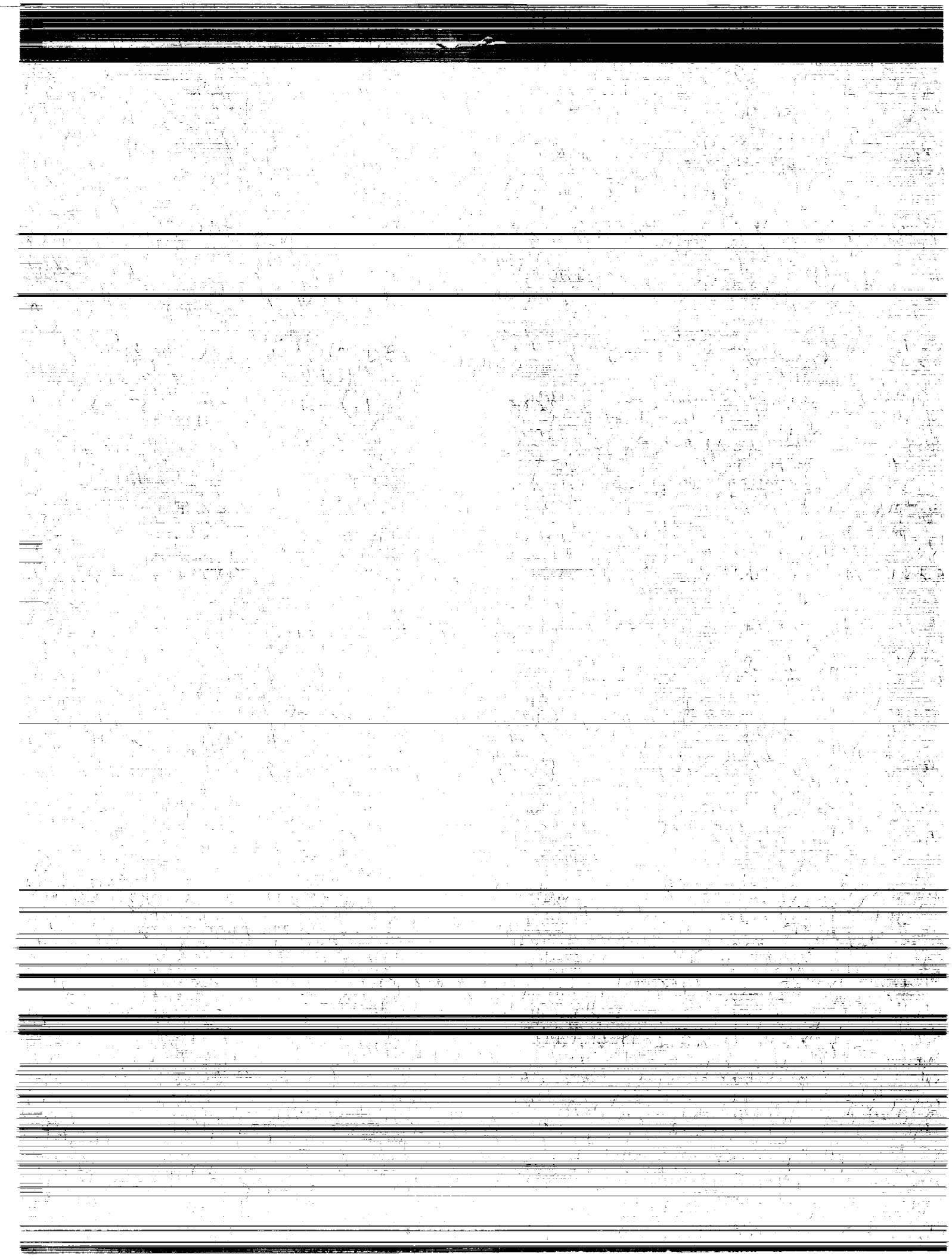
NASA
Technical
Paper
3163

October 1991

A Generalized Method for Multiple Robotic Manipulator Programming Applied to Vertical-Up Welding

Kenneth R. Fernandez,
George E. Cook,
Kristinn Andersen,
Robert Joel Barnett,
and Saleh Zein-Sabattou

NASA



NASA Technical Paper 3163

1991

A Generalized Method for Multiple Robotic Manipulator Programming Applied to Vertical-Up Welding

Kenneth R. Fernandez
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

George E. Cook,
Kristinn Andersen,
Robert Joel Barnett,
and Saleh Zein-Sabattou
Vanderbilt University
Nashville, Tennessee

NASA
National Aeronautics and Space Administration
Office of Management
Scientific and Technical Information Program

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DEFINITION OF SYMBOLS

Symbol	Definition
$\underline{A}_i$	variable transformation for link *i* in a robotic manipulator
CAD	computer aided design
CAM	computer aided manufacturing
DOF	degrees-of-freedom
$\underline{E}$	fixed transformation for an end effector
$\underline{Err}(i)$	variable transformation. (the difference between the actual and the desired torch location in the part coordinate frame.)
$\underline{f}_{err}$	a forcing function, used to eliminate the difference between the actual and the desired torch location in the part coordinate frame
$\underline{G}$	fixed transformation from the last link of the positioner to the part coordinate frame
$\underline{g}$	the workcell gravity vector
GTAW	gas tungsten arc welding
$\underline{I}$	an identity matrix
$\underline{J}$	a Jacobian matrix
mm	millimeter
MSFC	Marshall Space Flight Center
N-DOF	N degrees-of-freedom
NASA	National Aeronautics and Space Administration

$\underline{P}(i)$	variable robot path transformation, defined in a part (workpiece) coordinate frame
ROBOSIM	ROBOt SIMulation package
s	second
$\underline{T}_6$	transformation matrix for a 6 DOF manipulator
VPPAW	variable polarity plasma arc welding
$\underline{Z}$	fixed transformation indicating location in the world coordinates

UNUSUAL TERMS

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

vertical-up welding - Welding with the part maintained in an orientation such that its surface plane at the weld puddle is vertical, and the traveling motion of the puddle, relative to the part, is up.

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

wire feed - When welding with a nonconsumable electrode, the filler metal 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 MULTIPLE ROBOTIC MANIPULATOR PROGRAMMING APPLIED TO VERTICAL-UP WELDING

INTRODUCTION

The work presented in this paper is an extension of a past research effort to automate the programming of robotic mechanisms and positioners to satisfy geometrical constraints arising in robotic arc welding. The research, which forms the basis of the methodology used here, was originally carried out by Fernandez and Cook and published by NASA [1]. The work was demonstrated using the constraints imposed on the robotic manipulators by the gas tungsten arc welding (GTAW) process. The basic methodology developed by Fernandez and Cook, however, was very general and applicable to a variety of other welding processes. This was emphasized and discussed further by Andersen et al. [2], where specific modifications to the original programming algorithm necessary to facilitate vertical-up welding were given. In this paper, the full details for adapting the original algorithm for vertical-up welding are presented. Furthermore, graphical simulations are illustrated to verify the results from the algorithm modifications.

The practical problems reduced or eliminated by the programming algorithm are now summarized. Each teach point along any given weld has to be programmed with certain constraints satisfied. The workpiece has to be appropriately oriented, the torch tip has to be appropriately positioned and oriented, the wire feed mechanism should be oriented in a certain way, and welding speed is to be controllable at all times. The first three requirements can be met with considerable patience on the programmer's behalf using a robot and a positioner, manually

guided with a teach pendant. Although viable, this approach is tedious and prone to programming errors. In an integrated CAD/CAM system, this step may not be easily achievable without an algorithm such as the one presented here. Most welding systems have more than 6 DOF and the problem of how to coordinate the redundant joints and simultaneously satisfy the welding constraints has to be resolved. The fourth requirement, programmability for constant or controllable welding speed, is provided for by neither the teach-pendant approach nor the unmodified CAD/CAM approach.

A programmer using the proposed algorithm can save substantial time and ensure accuracy by applying the algorithm in the following manner. In the case of programming the weld path by guiding the robot with a teach pendant, the programmer can orient the workpiece by aligning the positioner joints into any convenient position at the beginning of the programming sequence as well as at any time throughout the programming process. The torch is guided to the first teach point on the workpiece, where the programmer ensures only adequate proximity to the desired teach point location and the desired torch orientation with respect to the workpiece. The programmer does not have to be concerned with wire feed orientation or workpiece orientation. This procedure can be repeated for all teach points. In the case of a CAD/CAM system, the coordinates and desired torch orientations at the selected teach points can be extracted from the workpiece data, from which all information for the algorithm can be derived. In either case, the desired welding speed or speeds are also specified. From this information, the proposed welding algorithm calculates the positioner, robot joint angles, and robot speeds required to make the weld with the desired welding constraints. In short, a process which was, at best, a series of tedious trial-and-error operations is turned into a relatively straightforward, one-pass task using the proposed algorithms.

WELDING CONSTRAINTS: VERTICAL-UP VERSUS DOWNHAND ORIENTATION

The various arc welding processes, such as GTAW and variable polarity plasma arc welding (VPPAW), demand that the welded workpiece and the welding torch be appropriately positioned and oriented at all times during welding to yield satisfactory welds. Additionally, the speed of the torch tip with respect to the surface of the workpiece has to be fully controllable.

The VPPAW process is used in a number of welding tasks at NASA, including welding on the space shuttle and planned welding on the space station. It utilizes a nonconsumable electrode which sustains the welding arc on the workpiece. The basic elements of the plasma arc torch are the electrode and the orifice. A relatively small flow of an inert gas (e.g., argon) is guided through the orifice to form the arc plasma. The arc may either be sustained between the electrode and the workpiece (transferred arc) or between the electrode and the constricting nozzle (nontransferred arc). Keyhole welding, a specific condition of VPPAW, is achieved with certain combinations of base metal thicknesses, gas flow rates, currents, and torch travel speeds. The weld pool is relatively small and contains a hole penetrating completely through the base metal. Manual VPPAW, as well as the GTAW process, can be carried out in most welding positions. Automatic VPPAW is more limited in terms of possible welding positions, and keyhole welding is frequently preferred with the torch always traveling vertically up along the seam.

The workpiece is held by a part positioner, which may be viewed as an N-DOF robotic mechanism itself. The robot (manipulator) holds the welding torch, including the electrode, shielding gas nozzle, wire feeder which supplies the molten weld pool with reinforcement material, and other associated hardware as applicable.

To obtain a satisfactory weld, the torch and the workpiece have to be manipulated in certain manners, depending on the applied welding process. The first requirement is that the torch tracks the joint to be welded within acceptable tolerances. Secondly, the orientation of both the workpiece and the torch has to be appropriate for the welding process. When the VPPAW process is applied, the workpiece has to be aligned so that the welded spot is on a vertical surface at all times, and, furthermore, the workpiece may have to be rotated so that the molten spot moves vertically up during the entire welding pass. The term "vertical-up welding" is derived from these requirements. In the case of the GTAW process, on the other hand, the molten weld pool is always on a horizontal surface section of the workpiece. The arc plasma is directed from the torch above the workpiece and down into the pool, which gives rise to the term "downhand welding." The workpiece orientation requirements for the different welding processes are usually accomplished with the part positioner.

Regarding the torch manipulator there are a number of requirements, or constraints, which have to be met as well. Usually the filler wire is fed from the front of the arc plasma as it moves along the welded joint. The angles between the arc plasma flow, the workpiece surface, and the instantaneous direction of the torch travel may need to be maintained at specific values of orientation during an entire weld pass. Finally, the speed of the torch tip, with respect to the workpiece surface, has to be completely controlled. Typically, the speed is kept constant at a value selected by the operator. For applications where the positioner is stationary during the entire welding pass this is a trivial task. In cases requiring simultaneous, coordinated movements of the positioner and the manipulator, however, speed control becomes more complex. The simultaneous motion of the positioner and the manipulator constitute complications for programming the teach points for the positioner and the manipulator as well. The methodology described in this paper

simplifies programming robotic welding systems where the constraints discussed above must be simultaneously satisfied.

THE GENERALIZED PROGRAMMING METHODOLOGY

A short review of the generalized programming algorithm, developed by Fernandez and Cook [1], is given in this section. Further details can be found in this original NASA paper. Robotic terminology and notations adhere largely to the conventions given by Paul [3].

The proposed weld path programming methodology is most clearly outlined by reference to the coordinate transformation chains through the manipulator on one hand and through the part positioner on the other. The position and the orientation of the robot (or, more precisely, a coordinate system attached to the robot base) with respect to the world coordinate system is specified by the fixed transformation matrix $\underline{Z}_R$. Note that this allows the system user to select the world coordinates arbitrarily. Therefore, the user may select the world coordinate system so that it coincides with the robot base coordinate system, in which case the displacement and rotations specified by $\underline{Z}_R$ are zero. The total transformation relating the final link of the robot to its base coordinate system is specified by the variable matrix $\underline{T}_R(i)$ which changes as the robot moves along the programmed points designated by the index i. For each program point the elements of $\underline{T}_R$ are functions of one or more joint variables. Here, it is assumed that the joint variables of the robot are $\theta_{1R}, \theta_{2R}, ..., \theta_{6R}$. The welding torch, or end effector, is mounted on the end link of the robot. The location of a coordinate system attached to the torch, with respect to the end link, is specified by the fixed transformation matrix $\underline{E}$. For the torch to track the welding seam and be properly oriented with respect to the seam, the total

transformation $\underline{Z}_R\ \underline{T}_R(i)\ \underline{E}$ has to be appropriately specified for each programmed point, i.

The positioner side of the transform chain is specified similarly. The positioner location in the world coordinates is specified by the fixed transformation $\underline{Z}_P$. The location of a coordinate system attached to the positioner mounting frame with respect to the positioner base coordinates is $\underline{T}_P(i)$. Only two joint variables, θ_{1P} and θ_{2P}, are assumed here for the positioner. Each workpiece, or part, mounted on the positioner is assigned a coordinate system, which is fixed with respect to the part. The transformation from the positioner mounting frame to the part coordinate system is specified by the fixed transformation $\underline{G}$. Now that a fixed coordinate system has been defined with respect to the workpiece, the trajectory of the welded seam is specified. To match the manipulator and positioner transform chains, the transformation $\underline{P}(i)$ (programmed point i on workpiece with respect to part frame) is specified so that the welding torch coordinates coincide with the coordinates of the programmed point on the workpiece:

$$\underline{Z}_R\ \underline{T}_R(i)\ \underline{E} \;=\; \underline{Z}_P\ \underline{T}_P(i)\ \underline{G}\ \underline{P}(i) \qquad (1)$$

for all programmed points, i.

The purpose of the generalized programming algorithm is to calculate the eight joint variables, θ_{1P}, θ_{2P}, and θ_{1R}, θ_{2R}, through θ_{6R}, for each programmed point so that the following four requirements are fulfilled at all times:

1. Torch position control: The torch tip has to track the weld seam and be properly oriented with respect to the seam throughout the entire sequence of programmed points.

2. Workpiece orientation control: While the torch has to be appropriately oriented with respect to welded seam at each programmed point, the part should simultaneously be appropriately oriented in the workcell. For example, this orientation depends on whether vertical-up or downhand welding is applied.

3. Wire feed orientation control: The wire feed contact tube is to be properly oriented in front of the moving torch throughout the entire weld sequence.

4. Speed control: The speed of the torch tip, relative to the part surface, is to be controllable. A special case of this is the constant speed requirement usually desired.

Basically, the first three control tasks are accomplished by determining the error, or difference, between the actual and desired vector relationships, and then Newton's algorithm is iteratively applied to reduce this difference to zero. Once the desired positions of the torch manipulator and the robot have been determined, the velocity of each joint is calculated, based on the desired welding speed.

The part of the overall algorithm which controls and maintains torch position with respect to the workpiece illustrates the general concept. Before the algorithm is executed, the two transform chains, through the welding manipulator and through the workpiece positioner, are generally not identical (because $\underline{T}_R$ and $\underline{T}_P$ are not yet determined), and the difference is represented by an error transformation $\underline{Err}$:

$$\underline{Err} = \underline{E}^{-1}\,\underline{T}_R^{-1}(i)\,\underline{Z}_R^{-1}\,\underline{Z}_P\,\underline{T}_P(i)\,\underline{G}\,\underline{P}(i) \qquad (2)$$

This error transformation is the transformation between the desired torch location (position and orientation in the part frame) to be reached by the algorithm and the actual, initial location. When the actual location equals the desired location, Err becomes the identity matrix. Generally, the error Err can be represented as successive operations of rotations and translations. Before the torch position control part of the algorithm is implemented, the eight joint variables of the system, θ_{1P}, θ_{2P} (two positioner joints), and θ_{1R}, θ_{2R},...,θ_{6R} (six manipulator joints), may have some arbitrary values. These eight variables have to be determined so that Err becomes the identity matrix. In general, there is no closed-form solution to this problem, and therefore an iteration method has to be applied. A six-element vector variable, or forcing function, $\underline{f6}_{p\text{-}err}$, is constructed from the three position errors and the three rotation errors, so that it becomes zero when the torch is correctly positioned and oriented. This is when Err becomes the identity matrix.

It may be helpful to keep in mind how a zero is found for an arbitrary function, f(x), of one variable using Newton's method. A new value for x is iteratively calculated and tried until the corresponding function value is reduced below a specified threshold:

$$x_{j+1} = x_j - [1/f'(x_j)]\, f(x_j) \tag{3}$$

until

$$f^2(x) < \epsilon \tag{4}$$

In this one-dimensional case $f'(x_j)$ is the first derivative of f(x), evaluated at the jth iteration estimate of x and ϵ, the convergence criterion, is a small constant. This

approach can be extended to functions of more than one variable, such as $\underline{f6}_{p\text{-}err}$. In that case the iteration algorithm becomes:

$$\underline{\theta}_{j+1} = \underline{\theta}_j - [\partial/\partial\underline{\theta}\ \underline{f6}_{p\text{-}err}(\underline{\theta}_j)]^{-1}\ \underline{f6}_{p\text{-}err}(\underline{\theta}_j) \tag{5}$$

until

$$[\underline{f6}_{p\text{-}err}(\underline{\theta})]^T\ [\underline{f6}_{p\text{-}err}(\underline{\theta})] < \epsilon \tag{6}$$

Here, the variable $\underline{\theta}_j$ is an eight-element vector consisting of the eight system joint variables, as they are evaluated at the jth step of the iteration process.

Differentiation of the $\underline{f6}_{p\text{-}err}(\underline{\theta})$ matrix yields its Jacobian, which relates joint angle rates to torch displacement and rotation rates. In this case the Jacobian, $\underline{J}$, is a 6-by-8 matrix of the following form:

$$\underline{J} = \begin{bmatrix} \partial x/\partial\theta_1 & \dots & \partial x/\partial\theta_8 \\ \partial y/\partial\theta_1 & \dots & \partial x/\partial\theta_8 \\ \partial z/\partial\theta_1 & \dots & \partial x/\partial\theta_8 \\ \partial\phi_x/\partial\theta_1 & \dots & \partial\phi_x/\partial\theta_8 \\ \partial\phi_y/\partial\theta_1 & \dots & \partial\phi_y/\partial\theta_8 \\ \partial\phi_z/\partial\theta_1 & \dots & \partial\phi_z/\partial\theta_8 \end{bmatrix} \tag{7}$$

where, e.g., $\underline{J}_{2,3}$ is the differential of the torch y-coordinate in the part frame with respect to the second joint in the system. $\partial\phi_x$ denotes a differential rotation about the x axis, and similar notations hold for rotations about the y and z axes. Each column of the Jacobian matrix is composed of three translational derivatives, followed by three rotational derivatives.

The Jacobian for any robotic mechanism is readily calculated as long as all link coordinates are assigned according to the Denavit-Hartenberg conventions, which are discussed by Paul [3]. Because the joint variables of $\underline{\theta}$ are eight while the

forcing function vector, $\underline{f6}_{p\text{-}err}$, contains only six variables, the system is underdetermined. In other words, the system is redundant and an infinite number of solutions exists for $\underline{\theta}$. One approach to resolving this is to replace the inverse of the Jacobian with its pseudoinverse. This guarantees a unique solution for $\underline{\theta}$ which minimizes the joint displacements in the least-square sense. For further control of the convergence, the inverted Jacobian may be multiplied by a constant scalar, h, which determines the step size for the iterations. Finally, different weights for the elements of $\underline{f6}_{p\text{-}err}$ may be preferred in the convergence criterion. This is accomplished by multiplication by a diagonal weighting matrix K. The resulting iteration algorithm for torch position control is therefore:

$$\underline{\theta}_{j+1} = \underline{\theta}_j - h\,[\underline{J}^T(\underline{J}\underline{J}^T)^{-1}]\,\underline{f6}_{p\text{-}err}(\underline{\theta}_j) \tag{8}$$

$$[\underline{f6}_{p\text{-}err}(\underline{\theta})]^T\,\underline{K}\,[\underline{f6}_{p\text{-}err}(\underline{\theta})] < \epsilon \tag{9}$$

These equations describe only the algorithm for position and orientation control of the torch with respect to the welded part. Constraints for appropriate workpiece orientation for vertical-up or downhand welding and wire feed orientation control require additional calculations along similar lines.

Proper workpiece orientation at each point along the weld joint is basically achieved by defining a two-element vector forcing function, $\underline{f2}_{err}$, and the two positioner joint variables are iteratively manipulated until this function becomes zero. For vertical-up welding, this function can be the cross product of a vector tangent to the welded seam and the gravity vector of the workcell. The iteration algorithm will alter the two positioner joint variables until the seam tangent becomes vertical. For downhand welding, the forcing function is defined as the cross product of the vector normal to the workpiece surface and the gravity vector. In that case, the algorithm iteratively orients the workpiece until its surface becomes

horizontal at the welding point. The generality of the basic methodology is clearly illustrated in that a simple redefinition of $\underline{f2}_{err}$ adapts the algorithm to a new welding process with totally different workpiece orientation requirements.

It should be noted that forcing both $\underline{f6}_{p\text{-}err}$ and $\underline{f2}_{err}$ to zero yields eight equations, and with the eight joint variables available in the system, there is only one set of solutions for the joint variables. In this case the normal inverse, rather than the pseudoinverse, of the system Jacobian can be used. A new forcing function, $\underline{f8}_{t\text{-}err}$, is composed of the elements of $\underline{f6}_{p\text{-}err}$ and $\underline{f2}_{err}$. Again, using Newton's method, the resulting set of equations can be solved for the joint variables stored in $\underline{\theta}$:

$$\underline{\theta}_{j+1} = \underline{\theta}_j - h\, \underline{J}^{-1}\, \underline{f8}_{t\text{-}err}(\underline{\theta}_j) \tag{10}$$

$$[\underline{f8}_{t\text{-}err}(\underline{\theta})]^T\, \underline{K}\, [\underline{f8}_{t\text{-}err}(\underline{\theta})] < \epsilon \tag{11}$$

The joint values resulting from this algorithm ensure that the torch is properly positioned and oriented with respect to the weld seam, and the workpiece is appropriately oriented based on the welding process being used.

In most welding applications, the wire fed into the molten pool is applied from the front of the welding torch as it moves along the seam. This constraint has to be added to the robotic welding system while maintaining proper torch position and workpiece orientation. However, because the system is already fully determined (eight equations for eight unknown variables), addition of one more constraint requires either an additional DOF or relaxation of one of the previous constraints. Rotation of the torch about the approach axis is irrelevant as far as the already established constraints are concerned, and therefore that DOF will be used here for wire feed control.

Again, a forcing function for aligning the wire feed orientation is determined. This scalar forcing function, $f1_{w\text{-}err}$, is defined so that it becomes zero when the vector along the wire feed direction is co-planar with the tangent to the weld path and the normal to the workpiece surface.

To satisfy all of the previously stated constraints, the user of these algorithms determines all eight joint variables using the iterative calculations used to make $\underline{f6}_{p\text{-}err}$ and $\underline{f2}_{g\text{-}err}$ equal to zero. Then one of the eight solutions is revised by the following algorithm:

$$\theta_{6R,j+1} = \theta_{6R,j} + h\,\underline{f1}_{w\text{-}err} \tag{12}$$

$$k\,(\underline{f1}_{t\text{-}err})^2 < \epsilon \tag{13}$$

Therefore, the last joint variable of the robot, θ_{6R}, as calculated here, replaces the one calculated previously when $\underline{f8}_{t\text{-}err}$ was forced to zero. The resulting vector of the eight joint variables, $\underline{\theta}(i)$, can be determined for each programmed point, i, in the same manner, and thus the calculated joint values will satisfy the required welding constraints.

Control of the torch speed in the part frame is an important concern for welding. A typical requirement is to keep the speed of the torch tip along the welding trajectory at a specified, constant value. It should be noted that in a multiple manipulator system (e.g., one that consists of a welding robot and a part positioner) the welding speed is not the same as the speed programmed on the robot. The former is specified in the part frame while the latter is measured with respect to the world frame. A speed of 2 mm/s along the weld seam may be obtained by a stationary torch (in the world frame) while the positioner rotates the workpiece to achieve the desired welding speed. Another scenario might consist of a robot

moving the torch with a speed of 10 mm/s in the world coordinates and the positioner moving the workpiece so that the weld pool moves in the same direction at 8 mm/s, still resulting in a net speed of 2 mm/s in the part frame. Because the actual movements of the positioner and the robot are not trivially defined after the downhand welding algorithms have redefined the joint angles, certain measures have to be taken to ensure controlled welding speed.

The original joint angles of the system are transformed by the welding constraint algorithms into a new set of angles for each of the programmed points. As a result, the new $\underline{T}_R(i)$ and $\underline{T}_P(i)$ can be found for each programmed teach point along the weld. Assume that the welding speed (i.e., torch speed in the part frame) is specified and assumed to be constant for each segment i, connecting teach points i-1 and i. These desired speeds in the part frame can be designated by the scalars $V_P(i)$ which may vary from one segment to another. For constant speed throughout the entire welding pass, all $V_P(i)$ are equal. Now, recall that the programmed teach points in the part frame are defined by a time-varying matrix P(i) which is of the following form:

$$\underline{P}(i) = \begin{bmatrix} n_{P,x} & o_{P,x} & a_{P,x} & p_{P,x} \\ n_{P,y} & o_{P,y} & a_{P,y} & p_{P,y} \\ n_{P,z} & o_{P,z} & a_{P,z} & p_{P,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Specifically, the position of each programmed point in the part frame is specified by the fourth column of this matrix, and thus the distance between any adjacent programmed points is readily found. The time required to traverse a given segment i, T(i), is found as the part frame distance between points i-1 and i, divided by the welding speed required for segment i:

$$T(i) = \sqrt{\{ [p_{P,x}(i)-p_{P,x}(i-1)]^2 + [p_{P,y}(i)-p_{P,y}(i-1)]^2 + [p_{P,z}(i)-p_{P,z}(i-1)]^2 \}} / V_P(i) \quad (15)$$

Now, recall that the position and orientation of the torch tip in world coordinates is given by $\underline{Z}_R \, \underline{T}_R(i) \, \underline{E}$. This results in a 4-by-4 matrix of which the fourth column is the world frame position of the end effector coordinates:

$$\underline{Z}_R \, \underline{T}_R(i) \, \underline{E} = \begin{bmatrix} n_{R,x} & o_{R,x} & a_{R,x} & p_{R,x} \\ n_{R,y} & o_{R,y} & a_{R,y} & p_{R,y} \\ n_{R,z} & o_{R,z} & a_{R,z} & p_{R,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

By calculating the segment lengths or distances between successive $\underline{p}$-vectors from this equation and dividing them by the time lengths required for each segment, the desired speed of the torch in the world frame, $V_R(i)$, is obtained:

$$V_R(i) = \sqrt{\{ [p_{R,x}(i) - p_{R,x}(i-1)]^2 + [p_{R,y}(i) - p_{R,y}(i-1)]^2 + [p_{R,z}(i) - p_{R,z}(i-1)]^2 \}} / T(i) \quad (17)$$

A more compact form for $V_R(i)$ is

$$V_R(i) = V_P(i) \, \frac{\| \underline{p}_R(i) - \underline{p}_R(i-1) \|}{\| \underline{p}_P(i) - \underline{p}_P(i-1) \|} \quad (18)$$

which yields the required speed of the end effector in world coordinates for each segment i. These values can be downloaded with the teach points to the actual robot. Note that even when the same welding speed is required throughout all programmed segments (i.e., $V_P(i)$ is constant for all i) the torch speed $V_R(i)$, in the

world frame, will in general vary due to the interactions between the robot and the part positioner.

PROGRAMMING FOR VERTICAL-UP WELDING

The preceding section has summarized the general approach of the programming algorithm. Further details, using downhand welding constraints as an example, were given in the original NASA Technical Paper by Fernandez and Cook [1]. In this section the specific details, required to accomplish vertical-up welding, are given.

A new orientation of the workpiece is the only change required to convert the robotic programming from downhand welding to vertical-up welding. Torch position and orientation relative to the workpiece, wirefeed orientation, and speed control are all unchanged.

Referring to the original presentation of the robotic path programming algorithm, the workpiece orientation requirements were demonstrated for downhand welding. In that example, a vector $\underline{n}_g(i)$, perpendicular to the workpiece surface at each programmed point i, was determined. This vector was oriented "into" the workpiece, i.e., in the same general direction as the arc plasma flow. The forcing function, $\underline{f2}_{g\text{-}err}$, was defined as:

$$\underline{f2}_{g\text{-}err,downhand} = \underline{R}\,([\underline{Z}_P\,\underline{A}_{1P}\,\underline{A}_{2P}\,\underline{G}\,\underline{n}_g(i)] \times g) \tag{19}$$

and it was reduced to zero through iterative adjustments of the two positioner joint variables. In essence, this forcing function was the cross product of the vector normal to the workpiece surface and the gravity vector, g. Clearly, this cross

product becomes zero when the gravity vector and the normal vector are parallel, i.e., when the tangent plane to the workpiece surface is horizontal.

To achieve vertical-up workpiece orientation, only a minor modification of the algorithm suffices. The basic approach is simply to replace the normal vector to the workpiece surface with the tangent to the weld seam. Two approaches to obtaining the tangent vector are mentioned here. The first one is to use the normalized secant vector, connecting teach points $\underline{p}(i-1)$ and $\underline{p}(i)$, as an approximation to the tangent:

$$\underline{n}_1(i) = \frac{\underline{p}(i) - \underline{p}(i-1)}{\| \underline{p}(i) - \underline{p}(i-1) \|} \tag{20}$$

The second approach is to calculate the tangent using the coordinate vector frame assigned to the torch. This was the method used to implement the graphical simulations illustrated in this paper. Assume that the convention of Fernandez and Cook [1] is maintained where the "approach" vector ($\underline{a}$) of the torch is aligned along the arc plasma flow. The "orientation" vector ($\underline{o}$) is tangent to the workpiece surface, but it is perpendicular to the travel direction and is pointed to the right with respect to the traveling torch. Finally, the "normal" vector ($\underline{n}$) is perpendicular to these two, so that:

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

If the arc plasma flow is perpendicular to the workpiece surface, the normal vector, $\underline{n}$, can serve as the tangent to the workpiece. If the electrode is to be tilted from the perpendicular orientation, as may sometimes be the case, the coordinates from $\underline{n}$ must be saved for the tangent vector before such tilting is implemented:

$$\underline{n}_2(i) = \underline{n}(i)_{\text{electrode perpendicular to part surface}} \tag{22}$$

Therefore, either $\underline{n}_1$ or $\underline{n}_2$ may be used for the tangent vector, $\underline{n}_t$. In either case the forcing function for vertical-up welding becomes:

$$\underline{f2}_{g\text{-}err,vert\text{-}up} = -\underline{R}\,([\underline{Z}_P\,\underline{A}_{1P}\,\underline{A}_{2P}\,\underline{G}\,\underline{n}_t(i)] \times \underline{g}) \tag{23}$$

i.e., $\underline{n}_g(i)$ from the downhand welding algorithm is simply replaced by $\underline{n}_t(i)$ to implement vertical-up welding. Furthermore, the minus sign arises from the fact that the vector $\underline{n}_t(i)$ converges to point up, opposite to the gravity vector which points down. The truncation matrix, $\underline{R}$, is defined as:

$$\underline{R} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{24}$$

As previously, by augmenting the torch position error function and the vertical-up error function, a single error forcing function, $\underline{f8}_{t\text{-}err}$, for torch position and vertical-up welding control is obtained:

$$\underline{f8}_{t\text{-}err} = \begin{bmatrix} \underline{f6}_{p\text{-}err} \\ \underline{f2}_{g\text{-}err} \end{bmatrix} \tag{25}$$

and this error vector becomes $\underline{0}$ when the torch is properly positioned and aligned with respect to the workpiece and when the workpiece is properly aligned with respect to the workcell gravity vector. In the same fashion as before, the $\underline{\theta}$ vector, which consists of all joint angles in this fully determined system, can be determined by Newton's iteration method:

$$\underline{\theta}_{j+1} = \underline{\theta}_j - h \underline{J}^{-1} \underline{f8}_{t\text{-}err}(\underline{\theta}_j) \quad (26)$$

$$[\underline{f8}_{t\text{-}err}(\underline{\theta})]^T \underline{K} [\underline{f8}_{t\text{-}err}(\underline{\theta})] < \epsilon \quad (27)$$

The simple modification of $\underline{f2}_{g\text{-}err}$ to $\underline{f2}_{h\text{-}err}$ is all that is needed to convert the algorithm from downhand welding to vertical-up welding. Therefore, the remaining sections in the algorithm are unchanged.

GRAPHIC SIMULATION RESULTS

To verify and demonstrate the accuracy of the modified algorithm, computer simulations of vertical-up welding were implemented. The modeled robot is the Cincinnati-Milacron T^3-776, which has six revolute joints. The positioner is an Advanced Robotics RP-25. It has two revolute joints, one revolving the mounting plate in its plane and one to tilt it off the horizontal orientation. In addition to these two DOF, the positioner height can be indexed off-line, but the index parameter is usually assumed fixed for any given application. For the simulations ROBOSIM, a graphic simulation package developed by Fernandez [4], was employed in conjunction with FORTRAN programs, used for implementation of the robot weld path programming algorithm. These simulations were run on a Hewlett-Packard 9000/350-SRX workstation.

The workpiece used for the vertical-up welding simulation consists essentially of two cylinder sections joined by the weld. It will be used in the planned space station berthing port. A weld of this type is frequently referred to as a saddle weld. Sequential images of the saddle welding simulation, after the weld programming algorithm has been applied, are shown in figures 1(a) and (b). Figure 1(a) shows the simulation from a viewpoint above the robotic manipulator, while Figure 1(b) is

viewed along the horizontal plane. Figure 2 shows simulation of a saddle weld which is performed on the space shuttle main engine. Although this particular weld is usually carried out in a downhand orientation using the GTAW process, it was simulated here to more clearly illustrate the vertical-up weld programming algorithm. This example illustrates clearly how the workpiece and the torch are oriented with respect to each other and, with respect to the workcell, throughout the entire weld. Close examination of these simulations reveal that the vertical-up welding requirements are met.

DISCUSSION AND CONCLUSIONS

A generalized robotic programming algorithm, applied to vertical-up welding, has been demonstrated. The fundamental algorithm has been published by Fernandez and Cook in an earlier NASA technical paper, and necessary modifications and simulations for vertical-up welding have been presented here. The vertical-up welding method is particularly important for VPPAW, which is extensively used in NASA applications. It has been demonstrated that adaptation of the programming algorithm to vertical-up welding requires only minor modifications.

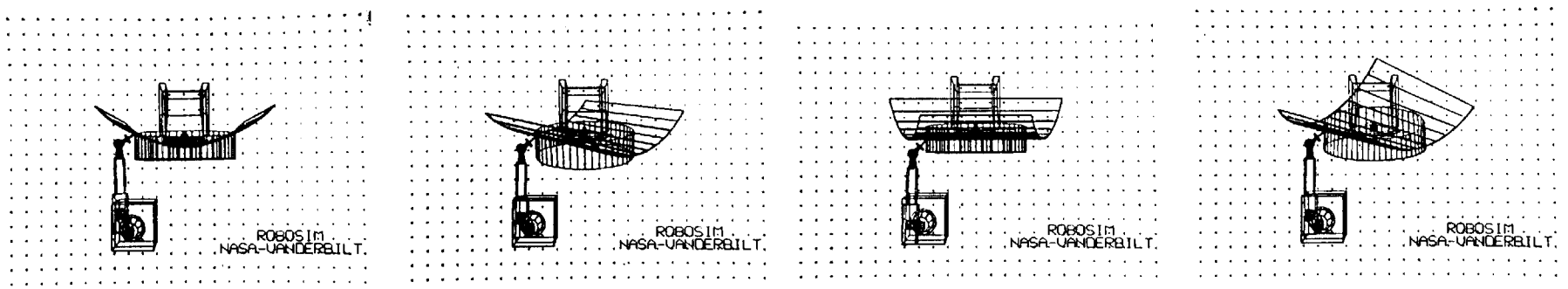
REFERENCES

[1] Fernandez, K.R., and Cook, G.E.: "A Generalized Method for Automatic Downhand and Wirefeed Control of a Welding Robot and Positioner." NASA Technical Paper 2807, February 1988.

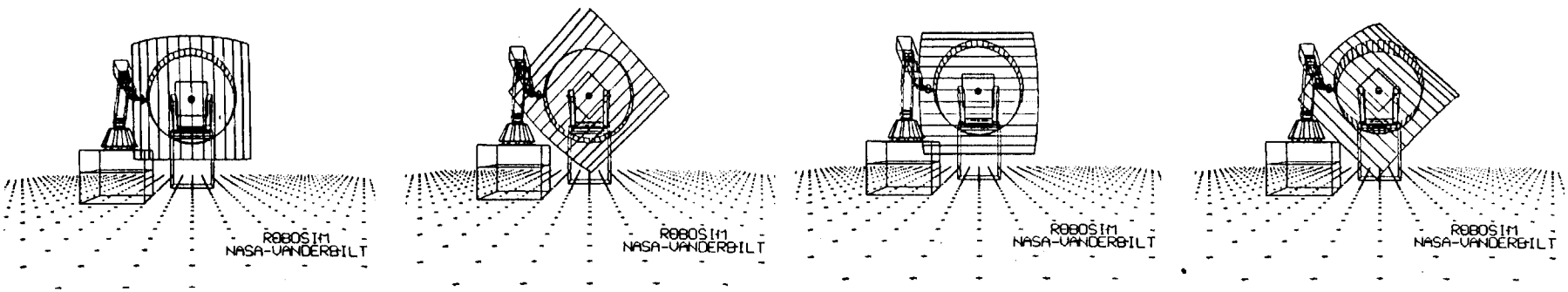
[2] Andersen, K.A., Barnett, R.J., Cook, G.E., and Zein-Sabattou, S.: "Robotic Weld Path Programming." SBIR Phase I Final Report, Mid-South Engineering, Nashville, Tennessee, August 1988.

[3] Paul, R.P.: "Robot Manipulators: Mathematics, Programming, and Control." The MIT Press, 1981.

[4] Fernandez, K.R.: "Robotic Simulation and a Method for Jacobian Control of a Redundant Mechanism with Imbedded Constraints." Ph.D. Dissertation, Vanderbilt University, May 1988.



(a)



(b)

Figure 1. Simulation of saddle welding.

Viewed from above (a) and from the front of the robot (b).

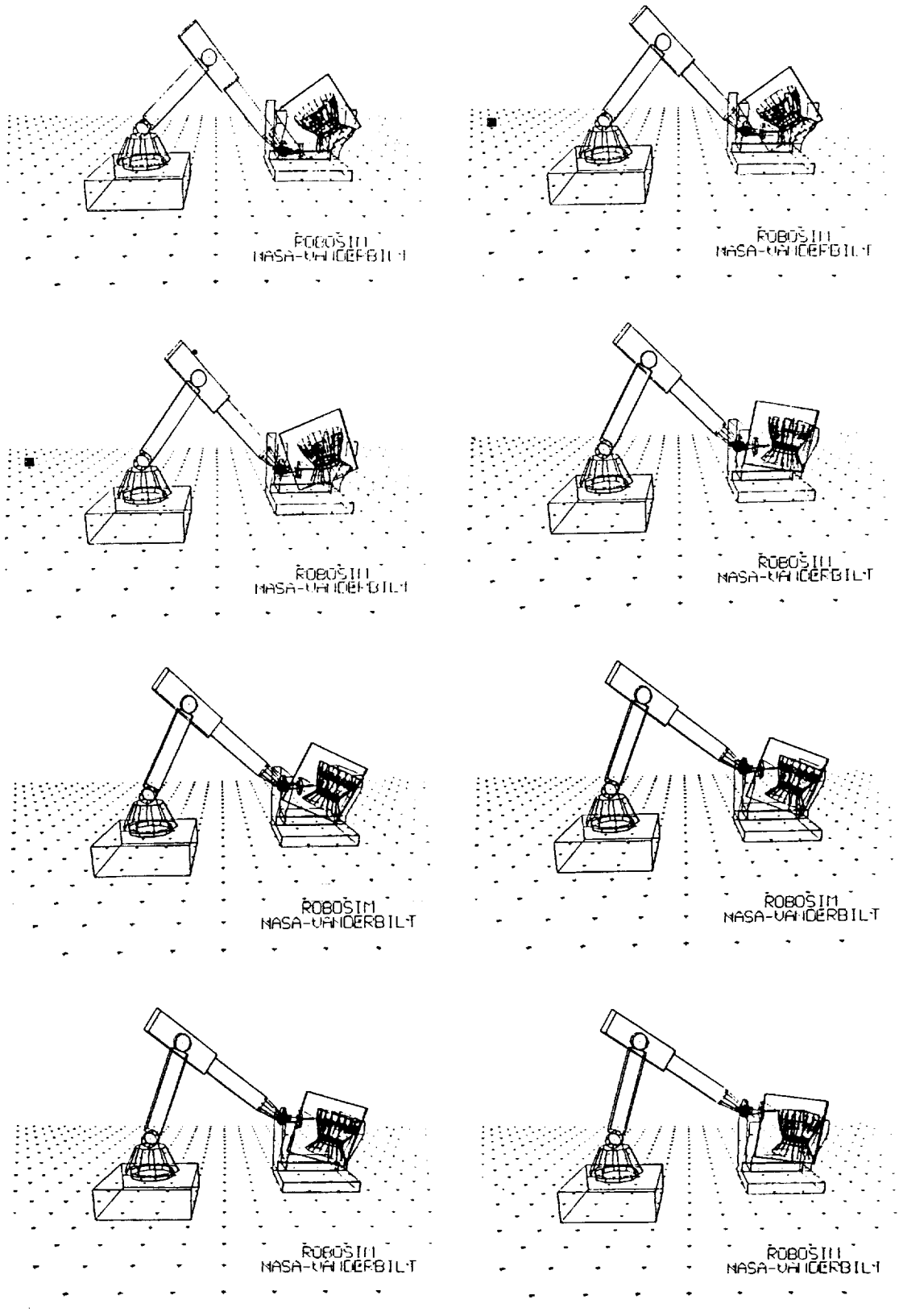


Figure 2. Nozzle welding.

This example simulates a weld on the space shuttle main engine.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1991	Technical Paper

4. TITLE AND SUBTITLE
A Generalized Method for Multiple Robotic Manipulator Programming Applied to Vertical-Up Welding

5. FUNDING NUMBERS

6. AUTHOR(S)
Kenneth R. Fernandez, George E. Cook*, Kristinn Andersen*, Robert Joel Barnett*, Saleh Zein-Sabattou*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

8. PERFORMING ORGANIZATION REPORT NUMBER
M-672

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TP-3163

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

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category: 31

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper discribes the application of a weld programming algorithm for vertical-up welding, which is frequently desired for variable polarity plasma arc welding (VPPAW). The Basic algorithm performs three tasks simultaneously: control of the robotic mechanism so that proper torch motion is achieved while minimizing the sum-of-squares of joint displacement; control of the torch while the part is maintained in a desired orientation; and control of the wire feed mechanism location with respect to the moving welding torch. This algorithm has been presented and demonstrated in earlier reports as applied to downhand welding, such as is required for gas tungsten arc welding (GTAW).

This paper also presents a modification of this algorithm which permits it to be used for vertical-up welding. The details of this modification are discussed and simulation examples are provided for illustration and verification.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Robotic Welding, Off-line programming, CAD/CAM/CIM, Vertical-Up position, ROBOSIM	32
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

NASA-Langley, 1991