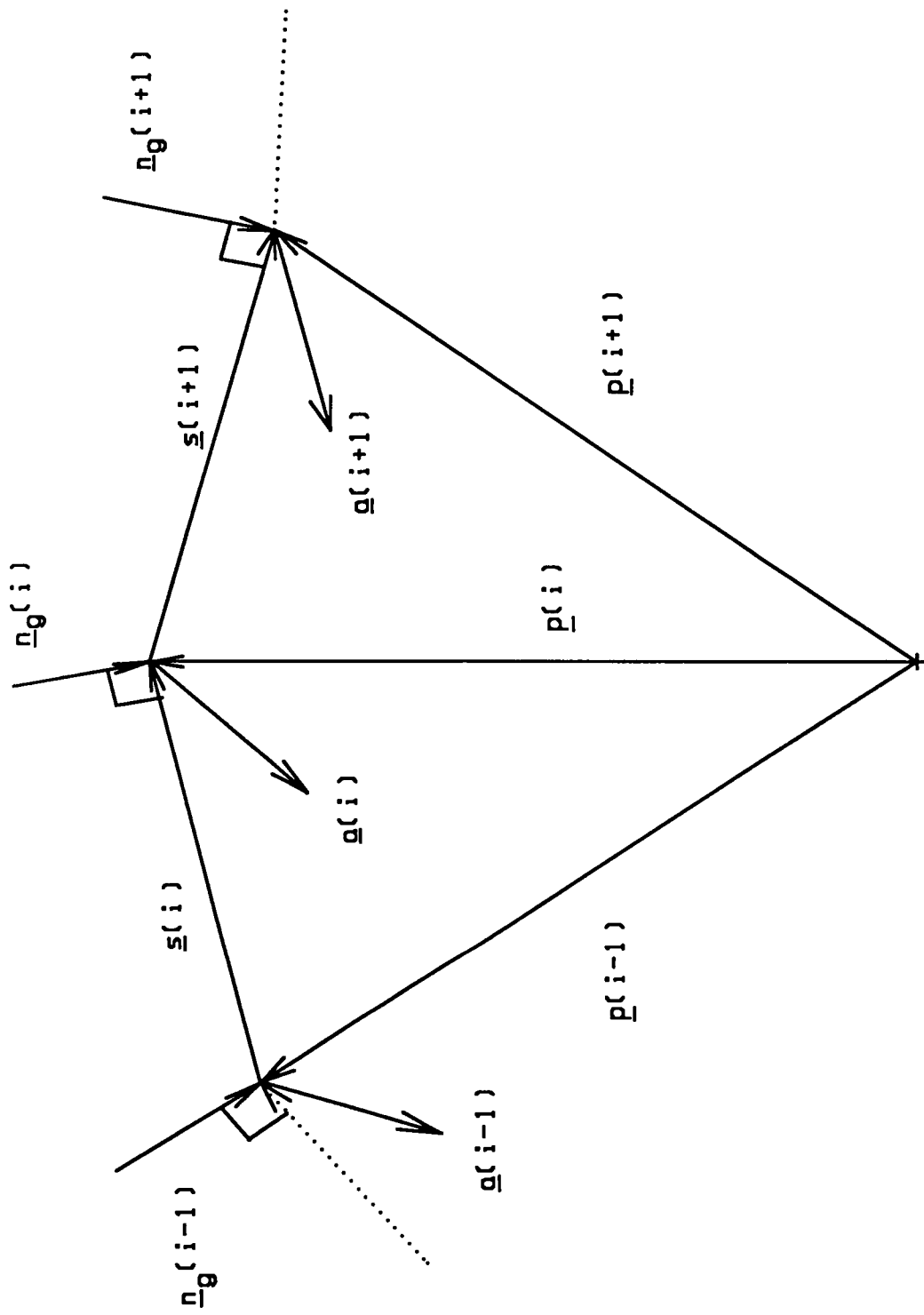
Figure 3.   Robot path in part frame.

$$\underline{q}(i) - [\underline{n}_s(i)\cdot\underline{q}(i)]\ \underline{n}_s(i)$$

$\underline{n}_g(i)$

$\underline{n}_s(i)$

$\underline{q}(i)$
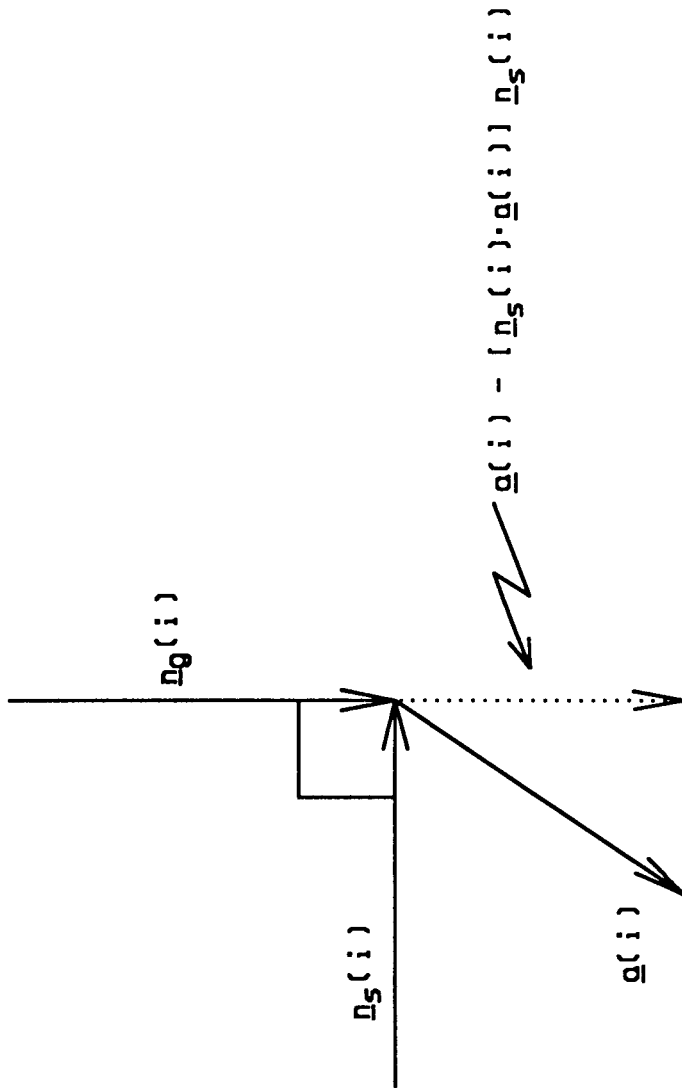
Figure 4. Path Point.
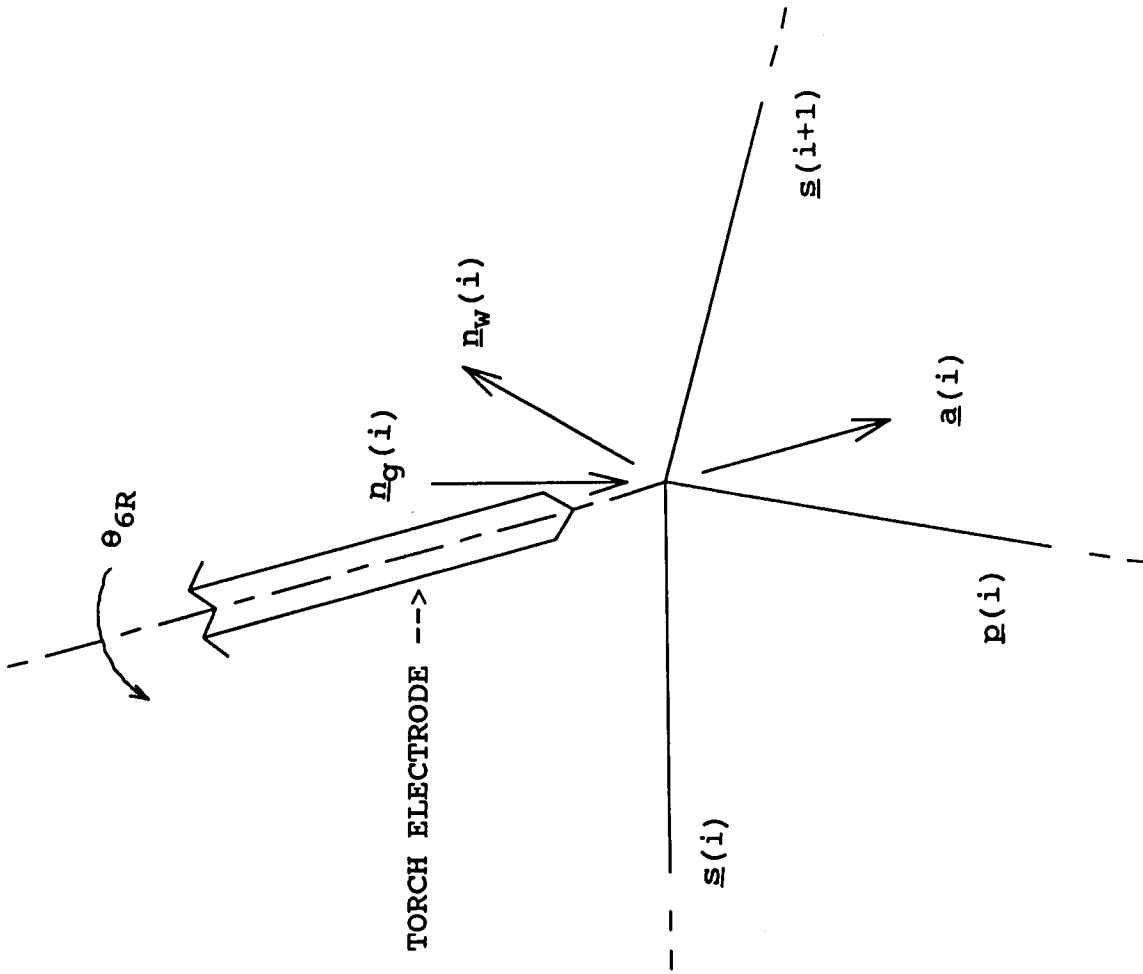
Figure 5.   Path definitions for wire feed orientation.

ROBOSIM
NASA/VANDERBILT
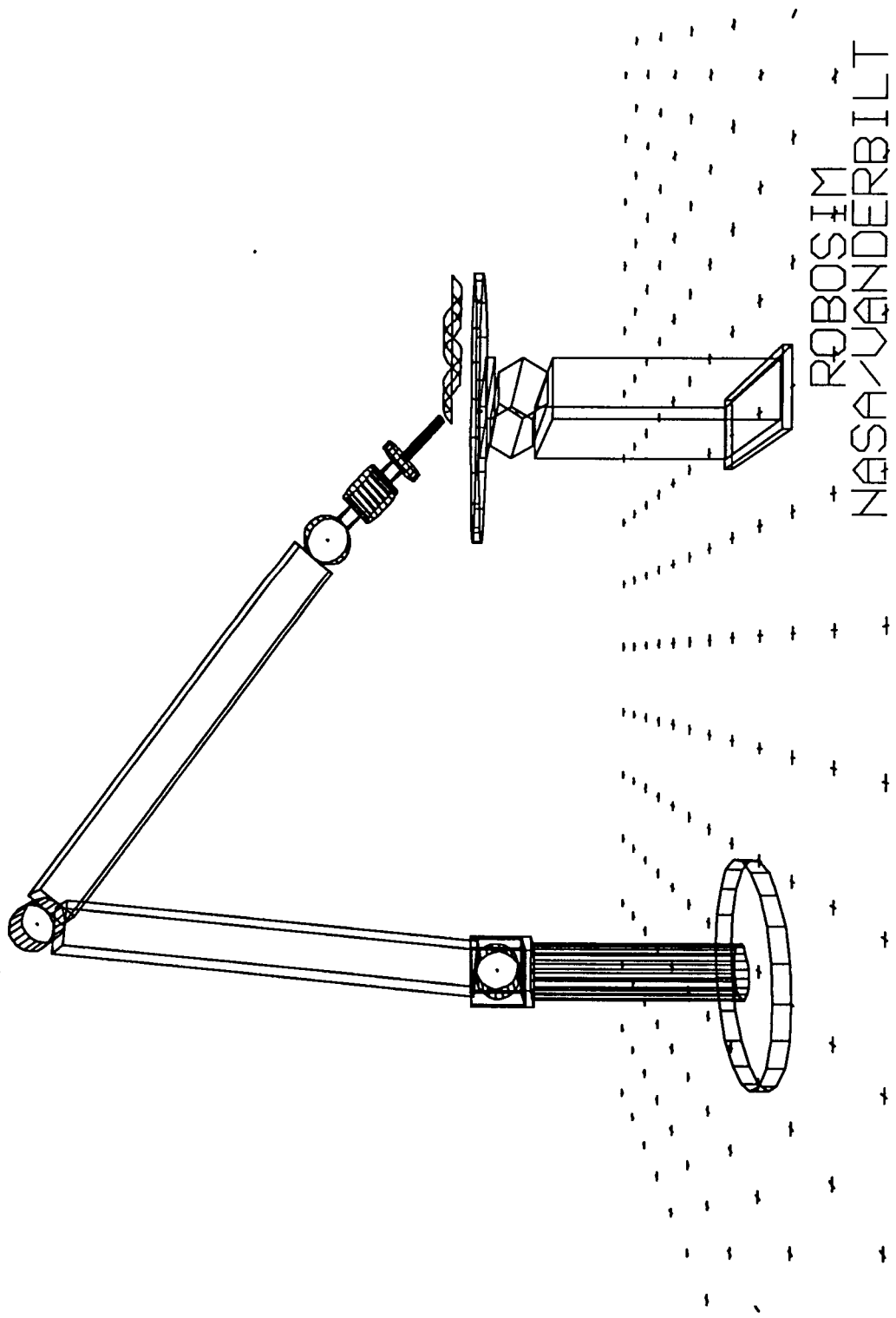
Figure 6. Simulated workcell.
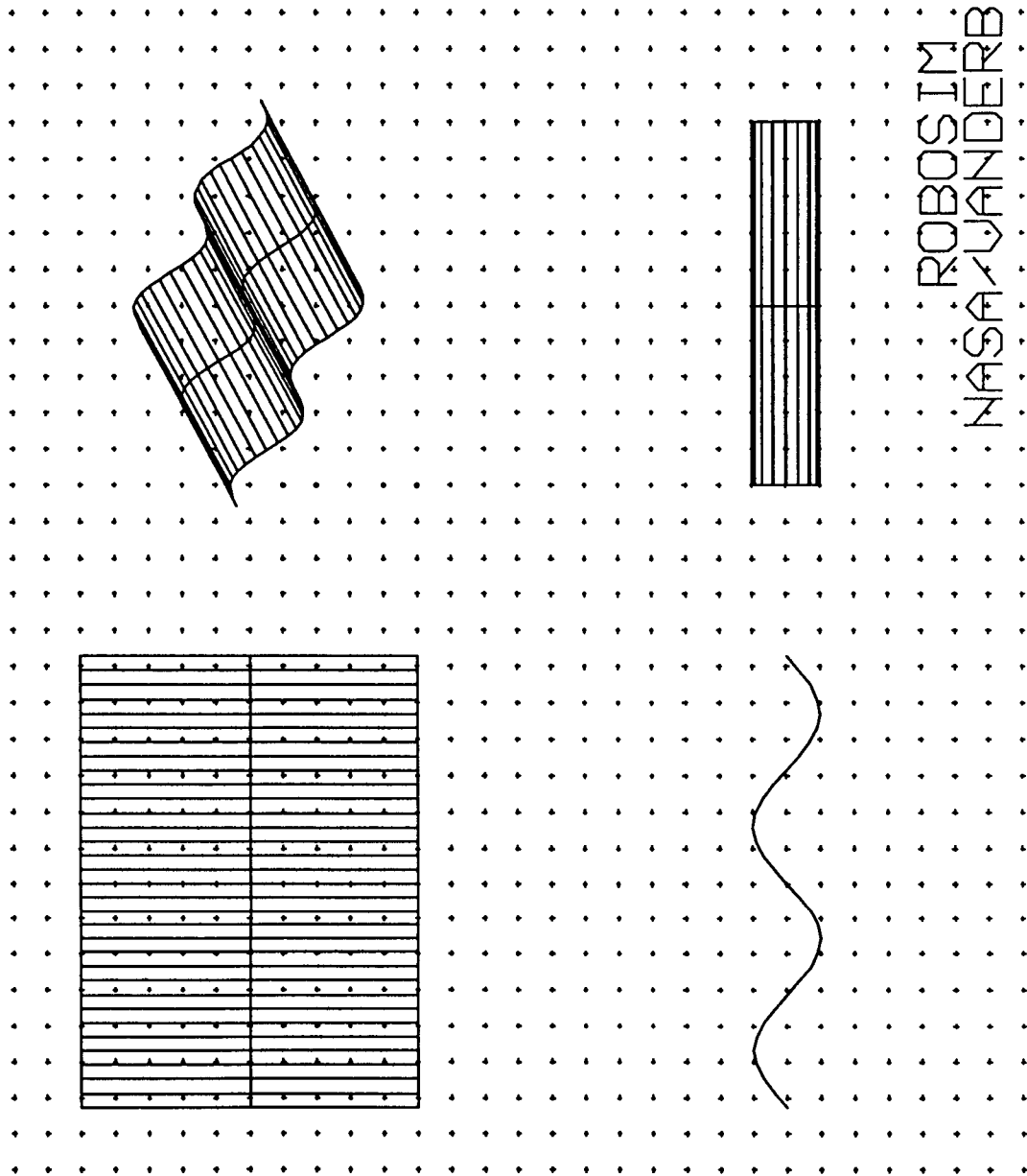
Figure 7. Part detail.
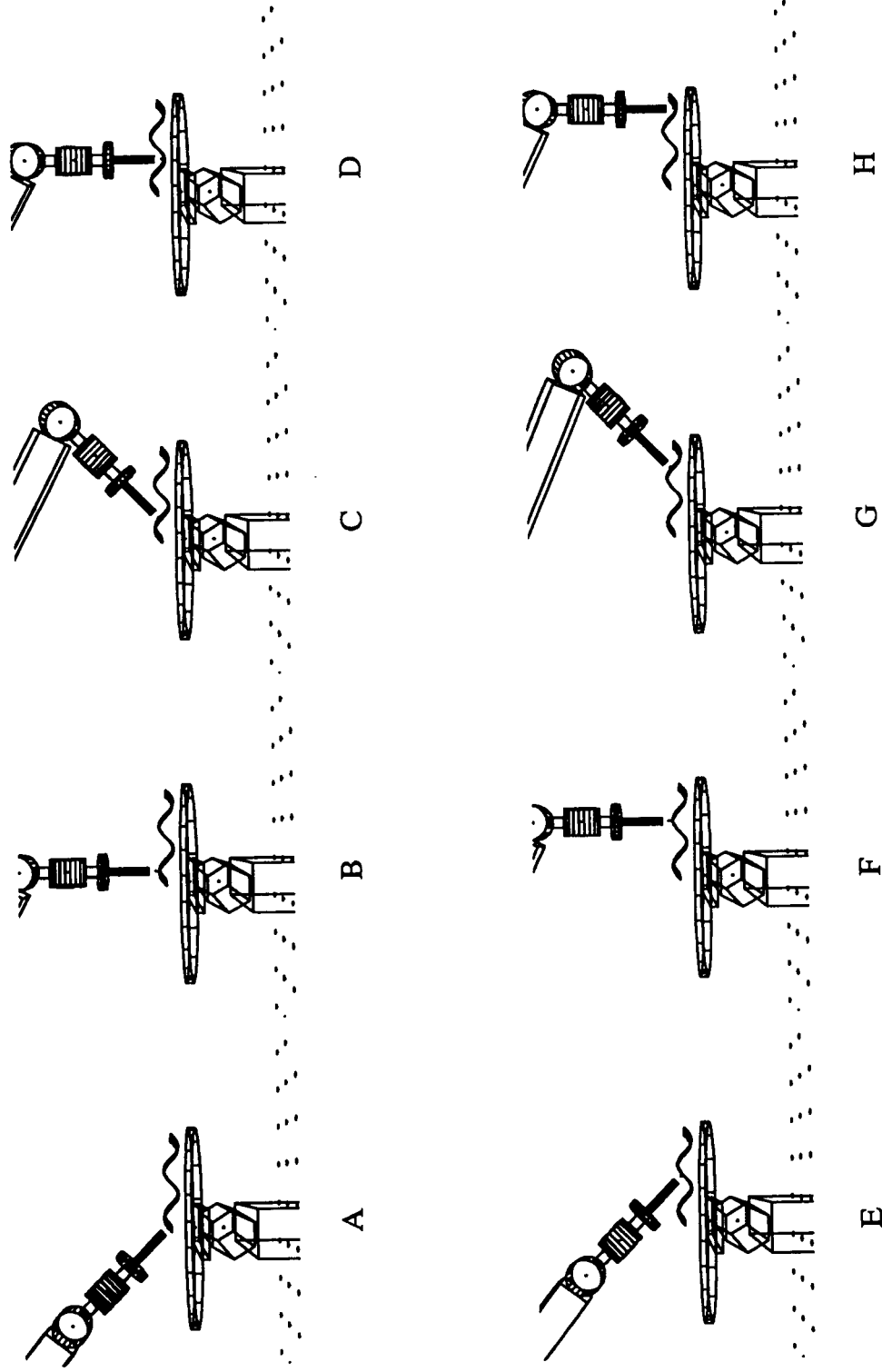
Figure 8. Robot path without downhand control.
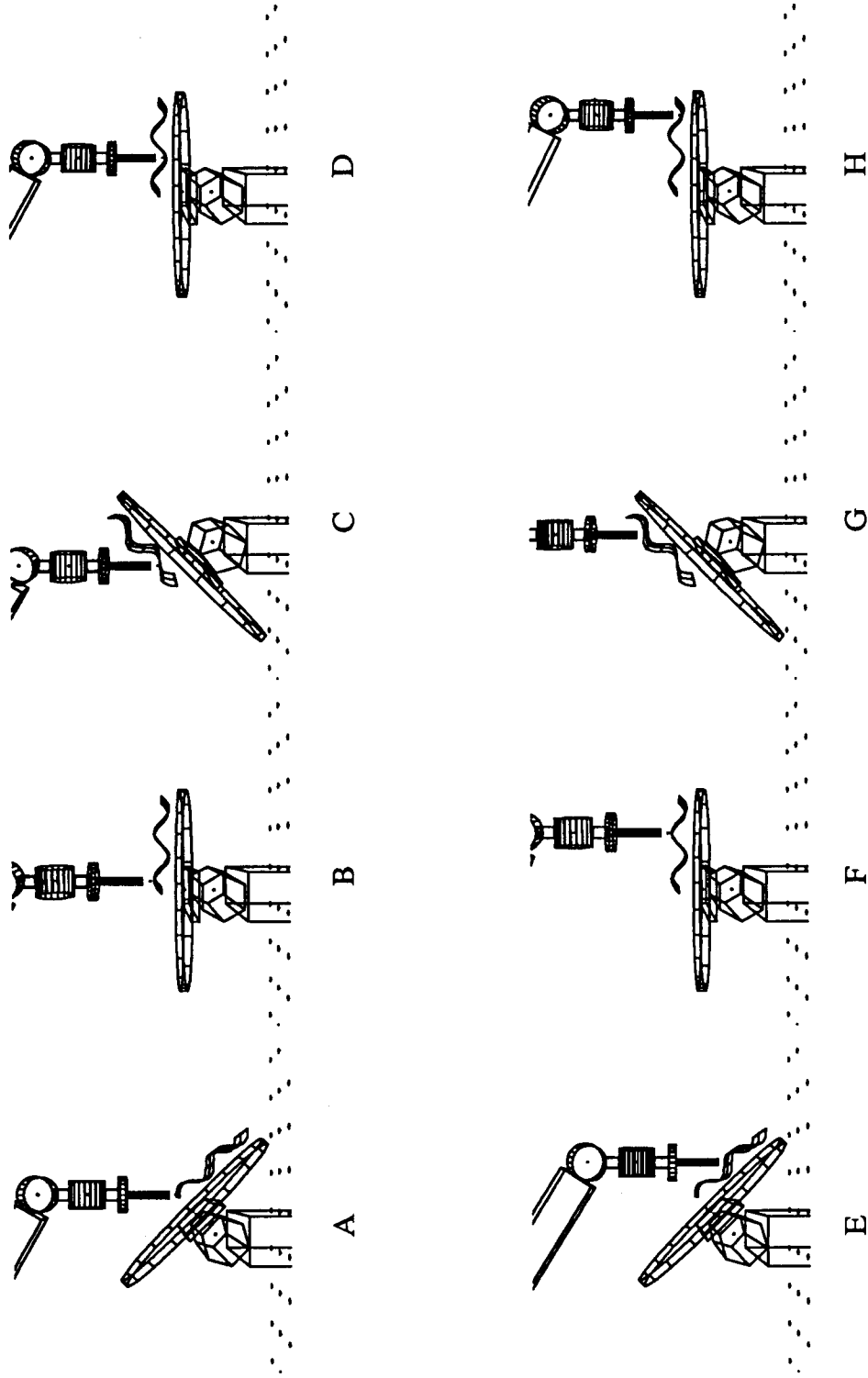
Figure 9.   Robot path with downhand control.

41
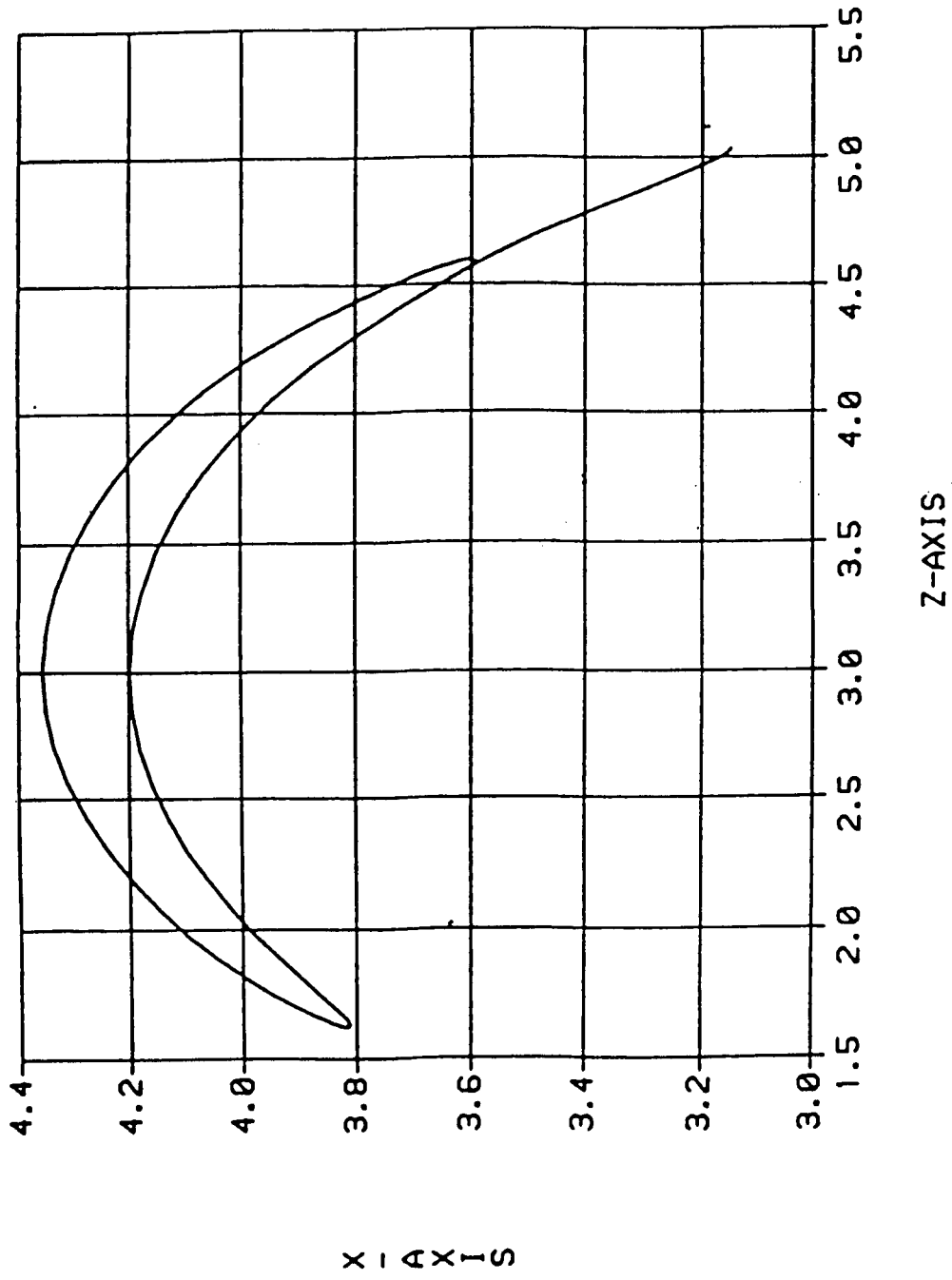
Figure 10.   Torch trajectory in workcell coordinates.

Z-AXIS

X-AXIS
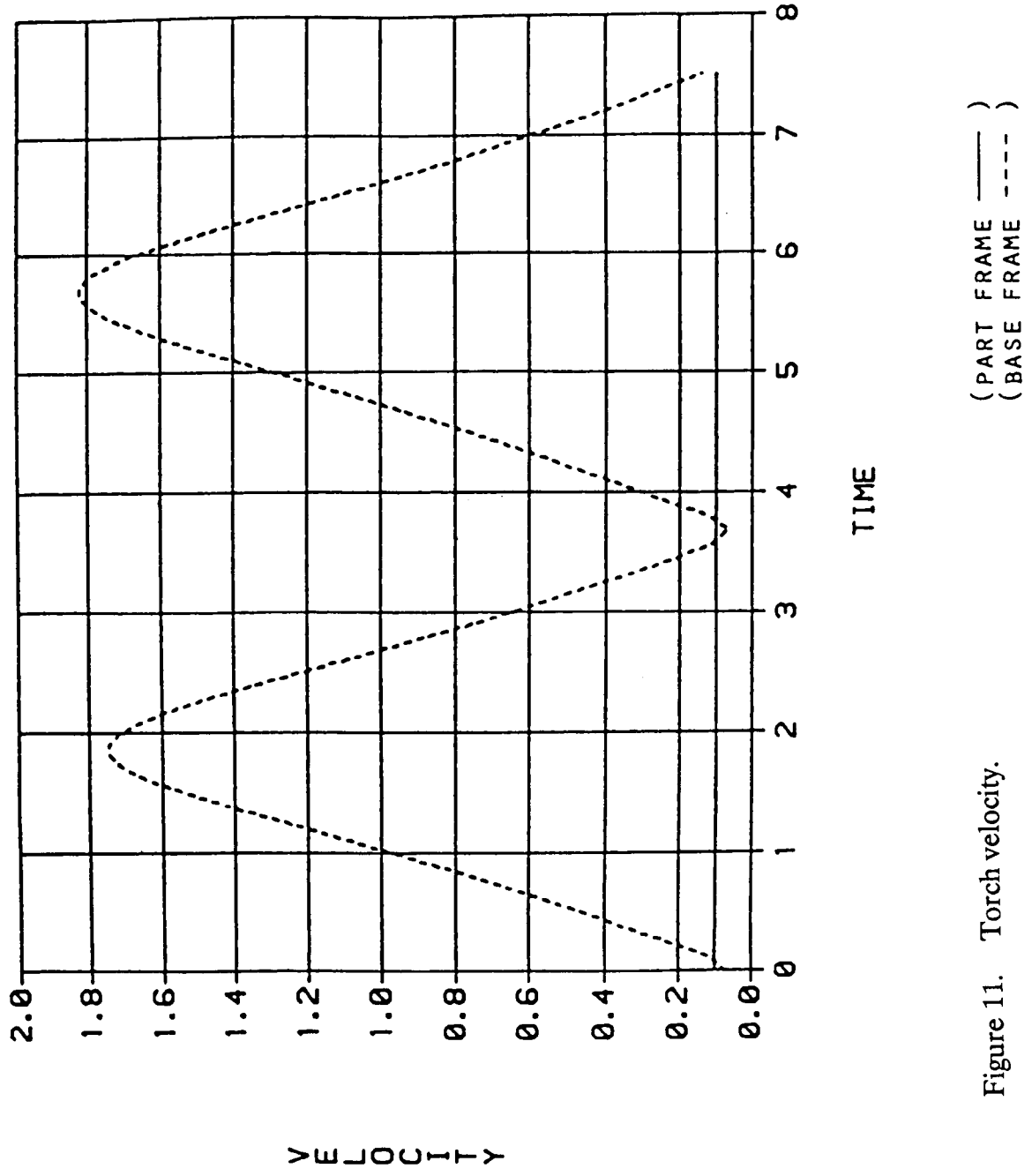
Figure 11.   Torch velocity.

Figure 12.   Part with oscillating path.
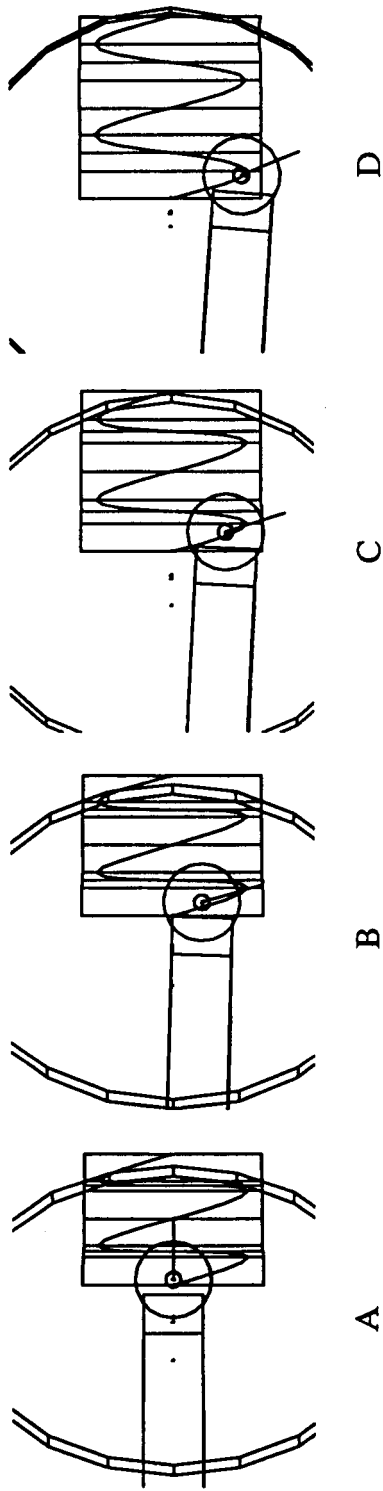
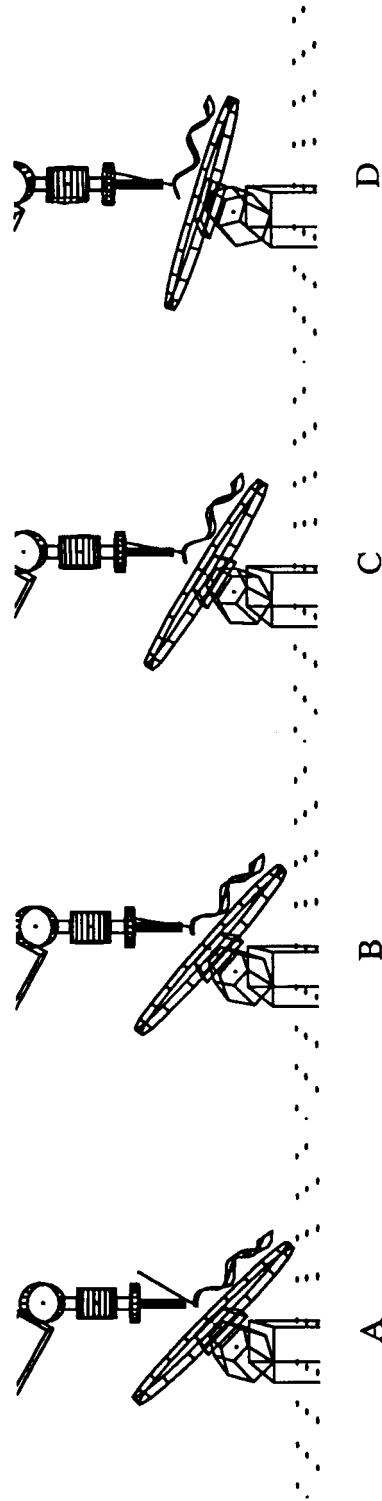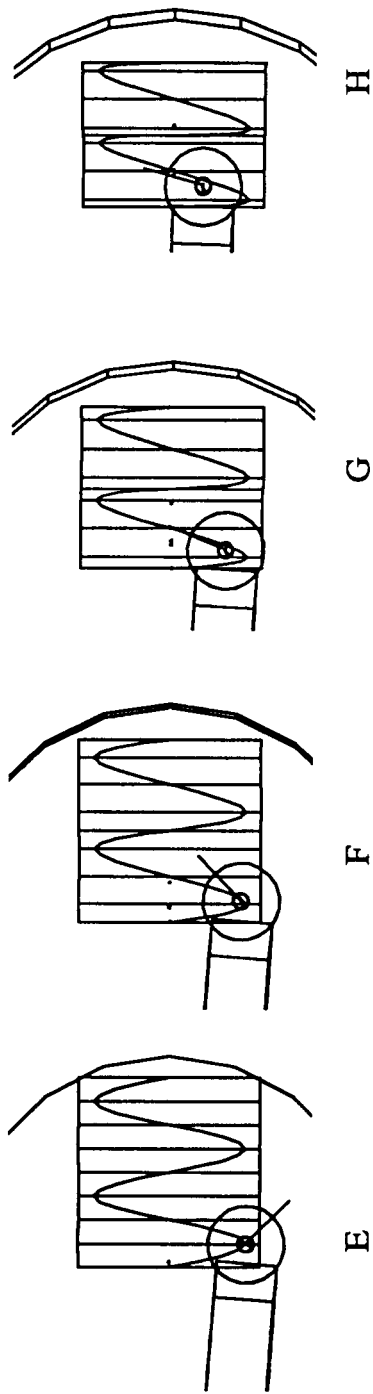Figure 14. Animation of downhand and wire feed orientation control function (top views).

Figure 13. Animation of downhand and wire feed orientation control function (front views).

Figure 14. Animation of downhand and wire feed orientation control function (top views).
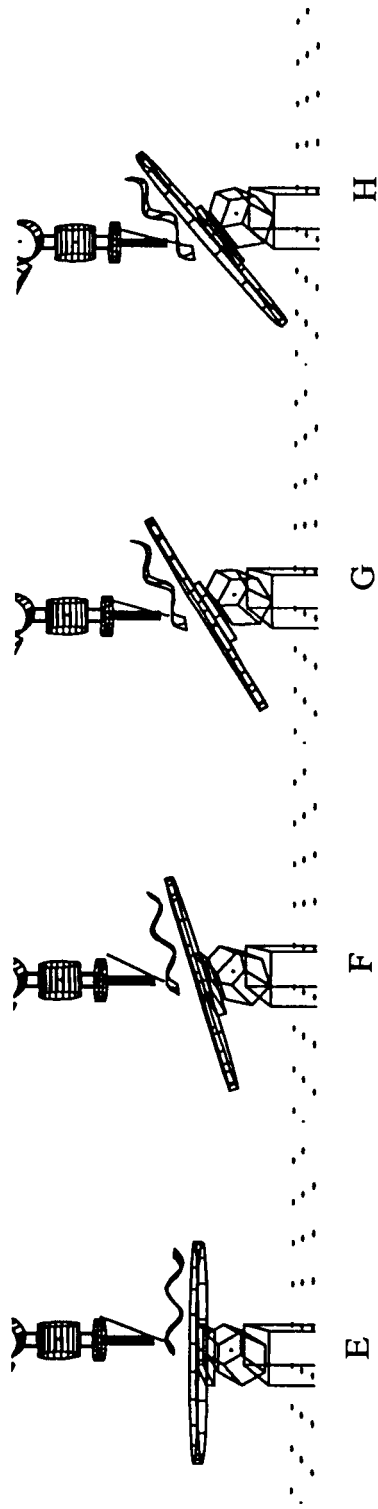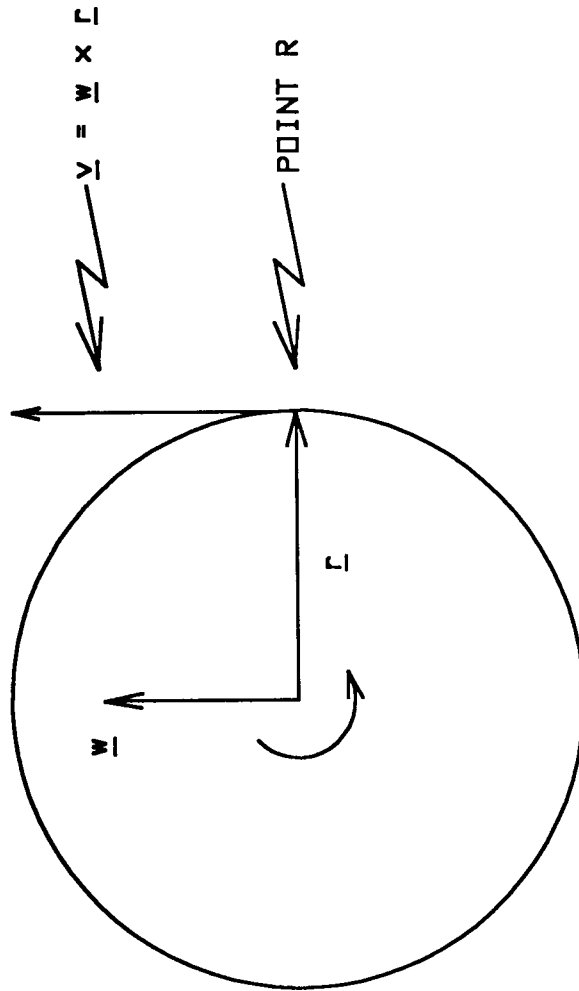

Figure 13. Animation of downhand and wire feed orientation control function (front views).

$$[J] = \begin{bmatrix} \underline{v} \\ \underline{w} \end{bmatrix} \text{ AT POINT R}$$

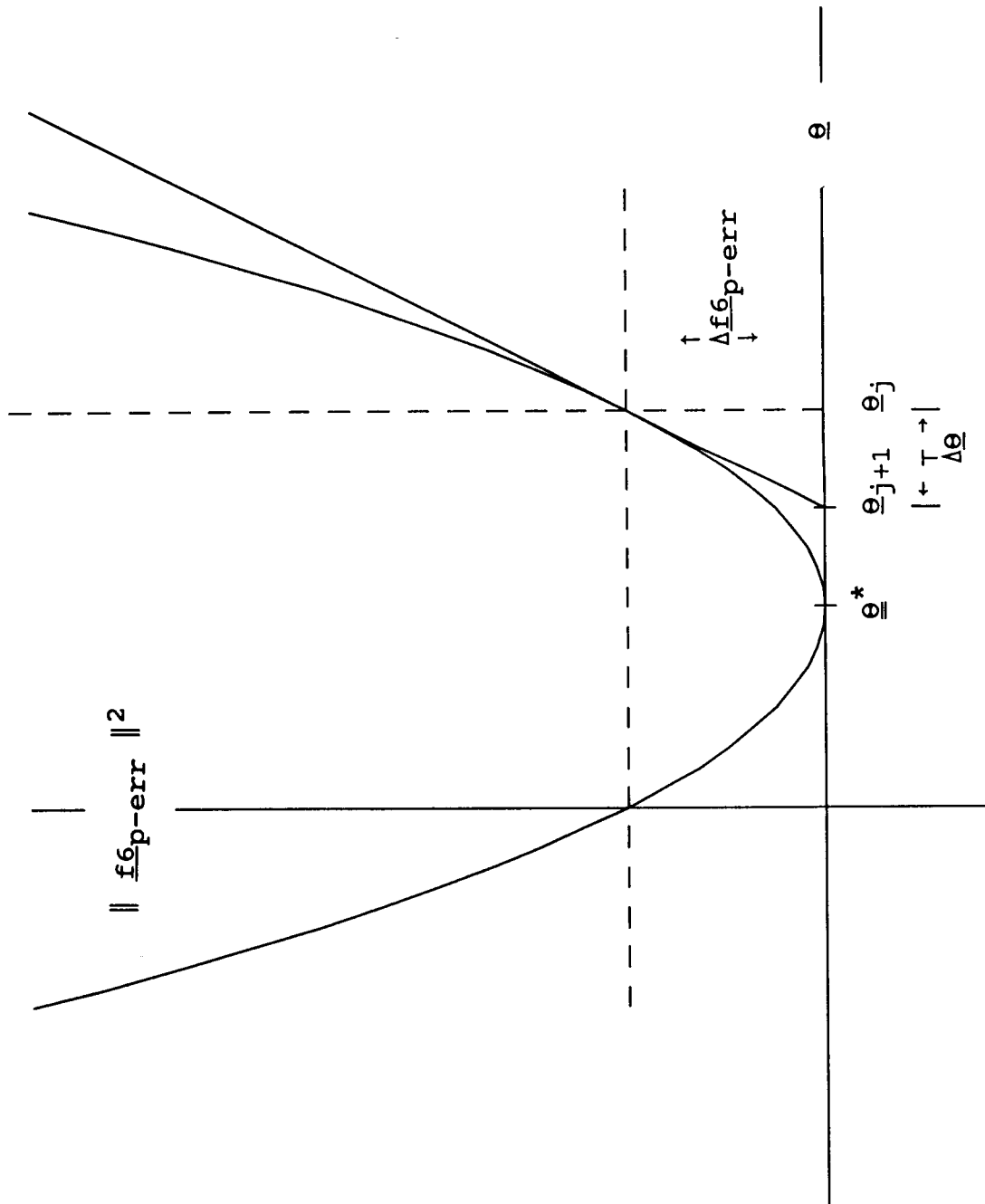Figure A-1.   Physical interpretation of the Jacobian.

Figure B-1. Six DOF error function characteristic.

1. REPORT NO. NASA TP-2807	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE A Generalized Method for Automatic Downhand and Wirefeed Control of a Welding Robot and Positioner		5. REPORT DATE February 1988
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Ken Fernandez and George E. Cook*		8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		10. WORK UNIT NO. M-583
		11. CONTRACT OR GRANT NO.
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546		13. TYPE OF REPORT & PERIOD COVERED Technical Paper
		14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Prepared by Information and Electronic Systems Laboratory, Science and Engineering Directorate. *Vanderbilt University, Nashville, Tennessee

16. ABSTRACT

This paper describes a generalized method for controlling a six degree-of-freedom (DOF) robot and a two DOF positioner used for arc welding operations. The welding path is defined in the part reference frame, and robot/positioner joint angles of the equivalent eight DOF serial linkage are determrined via an iterative solution. Three algorithms are presented: the first solution controls motion of the eight DOF mechanism such that proper torch motion is achieved while minimizing the sum-of-squares of joint displacements; the second algorithm adds two constraint equations to achieve torch control while maintaining part orientation so that welding occurs in the downhand position; and the third algorithm adds the ability to control the proper orientation of a wire feed mechanism used in gas tungsten arc (GTA) welding operations. A verification of these algorithms is given using ROBOSIM, a NASA developed computer graphic simulation software package designed for robot systems development.

17. KEY WORDS Robotic welding Off-line programming CAD/CAM/CIM Downhand position ROBOSIM	18. DISTRIBUTION STATEMENT Unclassified — Unlimited Subject Category: 31

19. SECURITY CLASSIF. (of this report) Unclassified	20. SECURITY CLASSIF. (of this page) Unclassified	21. NO. OF PAGES 54	22. PRICE A04