15

Kinematic Control of A Robot-Positioner System for Arc Welding Application

Anatoly P. Pashkevich [1,2], Alexandre B. Dolgui [1]
[1]*Ecole de Mines de Saint Etienne*
France
[2]*Belarusian State University of Informatics and Radioelectronics*
Belarus

Welding is one of the most successful applications for industrial robots, which encourages intensive research and development in the area of the CAD-based and off-line programming (Pires et al., 2003, Yagi, 2004). At present, relevant software tools allow to optimise the process parameters, which directly influence the product quality and the system working cycle. Besides, they enable users to complete most of the process preparation actions in advance, without access to the workcell, and therefore, to make the robotic systems competitive for both large and small series, or even for unique products (Ranky, 2004). However, resent advances in the arc welding technology motivate rethinking of some postulates and conventions incorporated in the existing off-line programming methods. One of related problems, the kinematic control of a robot-positioner system, is addressed in this chapter.

The welding position (or, more precisely, the weld joint orientation relative to gravity) is an essential issue in both manual and robotic welding, associated with the problem of the weld puddle control. As is known from long-term experience, the highest quality and productivity are achieved for the downhand (or flat) welding position, where the workpiece is oriented so that the weld tangent line is horizontal, and the weld normal vector is the opposite of the gravity direction (Cary 1995). This orientation is preferable because gravity draws the molten metal downward into the joint allowing it to flow appropriately along the weld contour, which makes the welding faster and easier. For this reason, the downhand welding has been adopted in robotics as a de facto standard (Bolmsjo 1987, Fernandez and Cook 1988).

To ensure the desired weld orientation, a typical robotic arc welding station (Fig. 1) includes two separate moving mechanisms: (i) a five- or six-axis industrial robot (welding tool manipulator) aimed at moving the weld touch with the required speed and orientation relative to the weld joint; and (ii) a two- or three-axis positioning table (workpiece manipulator), which ensures the downhand (or close to it) orientation of the weld joint with respect to gravity.

In contrast to the robot, a skilled human-welder is capable to perform such operations in other positions, such as the horizontal, vertical, or overhead ones. To make such type of welding easier, several companies recently proposed their technological innovations, the flux-cored wires, that create a protective coating, supporting the metal against gravity. This

Source: Industrial Robotics: Programming, Simulation and Applicationl, ISBN 3-86611-286-6, pp. 702, ARS/plV, Germany, December 2006, Edited by: Low Kin Huat

makes it possible to enlarge area of the non-downhand welding and employ for this industrial robots (Tolinski 2001). Besides, recent advances in computer vision have allowed enhancements in robot ability for the welding puddle control (Tarn et al. 2004). Hence, the existing kinematic control techniques for the arc welding robots must be revised in order to relax the downhand constraint.

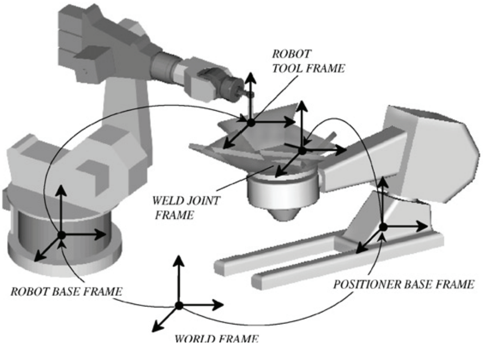


Fig. 1. Robotic arc welding station and its components.

2. Related Works

Automatic programming for robotic arc welding incorporates a number of nontrivial steps, ranging from specifying a manufacturing task to planning the robot-positioner motions (Kim et al. 1998, Bolmsjo 2002). They include, in particular, the welding feature extraction from CAD data, welding task planning and sequencing, coordinated robot-positioner control and numerous implementation issues (interfacing with the operator/controller, weldseam tracking, workcell calibration, etc.).

Since the beginning of robotic arc welding, many related studies focused on the kinematic control of the robot-positioner system. Theoretically, the arc welding requires (5+2)-axis manipulation, needed for the positioning/orienteering of the weld torch and the weld joint, respectively. However, because a standard industrial robot has 6 degrees-of-freedom, relevant papers cover topics from (6+2)- to (7+3)-axis kinematical structures. Accordingly, in all cases, the kinematic control problem is solved by imposing the following task-specific constraints:

(i) five constraints on the torch position and orientation, defined by three Cartesian coordinates of the welding tip and two Euler angles of the plasma flow-line;

(ii) two constraints on the weld joint orientation relative to gravity, defined as the verticality of the weld normal line (‘downhand welding’).

It is implicitly assumed here that both the torch rotation about the plasma flowline and the weld joint rotation about the vertical line are irrelevant to the welding process.

For the typical (6+2)-case, corresponding to a six-axis robot and two-axis balance or till-roll positioner, the problem has been studied by several authors (Bolmsjo 1987, Fernandez and Cook 1988, Nikoleris 1990, Kim et al. 1998). The common approach is based on the strictly downhand solution for the positioner inverse kinematics and the augmented solutions for the robot inverse kinematics, which depends on an arbitrary scalar parameter. Then, this free parameter is used for singularity or collision avoidance, optimisation of the manipulability, increase in robotic tool reach, etc. These results were extended to the (7+2)-

case by Ahmad and Luo (1989) whose study focused on a six-axis robot mounted on rail and a two-axis positioner; they also used the extra degrees of freedom to keep the robot out of singular configurations and to increase its reach. Recently, Wu et al. (2000) applied the genetic algorithm technique to solve a similar (7+3)-problem.

The positioner inverse kinematics, incorporated in the above control methods, was solved mainly for the strictly downhand weld orientation with respect to gravity. This essentially simplified the analytical expressions but was in certain disagreement with engineering practice that admits some tolerances. Hence, several authors considered more general formulations. In particular, Bolmsjo (1987) and Nikoleris (1990) stated the problem as aligning of any weld-associated and any given gravity-associated vectors. For this case, numerical and analytical solutions were obtained for both the roll-tilt and balance positioners. Later, the problem was reformulated by the authors (Pashkevich et al., 2003) and solved in terms of the weld slope-roll angles; the least-square solutions were also obtained for the case when exact ones do not exist.

In spite of common understanding, only Kim et al. (1998) have directly addressed the issue of the downhand-orientation tolerancing and its relation with the weld quality. These authors introduced the weld 'stability' metrics, thus allowing admissible orientations for the welding tool and welding joint (defined by work/ travel and slope/rotation angles, respectively) to be computed. The open question, however, is assigning reasonable 'stability limits' to obtain the required quality, which obviously needs a welding expert. Besides, no judgements on variations of the welding speed within the stability region have been proposed.

Another active area of the related research is the arc welding operations planning and sequencing. These works concentrate on specific non-trivial cases of the travelling-salesman problem (TSP) known from combinatorial optimisation (see Gutin and Punnen, 2002 for details and latest results). For arc welding, the scheduling problem with the minimum-time objective was first addressed by Rubinovitz and Wysk (1988), who suggested a heuristic algorithm based on the classical TSP nearest-neighbour method. Then, Fukuda and Yoshikawa (1990) applied to this problem a neural network technique focusing on reduction in the welding distortions. Later, the TSP-based method was enhanced by Kim et al. (1998, 2002a), who proposed several heuristics and genetic algorithms, which avoid the distortions caused by heat by imposing the problem-specific non-precedence constraints. In another paper, Kim et al. (2002b) reformulated the heat-affected zone constraint by setting the lower bound on the travel time between two consecutive welding operations ('cooling time') and proposed several heuristics based on the nearest-neighbour, 2-opt, 3-opt, and tabu search methods. Grenestedt (2003) applied to this problem the simulating annealing in combination with the approximate analytical distortion models. In the latest paper by Kim et al. (2005), there several enhanced welding sequencing heuristics were proposed, which also adopt the 'cooling time' concept.

This chapter summarises the authors' results in the area of the robotic arc welding (Pashkevich et al., 2003; Dolgui and Pashkevich, 2006) and presents techniques for both the closed-form inverse kinematics solutions and optimal planning of the welding sequences. It contributes to the research originated from the papers of Bolmsjo (1996), Nikoleris (1990) and Kim et al. (1998), assuming that the downhand constraint is relaxed and implicitly converted into the penalty function, which increases the welding time depending on the 'non-downhand' degree. The objective is to minimize the overall manufacturing cycle, by finding a reasonable trade-off between the positioner motion time and the time required for the welding itself.

3. Kinematic Control Architecture

3.1 Control Hierarchy

In contrast to the early robotic manipulators, in which capabilities were limited by the servo-control of separate joint axes, the modern industrial robotic systems should implement the task-level control that essentially simplifies the manufacturing task definition for the end user. It results in including a kinematic control module as a built-in part of the hierarchical control system, where the high-level command is sequentially decomposed to the lower level ones, up to the axis drives and the process variable controllers. However, in spite of the apparent simplicity, defining of a particular content of each control level requires development of specific mathematical methods that take into account particularities of the relevant technology.

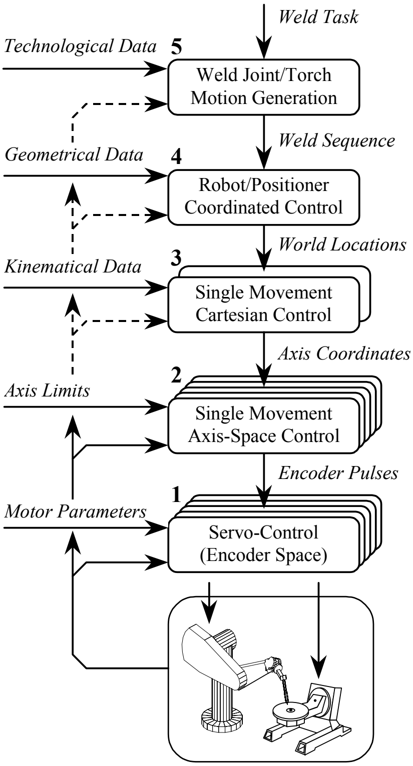

Weld Task
Technological Data
5
Weld Joint/Torch Motion Generation
Weld Sequence
Geometrical Data
4
Robot/Positioner Coordinated Control
World Locations
Kinematical Data
3
Single Movement Cartesian Control
Axis Coordinates
Axis Limits
2
Single Movement Axis-Space Control
Encoder Pulses
Motor Parameters
1
Servo-Control (Encoder Space)

Fig. 2. Multi-level control hierarchy.

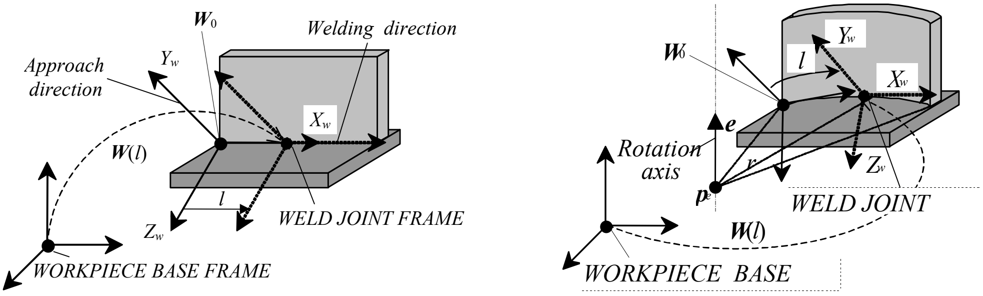
For the robotic arc welding, five levels of control are typically used (Fig. 2). The highest of them highly relies on the kinematic modelling and deals with obtaining the optimal technological and geometrical parameters, such as the orientation angles of the weld joint and the welding gun, the weld sequence, the weld speed, etc. The fourth level performs appropriate coordinate transformations via the direct/inverse kinematics of all mechanical components (robot, positioner, gantry). The remaining three levels deal with the implementing of tool/workpiece movements in the Cartesian space, in the manipulator axis space and, finally, in the motor shaft space. (It should be noted that for some manipulators, the interrelation between the joint axis angle and the motor shaft angle is non-trivial.) At present, control techniques for all the abovementioned hierarchical levels are being intensively developed. For instance, advanced commercial controllers already include the forward dynamic models, which essentially improve the operational speed and accuracy. However, various aspects of the fourth and the fifth control levels are still subject of intensive research.

3.2. Kinematic Description of the Welds

The spatial location of the welding object, as a general rigid body, can be defined by a single frame that incorporates six independent parameters (three Cartesian coordinates and three Euler angles). However, defining geometry of each weld requires some additional efforts, depending on the joint profile. Since capabilities of modern industrial robotic systems allow processing two basic types of the contours (linear and circular), only these cases are considered below.

For the linear joints, a moving frame with the specific definition of the axes can describe the weld geometry. In this chapter, this frame is defined so that (Fig. 3):

- The X_w- axis is directed along the weld joint (welding direction);
- The Y_w-axis is normal to the weld joint (weld torch approaching direction);
- The Z_w-axis completes the right-hand oriented frame ("weaving" direction).



(*a*) (*b*)

Fig. 3. Definition of the weld frames for a liner (*a*) and circular (*b*) welds.

It should be noted that, in practice, it is prudent to define the Y_w -axis as the bisectrix of the corresponding weld joint surfaces.

Taking into account the above definitions, the kinematic model of the linear weld relative to the *WB*-frame (i.e., the workpiece base frame, see Fig. 3a) can be described by the following homogenous parametric equation

$$^{WB}\boldsymbol{W}(l) = \begin{bmatrix} \boldsymbol{n}_w^s & \boldsymbol{s}_w^s & \boldsymbol{n}_w^s \times \boldsymbol{s}_w^s & \boldsymbol{p}_w^s + l \cdot \boldsymbol{n}_w^s \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times 4}, \qquad (1)$$

where the parameter l is the welding torch displacement, the left superscript "*WB*" refers to the workpiece base coordinate system, the right superscript "*s*" and the subscript "*w*" denote starting point of the weld, $\boldsymbol{n}_w^s$ is the unit vector of the welding direction (axis X_w), $\boldsymbol{s}_w^s$ is the unit vector of the approaching direction (axis Y_w), and $\boldsymbol{p}_w^s$ is the position vector of the weld starting point. It should be stressed that the vectors $\boldsymbol{n}_w^s$; $\boldsymbol{s}_w^s$; $\boldsymbol{p}_w^s$ are defined relative to the *WB*-frame and, in practice, they are easily derived from the workpiece 3D CAD model.

For the circular joints, a similar approach is used, but the moving frame is computed to ensure the tangency of the welding path and the X_w-axis at every point (Fig. 3b). It is evident that the initial frame is subject to the rotational transformation and the weld kinematics is described by the following parametric equation:

$$^{WB}\boldsymbol{W}(l)=\begin{bmatrix}\boldsymbol{R}_e(l/r)\cdot\boldsymbol{R}_w^e & \boldsymbol{R}_e(l/r)\cdot(\boldsymbol{p}_w^e-\boldsymbol{p}_e)+\boldsymbol{p}_e\\ \boldsymbol{0}_{1\times3} & 1\end{bmatrix}_{4\times4}, \quad (2)$$

where the parameter *l* and the sub/superscripts "*WB*", "*w*", "*s*" have the same meaning as in (1), the orthogonal 3×3 matrix is expressed as $\boldsymbol{R}_e=[\boldsymbol{n}_w^s\;\;\boldsymbol{s}_w^s\;\;\boldsymbol{n}_w^s\times\boldsymbol{s}_w^s]$ and defines the orientation of the weld frame at the starting point, *r* is the radius of the circular welding joint, $\vartheta=l/r$ is the angle of rotation, the vector $\boldsymbol{p}_e$ defines the position of the circle centre, and $\boldsymbol{R}_e(\vartheta)$ the general rotation matrix (Fu et al., 1987) around the axis, which is determined by the unit vector $\boldsymbol{e}=[e_x,e_y,e_z]$ (see Fig. 4):

$$\boldsymbol{R}_e(\vartheta)=\begin{bmatrix}e_x^2V_\vartheta+C_\vartheta & e_xe_yV_\vartheta-e_zS_\vartheta & e_xe_zV_\vartheta+e_yS_\vartheta\\ e_xe_yV_\vartheta+e_zS_\vartheta & e_y^2V_\vartheta+C_\vartheta & e_ye_zV_\vartheta-e_xS_\vartheta\\ e_xe_zV_\vartheta-e_yS_\vartheta & e_ye_zV_\vartheta+e_xS_\vartheta & e_z^2V_\vartheta+C_\vartheta\end{bmatrix}_{3\times3} \quad (3)$$

As in the previous case, the required vectors $\boldsymbol{n}_w^s$; $\boldsymbol{s}_w^s$; $\boldsymbol{p}_w^s$ and $\boldsymbol{e}$, $\boldsymbol{p}_e$ as well as the radius *r*; may also be easily derived from the workpiece 3D model using capabilities of the modern graphical simulation systems to generate straight lines, planes and circles.

Thereby, expressions (1)–(3) completely define spatial location (i.e., the position and the orientation) of the weld joint relative to the *WB*-frame (workpiece base), which should be adjusted by the positioner to optimise the weld orientation relative to the gravity (see Fig. 1). Hence, the absolute (world) location of the weld joint is described by the product of the homogenous matrices

$$^{0}\boldsymbol{W}(l)=\left[{}^{0}\boldsymbol{T}_{PB}\cdot\boldsymbol{P}(\boldsymbol{q})\cdot{}^{PF}\,\boldsymbol{T}_{WB}\right]\cdot{}^{WB}\boldsymbol{W}(l) \quad (4)$$

where the left superscript "0" refers to the world coordinate system, the matrix $^0\boldsymbol{T}_{PB}$ defines the absolute (world) location of the positioner base *PB*; the matrix $^{PF}\boldsymbol{T}_{WB}$ describes the workpiece base *WB* location relative to the positioner mounting flange *PF*; and the matrix function $\boldsymbol{P}(\boldsymbol{q})$ is the positioner direct kinematic model, while $\boldsymbol{q}$ is the vector of the positioner joint coordinates.

To ensure good product quality and to increase the welding speed, the weld joint should be properly oriented relative to the gravity. The exact interrelations between these parameters are not sufficiently well known and require empirical study in each particular case. But practising engineers have developed a rather simple rule of thumb that is widely used for both the online and off-line programming: "*the weld should be oriented in the horizontal plane so that the welding torch is vertical, if possible*" (Bolmsjo, 1987). It is obvious that the CAD-based approach requires numerical measures of the "horizontality" and the "verticality", which are proposed below.

Let us assume that the Z_0-axis of the world coordinate system is strictly vertical (i.e. directed opposite to the gravity vector), and, consequently, the X_0Y_0-plane is horizontal. Then, the weld orientation relative to the vector of gravity can be completely defined by two angles (Fig. 4):

- The weld slope $\theta \in [-\pi/2; \pi/2]$, i.e. the angle between the vector of the welding direction ${}^0\boldsymbol{n}_w$ and the Cartesian plane X_0Y_0;
- The weld roll $\xi \in (-\pi; \pi]$, i.e. the angle between the vector of the approaching direction ${}^0\boldsymbol{s}_w$ and the vertical plane that is parallel to the vector ${}^0\boldsymbol{n}_w$ and the Cartesian axis Z_0.

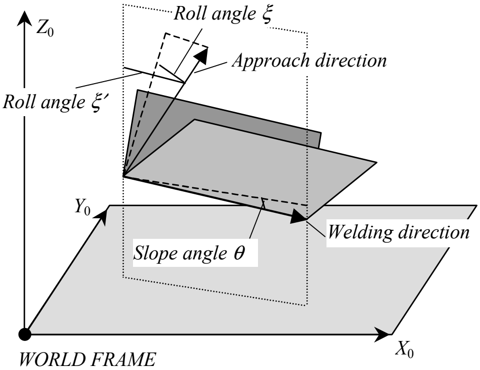


Fig. 4. Definition of the weld orientation angles.

The numerical expressions for θ, ξ can be obtained directly from the definition of the *RPY*-angles (Fu et al., 1987), taking into account that the weld orientation (θ, ξ) = (0, 0) corresponds to the horizontal direction of the axis X_w and the vertical direction of the Y_w (see Fig. 5):

$$ {}^0\boldsymbol{W}_R = \boldsymbol{R}_z(\psi)\cdot \boldsymbol{R}_y(\theta)\cdot \boldsymbol{R}_x(\pi/2-\xi) \tag{5}$$

where ${}^0\boldsymbol{W}_R$ is the 3 × 3 orientation submatrix of the 4 × 4 matrix of the weld location; $\boldsymbol{R}_x$; $\boldsymbol{R}_y$; $\boldsymbol{R}_z$ are the 3 × 3 rotation matrices around the axes *X*; *Y*; *Z*; respectively, and ψ is the yaw angle which is non-essential for the considered problem. Multiplication of these matrices leads to

$$ {}^0\boldsymbol{W}_R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta C_\xi - S_\psi S_\xi & C_\psi S_\theta S_\xi + S_\psi C_\xi \\ S_\psi C_\theta & S_\psi S_\theta C_\xi + C_\psi S_\xi & S_\psi S_\theta S_\xi - C_\psi C_\xi \\ -S_\theta & C_\theta C_\xi & C_\theta S_\xi \end{bmatrix} $$

where *C* and *S* denote respectively *cos* (.) and *sin*(.) of the corresponding angle specified at the subscript. Therefore, the weld joint orientation angles θ, ξ can be derived as follows:

$$ \theta = atan2 \frac{-{}^0n_w^z}{\sqrt{\left({}^0s_w^z\right)^2 + \left({}^0a_w^z\right)^2}}; \quad \xi = atan2 \frac{{}^0a_w^z}{{}^0s_w^z} \tag{6}$$

where ${}^0\boldsymbol{n}_w$, ${}^0\boldsymbol{s}_w$, ${}^0\boldsymbol{a}_w$ are the corresponding column vectors of the orthogonal matrix ${}^0\boldsymbol{W}_R$. Taking into account interrelations between these vectors, the angles θ, ξ can be finally expressed as functions of the weld joint direction ${}^0\boldsymbol{n}_w$ and approaching direction ${}^0\boldsymbol{s}_w$

$$ \theta = atan2 \frac{-{}^0n_w^z}{\sqrt{\left({}^0n_w^x\right)^2 + \left({}^0n_w^y\right)^2}}; \tag{7}$$

$$\xi = atan2\frac{{}^0n_w^x \cdot {}^0s_w^y - {}^0n_w^y \cdot {}^0s_w^x}{{}^0s_w^z} \tag{8}$$

It should be noted that it is possible to introduce *alternative* definition of the *weld roll*, which is non-singular for all values of the weld slope. It is $\xi' \in [0; \pi]$, which is the angle between the approaching direction ${}^0\mathbf{s}_w$ and the vertical axis Z_0 (see Fig. 4):

$$\xi' = \mathrm{a}\tan 2\frac{\sqrt{\left({}^0s_w^x\right)^2 + \left({}^0s_w^y\right)^2}}{{}^0s_w^z} \tag{9}$$

As in the case of angles (θ, ξ), the description (θ, ξ′) also defines the 3rd row of the weld joint orientation matrix ${}^0\boldsymbol{W}_R$, but the sign of the $\boldsymbol{a}_w^z$ may be chosen arbitrary. Hence, the interrelation between both the definitions of the roll angle ξ and ξ′ is given by the equation

$$cos(\theta)\cdot cos(\xi) = cos(\xi'), \tag{10}$$

and both (θ, ξ) and (θ, ξ′) can be used equally.

4. Weld Joint Orienting Problems

In the robotic welding station, the desired orientation of the weld relative to the gravity is achieved by means of the positioner, which adjusts the slope and the roll angles by alternating its axis coordinates. Using the kinematic model (4) and the definitions from the previous section, the problems of the welding joint orientation can be stated as follows.

Direct Problem. *For given values of the positioner* axis *coordinates* $\boldsymbol{q}$, *as well as known homogenous transformation matrices* ${}^0\boldsymbol{T}_{PB}$, ${}^{PF}\boldsymbol{T}_{WB}$ *and the weld frame location relative to the object base* $\boldsymbol{W}$, *find the weld frame orientation in the world coordinate system* ${}^0\mathbf{W}$ *and the slope/roll orientation angles* (θ, ξ).

Inverse Problem 1. *For given values of* the *slope/roll orientation angles* (θ, ξ), *as well as known homogenous transformation matrices* ${}^0\boldsymbol{T}_{PB}$, ${}^{PF}\boldsymbol{T}_{WB}$ *and the weld frame location relative to the object base* $\boldsymbol{W}$, *find the values of the positioner* axis *coordinates* $\boldsymbol{q}$.

There is also another version of the inverse problem for the welding positioner (Nikoleris, 1990) that deals with a reduced version of the expression (4), which describes only a single unit vector transformation

$${}^0\mathbf{s}_w = \left[{}^0\mathbf{T}_{PB}\cdot\mathbf{P}(\mathbf{q})\cdot{}^{PF}\mathbf{T}_{WB}\right]_{3\times3}\cdot\mathbf{s}_w{}' \tag{11}$$

Using the accepted notations, this formulation can be stated as follows:

Inverse Problem 2. *For given values of* the *world coordinates of the weld approach vector* ${}^0\mathbf{s}_w$, *as well as for known homogenous transformation matrices* ${}^0\boldsymbol{T}_{PB}$, ${}^{PF}\boldsymbol{T}_{WB}$ *and the normal vector orientation relative to the object base* $\mathbf{s}_w$, *find the values of the positioner* axis *coordinates* $\boldsymbol{q}$.

It should be stressed that both the formulations require two independent input parameters (two angels or a unit vector); however, they differ by the elements of the matrix ${}^0\boldsymbol{W}_R$ they deal with. Thus, the first formulation deals with the third row of the matrix ${}^0\boldsymbol{W}_R$, which includes only $\boldsymbol{Z}$-coordinates $[n_z\ s_z\ a_z]$ that are not sensitive to the rotation around the gravity. In contrast, the second formulation operates with second column of this matrix $[s_x\ s_y\ s_z]^T$, which incorporates X,Y-coordinates that are sensitive to mentioned rotation. As a result, the latter approach does not allow achieving desired weld slope and roll simultaneously. Therefore, the second formulation of the inverse problem is less reasonable from technological point of view.

The only case when the second formulation is sensible, is the "optimal weld orientation", for which the approaching vector is strictly vertical and, consecutively, the weld direction vector ${}^{0}\boldsymbol{n}_{w}$ lies in the horizontal plane. But the first formulation also successfully tackle this case, as it corresponds to the $(\theta, \xi)=(0,0)$. However, the second formulation can be successfully applied in the singular for the first approach case ($\theta=\pm\pi/2$), when defining the roll angle does not make sense. For this reason, both formulations of the inverse problem will be considered below.

While applying the inverse formulation to real-life problems, it should also be taken into account that engineering meaning of the slope and the roll is not sensitive to the sign of this angles. For instance, the negative slope can be easily replaced by the positive one, if the weld starting and ending point are interchanged. Also, the positive and negative rolls are equivalent with respect to gravity forces. Therefore, four cases $(\pm\theta, \pm\xi)$ must be investigated while orienting the weld joint. Similar conclusion is valid for the alternative definition of the weld orientation angles (θ, ξ'), where $\xi'>0$ but two cases $(\pm\theta, \xi')$ yield four different matrices $\boldsymbol{W}_{R}$.

4.1 Direct Kinematic Problem

As follows from (4), successive multiplication of the corresponding homogenous matrices gives, for given axis coordinates $\boldsymbol{q}$, the full world location (position and orientation) of the weld frame. Then, the required angles (θ, ξ) or (θ, ξ') are extracted from the matrix ${}^{0}\boldsymbol{W}$ in accordance with the expressions (6)-(9). Therefore, the only problem is to find the matrix $\boldsymbol{P}(\boldsymbol{q})$ that describes transformation from the positioner base to the its mounting flange (or face plate).

Because the weld joint orientation relative to the gravity is completely defined by two independent parameters, a universal welding positioner has two axes. Though, the simplest robotic cells utilise a one-axis positioners (turntables and turning rolls) that are not capable to provide full weld orientation but also increase potential of the welding station. Robotic manufactures also produce five-axis positioners that are in fact combination of two two-axis machines that are moved to the robot workspace in turn (using the 5th axis), to make possible to change the workpiece while the robot is welding the other side. Therefore, a two-axis positioner can be considered as a basic orienting component of a welding station, so the reminder of this section is devoted to positioners with two d.o.f.

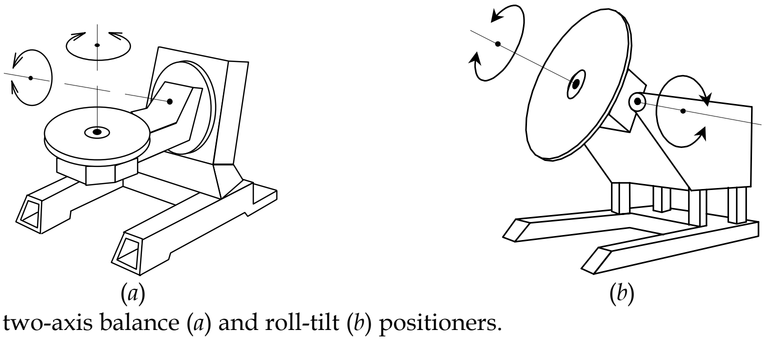

(*a*) (*b*)

Fig. 5. The two-axis balance (*a*) and roll-tilt (*b*) positioners.

While building the positioner model, it should be taken into account that the intersection point of the axes may be located above the faceplate, to be closer to the workpiece centre of gravity (Fig. 5a).

Such design allows avoiding large payload moments specific for heavy and bulky objects. But in some cases this point may lie above the plate (Fig. 5b). For this reason, it is prudent to release the usual constraint that locates the positioner base frame at the intersection of its two axes.

The kinematic model of a general 2-axis positioner is presented in Fig. 6. It includes four linear parameters (a_1, d_1, a_2, d_2) and one angular parameter α that defines direction of the *Axis*$_1$. Without loss of generality, the *Axis*$_2$ is assumed to be normal to the faceplate and directed vertically for q_1=0. The geometrical meanings of the parameters are clear from the figure.

Similar to other manipulators, the kinematics of a positioner can be described by the Denavit-Hartenberg model (Fu et al., 1987). However, for the considered 2-axis system it is more convenient to use a product of elementary transformations that can be derived directly from the Fig. 7:

$$\boldsymbol{P}(q_1,q_2) = {}^{PB}\boldsymbol{T}_1 \cdot \boldsymbol{R}_x(q_1) \cdot {}^{1}\boldsymbol{T}_2 \cdot \boldsymbol{R}_z(q_2) \tag{12}$$

where ${}^{PB}\boldsymbol{T}_1 = \boldsymbol{T}_x(a_1)\cdot \boldsymbol{T}_z(d_1)\cdot \boldsymbol{R}_y(-\alpha)$; ${}^{1}\boldsymbol{T}_2 = \boldsymbol{R}_y(\alpha)\cdot \boldsymbol{T}_x(a_2)\cdot \boldsymbol{T}_z(d_2)$, and ***T***(.), ***R***(.) are the 4×4 homogenous transformation matrices that describe translation/rotation with respect to the axes specified by the subscript.

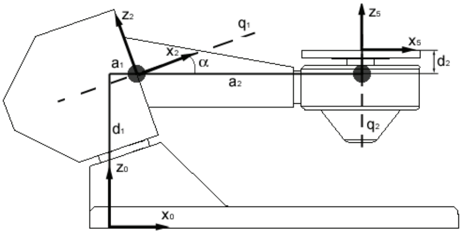


Fig. 6. The coordinate frames of the two-axis positioner.

Substituting in (12) regular expressions for translational and rotational matrices yields the final result for the not-trivial components of the positioner transformation matrix ***P***(q_1,q_2):

$$n_x = (C_1 + C_\alpha^2 V_1)C_2 - S_\alpha S_1 S_2\,; \quad n_y = S_\alpha S_1 C_2 + C_1 S_2; \quad n_z = C_\alpha S_\alpha V_1 C_2 + C_\alpha S_1 S_2 \tag{13}$$

$$s_x = -(C_1 + C_\alpha^2 V_1)S_2 - S_\alpha S_1 C_2; \quad s_y = -S_\alpha S_1 S_2 + C_1 C_2; \quad s_z = -C_\alpha S_\alpha V_1 S_2 + C_\alpha S_1 C_2; \tag{14}$$

$$a_x = C_\alpha S_\alpha V_1; \quad a_y = -C_\alpha S_1; \quad a_z = C_1 + S_\alpha^2 V_1; \tag{15}$$

$$\begin{aligned} p_x &= (C_1 + C_\alpha^2 V_1)\cdot a_2 + C_\alpha S_\alpha V_1 \cdot d_2 + a_1 \\ p_y &= S_\alpha\, S_1 \cdot a_2 - C_\alpha S_1 \cdot d_2 \\ p_z &= C_\alpha S_\alpha V_1 \cdot a_2 + (C_1 + S_\alpha^2 V_1)\cdot d_2 + d_1 \end{aligned} \tag{16}$$

where, similarly to the section 2, vectors ***n, s, a, p*** define the upper 3×4 block of the matrix ***P***, and *C, S, V* denote respectively *cos*(.), *sin*(.), *vers*(.) of the angle specified at a subscript. It should be noted, that compared to the model proposed by G.Bolmsjo (1987), the developed one includes less geometrical parameters while also describes the general case. Besides, the obtained expressions are less awkward and more computationally efficient than the known ones.

Therefore, expressions (13)-(16) completely define direct kinematics of the 2-axis positioner. But the obtained model can be also reduced to describe kinematics of a general 1-axis mechanism. It is achieved by fixing the *Axis*$_1$ or *Axis*$_2$ and choosing appropriate value of α. For instance, for turntables the axis variable is q_2 while q_1=0. But for turning rolls the axis variable is q_1 while q_2=0 and α=0.

5. Inverse Kinematic Problems

In accordance with Section 3, solving the inverse kinematic problem for the positioner means finding the axis angles (q_1, q_2) that ensure the desired world orientation of the weld joint, which is defined by the pair of the orientation angles (Problem 1) or by the unit vector (Problem 2). Let us consider these cases separately.

5.1. Solution of the Inverse Problem 1

Since the weld joint orientation angles (θ, ξ) or (θ, ξ′) completely define the third row of the orthogonal 3×3 matrix ${}^{0}\boldsymbol{W}_R$, the basic kinematic equation (4) can be rewritten as

$$\boldsymbol{\eta}^T \cdot {}^{0}\boldsymbol{W}_R = \boldsymbol{\eta}^T \cdot \left[{}^{0}\boldsymbol{T}_{PB} \cdot \boldsymbol{P}(\boldsymbol{q}) \cdot {}^{PF}\boldsymbol{T}_{WB}\right]_{3\times3} \cdot \boldsymbol{W}_R, \tag{17}$$

where the subscript ′′3 × 3′′ denotes the rotational part of the corresponding homogenous transformation matrix, and η^T=[0 0 1]. Then, after appropriate matrix multiplications, it can be converted to the form

$$\boldsymbol{v}^T = \boldsymbol{\eta}^T \cdot \boldsymbol{P}(\boldsymbol{q})_{3\times3}, \tag{18}$$

where $\boldsymbol{v}^T = \left[-S_\theta \quad C_\theta C_\xi \quad C_\theta S_\xi\right] \cdot \left[{}^{PF}\boldsymbol{T}_{WB} \cdot \boldsymbol{W}\right]^T_{3\times3}$ and, without loss of generality, transformation ${}^{0}\boldsymbol{T}_{PB}$ is assumed not to include the rotational components other then $\boldsymbol{R}_z$. Further substitution in accordance with (13) yields three mutually dependent scalar equations of two unknowns (q_1, q_2):

$$C_\alpha S_\alpha V_1 C_2 + C_\alpha S_1 S_2 = v_x; \quad C_\alpha S_1 C_2 - C_\alpha S_\alpha V_1 S_2 = v_y; \quad C_1 + S_\alpha^2 V_1 = v_z \tag{19}$$

where v_x, v_y, v_z are the corresponding components of the vector $\boldsymbol{v}$. The third of these equations can be easily solved for q_1:

$$q_1 = \pm\, acos\frac{v_z - S_\alpha^2}{C_\alpha^2} \tag{20}$$

The value of q_2 can be found by solving the first and the second equations for C_2 and S_2:

$$C_2 = \left(S_1 \cdot v_y + S_\alpha V_1 \cdot v_x\right) / C_\alpha \left(S_\alpha^2 V_1^2 + S_1^2\right); \quad S_2 = \left(S_1 \cdot v_x - S_\alpha V_1 \cdot v_y\right) / C_\alpha \left(S_\alpha^2 V_1^2 + S_1^2\right).$$

that leads to the following expression for q_2:

$$q_2 = atan2\frac{S_1 \cdot v_x - S_\alpha V_1 \cdot v_y}{S_1 \cdot v_y + S_\alpha V_1 \cdot v_x} \tag{21}$$

Therefore, Eqs. (18) and (19) represent the closed-form solution of the first inverse problem, which in the general case for given weld orientation angles (θ, ξ) or (θ, ξ′) yields two pares of the positioner axis angels (q_1, q_2).

5.2. Solution of the Inverse Problem 2

For the second formulation, the input data defines the second column of the matrix ${}^{0}\boldsymbol{W}_R$, so the basic kinematic equation (4) can be rewritten as follows:

$$ {}^{0}\boldsymbol{W}_{R}\cdot\boldsymbol{\eta} = \left[{}^{0}\boldsymbol{T}_{PB}\cdot\boldsymbol{P}(\boldsymbol{q})\cdot{}^{PF}\boldsymbol{T}_{WB}\right]_{3\times3}\cdot\boldsymbol{W}_{R}\cdot\boldsymbol{\eta}, \tag{22} $$

where $\eta=[0\ 1\ 0]^{T}$. Then, after appropriate matrix multiplications, this equation can be converted to the form

$$ \boldsymbol{u} = \boldsymbol{P}(\boldsymbol{q})_{3\times3}\cdot\boldsymbol{w}, \tag{23} $$

where

$$ \boldsymbol{w} = \left[{}^{PF}\boldsymbol{T}_{WB}\cdot\boldsymbol{W}\right]_{3\times3}\cdot\boldsymbol{\eta}; \qquad \boldsymbol{u} = \left[{}^{0}\boldsymbol{T}_{PB}\right]_{3\times3}^{T}\cdot{}^{0}\boldsymbol{W}_{R}\cdot\boldsymbol{\eta}\cdot $$

and the subscript "3×3" means the upper left 3×3 submatrix of the corresponding homogenous matrix (i.e. its orthogonal rotational part).

Further expansion of $\boldsymbol{P}(\boldsymbol{q})$ in accordance with (12) and relevant regrouping yields

$$ \left[\boldsymbol{R}_{y}(-\alpha)\cdot\boldsymbol{R}_{x}(q_1)\boldsymbol{R}_{y}(\alpha)\right]_{3\times3}\cdot\boldsymbol{u} = \left[\boldsymbol{R}_{z}(q_2)\right]_{3\times3}\cdot\boldsymbol{w}' \tag{24} $$

or, in a detailed form,

$$ \begin{bmatrix} \left(1-S_\alpha^2 V_1\right) & S_\alpha S_1 & S_\alpha C_\alpha V_1 \\ -S_\alpha S_1 & C_1 & C_\alpha S_1 \\ S_\alpha C_\alpha V_1 & -C_\alpha S_1 & \left(1-C_\alpha^2 V_1\right) \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} $$

It leads to the following scalar equations:

$$ \begin{gathered} \left(1-S_\alpha^2 V_1\right)u_x + S_\alpha S_1 u_y + S_\alpha C_\alpha V_1 u_z = C_2 w_x - S_2 w_y \quad -S_\alpha S_1 u_x + C_1 u_y + C_\alpha S_1 u_z = S_2 w_x + C_2 w_y \\ S_\alpha C_\alpha V_1 u_x - C_\alpha S_1 u_y + \left(1-C_\alpha^2 V_1\right)u_z = w_z \end{gathered} \tag{25} $$

from which the third one can be transformed to the form

$$ C_\alpha u_{xz}\cdot C_1 + C_\alpha u_y\cdot S_1 = \left(u_z - w_z\right) + C_\alpha u_{xz} $$

and solved for q_1:

$$ q_1 = atan2\frac{u_y}{u_{xz}} \pm acos\frac{\left(u_z - w_z\right) + C_\alpha u_{xz}}{C_\alpha\cdot\sqrt{u_{xz}^2 + u_y^2}}, \tag{26} $$

where $u_{xz} = S_\alpha u_x - C_\alpha u_z$. It should be noted that these two alternative solutions for q_1 correspond to different "*configurations*" of a positioner, which are strictly defined below. Besides, both the solutions must be adjusted to the feasible interval (-π,π], since the sum of *atan*2(.) and *acos*(.) can be out of the mentioned limits.

To find the value of q_2, let us consider the first two equations of system (25) and solve them for C_2 and S_2:

$$ \begin{aligned} C_2 &= \left(w_x\cdot v_x + w_y\cdot v_y\right)/\left(w_x^2 + w_y^2\right); \\ S_2 &= \left(w_x\cdot v_y - w_y\cdot v_x\right)/\left(w_x^2 + w_y^2\right), \end{aligned} \tag{27} $$

where

$$ v_x = u_x + S_\alpha\left(S_1 u_y - \mathrm{V}_1 u_{xz}\right); \quad v_y = C_1 u_y - S_1 u_{xz}\cdot $$

It leads to the following expression for q_2:

$$q_2 = atan2\frac{w_x \cdot v_y - w_y \cdot v_x}{w_x \cdot v_x + w_y \cdot v_y}. \tag{28}$$

Therefore, Eqs. (26) and (28) represent the closed-form solution of the second inverse problem, which in the general case for given unit vectors (***u***, ***w***) yields two pares of the positioner axis angels (q_1, q_2).

5.3. Solution Existence and Singularities

As follows from Eqs. (20), (21) and (26), (28), the inverse kinematic problems possess solutions for certain sets of input data that can be treated as the positioner ''*orientation workspace*''. So, for some inputs, the computation may fail and a solution does not exist (if, for instance, the *cos* argument is out of the interval [-1; 1]). In other cases, the *singularities* arise if any value of q_1 or q_2 satisfies the kinematic equation (if, for example, both arguments of *atan*2 are equal to zero). For the first inverse problem, a detailed investigation of Eq. (20) shows that the value of q_1 can be definitely computed if and only if

$$-cos(2\alpha) \le v_z \le 1\,. \tag{29}$$

Taking into account the geometrical meaning of v_z , which is the scalar product of the unit vectors extracted from the third rows of the orthogonal matrices ${}^{\circ}\boldsymbol{W}_R$ and $[{}^{PF}\boldsymbol{T}_{WB} \cdot \boldsymbol{W}_R]$ (see equations (17), (18)) , and denoting

$$v_z = cos(\chi); \quad \chi \in [0; \pi] \tag{30}$$

this condition can be presented as follows:

Proposition 1a. *For the Inverse Problem* 1, *the values of* q_1 *can be computed definitely from the expression* (20), *if and only if the angle* χ *between Z-axes of the conjugate frames* ${}^{\circ}\boldsymbol{W}_R^T$ *and* $[{}^{PF}\boldsymbol{T}_{WB} \cdot \boldsymbol{W}_R]^T$ *describing, respectively, the desired world orientation of the weld joint and its orientation relative to the positioner faceplate is less than* (π-2α) *or equal to it*:

$$0 \le \chi \le \pi - 2\alpha\,. \tag{31}$$

For a typical industrial application case, when the *Z*-axis of the workpiece frame is parallel to the positioner *Axis*$_2$, expression (29) can be also rewritten as

$$v_z = -S_\theta \cdot n_w^z + C_\theta C_\xi \cdot s_w^z + C_\theta S_\xi \cdot a_w^z \ge -C_{2\alpha} \tag{32}$$

The corresponding value of q_2 is uniquely defined by expression (21) if either its numerator or denominator is not equal to zero. A detailed investigation of the opposite case yields v_z=1 and consequently q_1=0 (case of v_z = -1 is excluded because of inequality (29)). So, the existence and uniqueness of solutions for q_2 are subject to the following proposition:

Proposition 1b. *For Inverse Problem* 1, *the value of* q_2 (*for given* q_1) *can be computed uniquely from the Eq.* (21), *if and only if the Z-axes of the conjugate frames* ${}^{\circ}\boldsymbol{W}_R^T$ and $[{}^{PF}\boldsymbol{T}_{WB} \cdot \boldsymbol{W}_R]^T$ *are not coincide, i.e.* χ>0. *Otherwise*, if *these axes coincide* (i.e. χ=0), *then* q_1=0 *and any value of* q_2 *satisfy the kinematic equation*.

Therefore, for the first inverse problem, the singularity exists only with respect to the positioner *Axis*$_2$, while it is oriented strictly vertically and upward (i.e. when q_1 = 0).

However, for the *second inverse problem*, the singularity may also arise for *Axis*$_1$. As follows from the analysis of Eq. (26), the *atan*2 is indefinite if u_{xz}=0 and u_y=0. Moreover, the corresponding kinematic equations are converted to the identity, if u_z=w_z. So any value of q_1 is a solution for such input data. The corresponding condition can also be presented as the parallelism of the vector ***u*** and *Axis*$_1$, as well as the equality for the *z*-components of ***u*** and ***w***, i.e.

$$\boldsymbol{u}=[\pm C_\alpha \;\; 0 \;\; \pm S_\alpha]^T; \quad v=[* \;\; * \;\; \pm S_\alpha]^T, \tag{33}$$

To ensure definite computing of q_1, it is additionally required that the *acos* argument in Eq. (26) belongs to the interval [-1; 1]. After appropriate rearranging, this condition can be presented as

$$|S_\alpha(C_\alpha u_x + S_\alpha u_z) - w_z| \le C_\alpha\sqrt{1-(C_\alpha u_x + S_\alpha u_z)^2}$$

After denoting the angles between the vectors ***u, v*** and the *Axis*$_1$, *Axis*$_2$ as μ, η

$$C_\alpha u_x + S_\alpha u_z = cos(\mu); \qquad w_z = cos(\eta) \tag{34}$$

and assuming that (μ, η)∈ [0;π]×'(0;π], this inequality can be rewritten as

$$sin(\alpha-\mu) \le cos(\mu) \le sin(\alpha+\mu) \tag{35}$$

that yields the following domain for (μ, η):

$$\begin{cases} \mu+\alpha-\pi/2 \le \eta \le \;\; \mu-\alpha+\pi/2 \\ -\mu-\alpha+\pi/2 \le \eta \le -\mu+\alpha+3\pi/2 \end{cases} \tag{36}$$

So, the results for q_1 can be summarised as follows:

Proposition 2a. *For Inverse Problem* 2, *the values of* q_1 *can be computed definitely from expression* (26) *if and only if the angles* μ, η *between the positioner Axis*$_1$, *Axis*$_2$ *and the vectors* ***u, w***, *respectively, satisfy inequalities* (36) and, *additionally*, μ≠0 and μ≠π. *Otherwise, if* (μ, η)=(0, π/2-α) *or* (μ, η)=(π, π/2+α), *any value of* q_1 *satisfy the kinematic equation*.

As follows from the relevant analysis, the highest "*reachability*" in the positioner ***u***-space is achieved for η∈[π/2-α; π/2+α]. And, in contrast, for η=0 or η=π, the "workspace" is reduced to a single cone with the parameter μ=π/2-α or μ=π/2+α.

In accordance with Eq. (26), computing of q_2 can fail only in the case of w_x=w_y=0, i.e. for η=0 or η=π. Geometrically, it corresponds to the vector ***w***, which is normal to the positioner faceplate and, consequently, cannot be alternated by rotation around the *Axis*$_2$. So, the existence and uniqueness of solutions for q_2 are subject to the following proposition:

Proposition 2b. *For Inverse Problem* 2, *the value of* q_2 (*for given* q_1) *can be computed uniquely from expression* (27), *if and only if the angle* η *between the positioner Axis*$_2$ *and the vectors* ***w*** *satisfy the conditions* η≠0 and η≠π. *Otherwise, if* η=0 or η=π, *any value of* q_2 *satisfies the kinematic equation* (*provided that the solution for* q_1 *exists*).

Therefore, for the second inverse problem, the singularity may exist for both axes, when ***u*** is parallel to *Axis*$_1$ or ***w*** is parallel to *Axis*$_2$.

5.4. Positioner Configurations

Similar to other manipulating systems, the positioner inverse kinematics is non-unique because of existence of two solution branches (see ± sign in Eqs. (20) and (26)). However, both the off-line programming and the real-time control require distinguishing between them to ensure continuity of the positioner motions. For this reason, the direct kinematics must yield an additional output, *configuration index* M=±1 describing positioner posture, which is also used as an additional input for the inverse transformation, to produce a unique result.

For inverse problem 1, the configuration index is defined trivially (see Eq. (20)), as the sign of the coordinate q_1 :

$$M_1 = sgn\left(q_1\right). \tag{37}$$

But for inverse problem 2, such index must identify the sign of the second term only (see Eq. (26)). So, it should be defined as

$$M_2 = sgn\left(q_1 - atan2\frac{u_y}{u_{xz}}\right). \tag{38}$$

From the geometrical point of view, the index M_2 indicates relative location of two planes, passing the $Axis_1$. The first of them is obtained by rotating of the X_0Z_0-plane around the $Axis_1$ by the angle q_1. And the second plane is passed via the $Axis_1$ and the vector $\boldsymbol{u}$. It should be also noted, that the index M_2 substantially differs from the traditional for robotics orientation index M_5=sgn(q_5), which describes the wrist configuration of the typical 6 d.o.f. manipulator.

5.5. Optimal Orienting of the Weld Joint

As adopted by practising engineers, the optimal weld orientation is achieved when the approaching vector is strictly vertical and consecutively, the weld direction vector lies in the horizontal plane, i.e. (θ, ξ)=(0, 0) and ${}^{o}\boldsymbol{s}_w$= [0 0 1]. Let us investigate this particular case in details.

For both the inverse problems, substation of the values (θ, ξ) and the vector ${}^{o}\boldsymbol{s}_w$ into Eqs. (20), (21) and (26), (28) yields the similar result:

$$q_1 = \pm\ \text{acos}\frac{s_w^z - S_\alpha^2}{C_\alpha^2}; \qquad q_2 = \text{atan2}\ \frac{S_1 \cdot s_w^x - S_\alpha V_1 \cdot s_w^y}{S_1 \cdot s_w^y + S_\alpha V_1 \cdot s_w^x}. \tag{39}$$

So, the condition of the solution existence (36) is reduced to

$$s_w^z \geq -C_{2\alpha} \quad \text{or} \quad \eta \leq \pi - 2\alpha \tag{40}$$

It means, that the "*working space*" of the positioner does not include the cone with the downward directed central axis and the aperture angle 4α. And, thereby, the corresponding welds can not be optimally oriented. But it can be proved that applying the first inverse problem with the input parameter

$$\xi' = max\{0;\ 2\alpha + \eta - \pi\}, \tag{41}$$

the orientation of such welds can be essentially improved and approached to the optimal one. The corresponding "suboptimal" solution is defined by the axis angles

$$q_1 = \pi; \quad q_2 = -a\,tan\,2\left(s_w^y / s_w^x\right) \tag{42}$$

i.e. exact equalities are achieved for the first and the second equations of system (19), while for the third one the residual is minimised only.

6. Welding Task Planning

6.1. Components of the Cycle Time

The overall cycle time of the robotic arc welding cell is basically determined by two major components: (i) the "arc time" which is actually spent for the welding and (ii) the "motion time" required for the torch and workpiece repositioning between the welding operations. There are also several other time components related to the workpiece cooling and downloading/uploading, torch-tip cleaning, equipment adjustment, maintenance, etc., but these are beyond the scope of the welding process model considered here.

As stressed above, the arc time is minimal in the downhand case, and the torch speed should be reduced for the "out-of-position" welding. Since the downhand location corresponds to θ = ξ = 0, we propose approximating the welding speed reduction by the following expression:

$$V(\theta,\xi) = (1 + k_v \ \| \ (\theta,\xi)^T \ \|)^{-1} \cdot V_o \tag{43}$$

where V_o is the downhand welding speed, k_v is the scaling factor, and $\| . \|$ denotes the algebraic norm of the vector $(\theta, \xi)^T$. It is obvious that definition of this norm for a particular welding task is not trivial and must rely on the expert judgements, which assign the maximum allowable tolerances (θ_{max}, ξ_{max}) and corresponding reduction factor k_{max}. An example of such a definition is given in our paper (Dolgui and Pashkevich, 2005).

The robot and positioner motion times highly depend on the welding task sequencing, which prevents unreasonable movements of the torch and workpiece. For a single trajectory, the motion time depends on the control mode (simultaneous or sequential), travel distance, and velocity/acceleration limits incorporated in the path-planning algorithms. It can be proved that the minimum time for a single joint movement (for both the robot and positioner) is defined by the equation

$$\tau = \begin{cases} |\Delta q| \, / \, \dot{q}_{\max} + \dot{q}_{\max} \, / \, \ddot{q}_{\max} \, , & if \quad |\Delta q| > \dot{q}^2_{\max} / \ddot{q}_{\max} \\ 2\sqrt{|\Delta q| \, / \, \ddot{q}_{\max}} \, , & \text{otherwise} \end{cases} \tag{44}$$

where Δq is the joint displacement, and $\dot{q}_{\max}$, $\ddot{q}_{\max}$ are the velocity/acceleration limits.

Then, the *robot* motion time for a single torch displacement (simultaneous axis control) is defined by the slowest axis drive

$$t_R = \max_i \{ \tau_i \} \tag{45}$$

where i=1,…6 is the axis number. It should be noted that the latter expression is valid, if the path planning is performed in the manipulator joint-space. In the alternative case (the Cartesian-space path planning), the index variable must be extended to i=0,…6, where τ_0 is computed via the Cartesian displacement and relevant velocity/acceleration constraints in a similar way.

In contrast, the *positioner* motion time for a single workpiece reconfiguration (sequential axis control) is defined as the sum of the axis motion times, the pauses $\Delta\tau_i$ between the successive axis movements and also the auxiliary time τ_R

$$t_P = \sum_i \tau_i + \sum_i \Delta\tau_i + \tau_R \ , \tag{46}$$

where τ_R is required for the welding torch removing to the safe location, to avoid possible torch-workpiece collisions. As follows from this equation, its preferable to avoid achieving the downhand location for each weld individually, since it requires extra time for the torch removing that may overcompensate the downhand welding benefits.

6.2. Welding Task Sequencing

Because of its complexity, the general problem of the robotic *welding task planning* is usually broken in several hierarchical stages (Kim et *al*., 1998). This decomposition implements a problem-specific mechanism, which makes its possible to reduce the size of the related combinatorial optimisation problems, while maintaining the productivity/quality compromise. These stages are defined as follows:

(i) The *weld seam clustering*, which arranges the welds into the groups that can be welded without changing the positioner configuration (in the downhand position preferably, within the allowable tolerances);

(ii) The *intra-cluster sequencing*, which determines the seam start/end points and the welding order for each cluster individually (minimising the robot motions subject to the heat-distortion-related constrains);

(iii) The *inter-cluster sequencing*, which determines the cluster processing order that minimises the positioner motions subject to the downhand-related constrains.

After completing these stages, each operation is further broken down into detailed robot-positioner movements, which finally yields the workcell control programs.

For the first stage, one can be applied the well-developed general clustering techniques (Everitt et al., 2001) that group together the welds with similar $\boldsymbol{s}_w$-vectors, which indicate the seam normal line directions relative to the workpiece base. However, taking into account the welding specificity, it is prudent to perform the clustering in terms of the θ-, ξ-angles introduced above. This poses the following optimisation problem for obtaining the positioner coordinates $\boldsymbol{q}_p$

$$\max_{i\in C_i}\left\{\|\,(\theta_i(\boldsymbol{q}_p),\,\xi_i(\boldsymbol{q}_p)\,)^T\,\|\right\}\to\min_{\boldsymbol{q}_p}, \tag{47}$$

which is solved within the usual clustering algorithms while evaluating the cluster diversity. In this expression, C_i denotes the *j*th cluster index set, $\|\,.\,\|$ is the norm of the vector $(\theta,\xi)^T$, and the functions $\theta_i(\boldsymbol{q}_p)$, $\xi_i(\boldsymbol{q}_p)$ define the *i*th weld orientation relative to gravity for the given positioner coordinate vector $\boldsymbol{q}_{p.}$. Obviously, in order to ensure the desired welding quality, it is necessary to constraint the inter-cluster diversity by assigning the upper bound $\Delta_{\max}$ for the above norm

$$\|\,(\theta(i,\boldsymbol{q}_p^i),\,\xi(i,\boldsymbol{q}_p^i)\,)^T\,\|\le\Delta_{\max}\,,\ \ \forall i\in C_j, \tag{48}$$

which is also easily incorporated in the existing clustering methods. An example of slightly different weld clustering techniques, based on the "representative weld-line" concept, is given in Kim et *al*. (1998).

For the second stage, there are already exist a number of problem-specific heuristics, neural network based methods, and genetic algorithms that allow generating the minim-time inter-cluster welding sequence. These take into account the heat-related distortions by assigning the minimum cooling time, size of the heat-affected zone, etc. A comprehensive review of recent advances in this area is given by Kim et *al*. (2005).

This section focuses on the third welding task planning stage, the inter-cluster sequencing, related to the optimisation of the positioner motions. To our knowledge, the only paper that addresses this problem directly is that of Kim et *al*. (1998) devoted to the welding operations planning in a robotic workcell with a rotating-tilting positioner. However, their approach assumes that each cluster is oriented using the representative weldline, which is transformed to the strictly downhand location by the positioner. Besides, after such orienting, the cluster welding time is assumed to be constant and computed directly from the welding speed and the weld line length.

In contrast to the known approach, the proposed technique admits the *out-of-position* (i.e. not exactly downhand) weld location, which is charged by the welding speed reduction. From this, one can pose a problem of minimum-time cluster sequencing by finding a o *trade-off* between the positioner motion time and the cluster processing time. Another useful feature of the proposed approach is the *re-clustering* ability. This means that the developed algorithm is able to find the same positioner configuration for several neighbouring clusters (i.e. to merge them), if the corresponding increase of the welding time is over-compensated by the reduction of the robot-positioner motion time. This allows to modify the clustering stage, which may impose very strong constraints on the inter-cluster similarity, and partly combine the clustering stage with the cluster sequencing one.

6.3. Associated optimisation problem

In a more formal way, the considered problem may be stated as follows. Let us assume the whole set of the welds is preliminary *clustered* in *n* groups, while the positioner joint coordinate space is *sampled* and presented by the uniform grid with the set of nodes $\{Q_p\}$. For each such node and each welding cluster, let us evaluate the orientation feasibility of the all the inter-cluster welds (within the given tolerance $\Delta_{\max}$) and compute corresponding processing times required for both the welding and the time-optimal torch movements between welds. In this step, the cluster welding time is adjusted in accordance with the allowable welding speed. Using the data obtained, let us create a cluster set $\{Q_p^{(1)}, Q_p^{(2)}, \cdots Q_p^{(n)}\}$ in the sampled positioner coordinate space, where each cluster $Q_p^{(i)}$ is composed of admissible nodes $\{Q_p^{(ik)} \mid k = \overline{1, m_i}\}$ with their processing time attributes $\{T_w(i,k) \mid k = \overline{1, m_i}\}$. Besides, let us assume that the positioner inter-node motion times $\{T_m(i_1, k_1, i_2, k_2) \mid k_i = \overline{1, m_i}\}$ are also computed, and is given the auxiliary time τ_R required for the torch moving before/after positioner re-configuration. Then, the minimum-time objective for the inter-cluster sequencing may be written as:

$$\sum_{i=1}^{n} T_w(i,k_i) + \sum_{(i_1,i_2)\in I} T_m(i_1,k_{i1},i_2,k_{i2}) + \tau_R \cdot \sum_{i=1}^{n} 1(Q_p^{(i_1k_{i1})} \neq Q_p^{(i_2k_{i2})}) \to \min_{I,K} \tag{49}$$

where *I*, *K* denote the optimal cluster sequence and corresponding cluster node numbers, k_i defines the optimal node within the *i*th cluster, the first term accumulates the cluster-processing times (welding and torch travel times) , the second term represents the positioner motion times (between the clusters), and the third term takes into account the auxiliary robot motions between/after the positioner re-configurations via the indicator function 1(.).

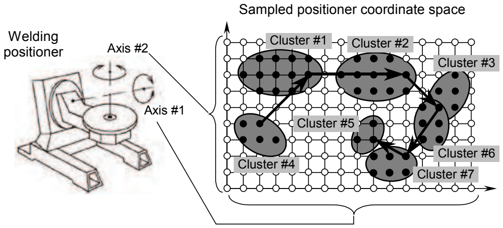


Fig 7. Representation of the inter-cluster sequencing problem in the positioner coordinate space.

A geometrical interpretation of this formulation is presented in Fig. 7 , where each welding cluster is defined by the set of feasible nodes in the positioner coordinate space. For each node within each cluster, we assign the *node-processing time* that varies within the cluster, with the minimum located in the cluster centre. For each pair of nodes, there are also given the *inter-node travel times*. Besides, it is assumed that each non-zero travel requires some additional *pause-time*, needed for preparations of the safe movements. The goal is to find the minimum-time tour, which visits each cluster exactly ones, defined by both the cluster sequence and the set of visiting nodes within the clusters.

The hypothetical solution (see Fig. 7) shows a compromise between visiting cluster centres and their peripheries, since the node-processing times and the inter-node travel times compete inside the total sum to be minimised. Moreover, this solution utilises advantages of cluster overlapping by processing the clusters #3, #6 in the same positioner configuration (this removes the corresponding pause-time between successive positioner motions). Another interesting feature is the one-axis motion between the clusters #1 and #2, which is more time-economical than the two-axis motion requiring a pause between activating the axis drives.

It should be noted that this study focuses on generating the *non-closed* optimal tours for cluster visiting and processing, assuming that the workpiece downloading/uploading is performed at the tour start/end points interchangeably. However, the proposed technique is easily modified for the case of the closed tour, as well as for the case of a predetermined loading positioner configuration.

Another essential specificity of the studied optimisation problem is related to cluster geometry, since the welding-task clusters are usually composed of several disjoint regions in the positioner coordinate space, corresponding to different inverse kinematic solutions. The next subsection proposes an optimisation algorithm that takes into account this problem specificity.

6.4. Optimisation Algorithm

As shown above, the considered problem of the inter-cluster operation planning can be converted into the *generalised TSP* (*GTSP*), in which the set of given nodes consists of several clusters (*overlapped,* in the general case) and the objective is to find the shortest route passing through each cluster exactly once. The GTSP was first mentioned in operation research literature in relations to computer files sequencing (Henry-Labordere, 1969). Further applications dealt with flexible manufacturing scheduling and material-flow systems design (Laporte and Palekar, 2002).

Since the GTSP is NP-hard and the exact algorithms usually are not able to obtain the optimal solution in a reasonable time, a variety of heuristics exist. The simplest ones are based on adaptation of the standard TSP techniques (such as nearest-neighbour, farthest-insertion, etc.). The most sophisticated to our knowledge GTSP heuristic, GI^3, was proposed by Renaud and Boctor (1998). However, it performs well for small cluster sizes only, and their simplest structure. Besides, the known technique employs some assumptions (Euclidian distances, for instance) that are not valid for the robotic welding application studied here.

To improve efficiency and computational speed in the case of the overlapping multi-region clusters and non-Euclidean distances, the the GI^3 was revised by simplifying the first two phases and replacing them with a straightforward generation of a random initial solution. Besides, the third phase is run in a slightly different manner, using two nested loops. The internal loop contains a classical TSP tour-improvement routine, which is repeated until no improvement is achieved by modifying the cluster sequence (for a fixed node subset). Then, within the external loop, the node-improvement routine is invoked to optimize the node subset, while the cluster sequence (tour) is considered not to vary. The external loop is also repeated until no improvement is achieved, converging to a local optimum. To increase the chances of attaining the global optimum, the algorithm is repeated several times, each time starting from a new random initial solution and finishing by updating the best solution. The basic idea and outline of the proposed composite algorithm (called GI^{2+}) are given below:

```
Algorithm: Heuristic GI2+
(1) Generate random solution {Tour, Node}
(2) Set done:=0;
(3) Repeat while done=0
    (a) Repeat while done=0
        (i) Set done:=1;
        (ii) ImproveTour( );
           Set done:=0, if improved
    (b) ImproveNodes( );
           Set done:=0, if improved
(5) Update best solution and  repeat from Step 1, if
    desired.
```

In contrast to the original GI^3 technique, the proposed heuristic completely revises the subset after each tour alteration, and the tour improvement is run indefinite number of times, until no improvement is achieved (The GI^3 favours to the predetermined number of iterations for the tour improvement and for the reduced node improvement, executed for several neighbouring clusters only). The main reason for this amendment is the multi-region cluster geometry that disables the ideas implemented in the GI^3. Besides, the proposed algorithm structure (with two nested loops) simplifies the coding while keeping reasonable computational speed. The main procedures incorporated in the GI^{2+} algorithm (generation of an *initial solution*; *tour-improvement* and *subset-improvement*) are described in details in (Dolgui and Pashkevich, 2005).

6.5. Computational Results

The developed algorithm was tested on a number of randomly generated problems comprising from 50 to 1000 nodes and compared to the exact branch-and-bound technique. It was run on a PC with a 2.8 GHz Intel Pentium IV processor and 512 MB memory. To take into account specific properties of the considered robotic welding application, each cluster was build as a union of two disjoint circles, and the clusters were allowed to have essential intersection regions.

Problem Size			Reaching of Optimum (Heuristics/ Exact)				Relative Speed (Exact/Heuristics)		
n	*m*		Succ%	Mean	Max		Mean	Min	Max
5	10		87	1.005	1.202		2.73	1.82	3.87
	100		95	1.003	1.137		2.84	1.55	3.73
6	10		87	1.011	1.220		10.24	6.37	14.53
	100		89	1.003	1.166		11.20	6.92	16.70
7	10		55	1.013	1.182		41.03	22.96	56.05
	100		77	1.010	1.130		50.03	7.80	79.75

Table 1. Quality of the heuristic solutions (10 iterations)

The results presented in Table 1 indicate that problems involving up to 1000 nodes and 10 clusters were solved to optimality in 70-80% of cases, and an average solution is only 0.3…2.6 %

over the optimum, that is quite acceptable for engineering applications. In contrast, the exact algorithm (based on the branch-and-bound with the nested dynamic programming) can handle only low-dimensional problems (up to 5×100, 6×50, 7×20, 8×10). However, for smaller number of clusters ($n < 5$) the branch-and-bound method takes over the heuristic. To summarise the heuristic empirical performance, the running times were approximated by the expression $Time = c \cdot n^a m^b$ using the log-least-square technique that yielded $a \approx 1.44$ and $b \approx 1.92$. This shows that the heuristics remains rather moderate with respect to n, while for the exact algorithm the problem difficulty severely increases with the number of clusters.

7. Conclusion

Resent advances in arc welding technology motivate rethinking of some postulates and conventions incorporated in the existing robot control methods. This chapter addresses relaxing of the downhand-position assumption, which became a de-facto standard in robotic welding and requires the weld normal vector to be opposite to gravity. In contrast to the standard techniques, the developed method explicitly assumes that a weld may be processed in the non-downhand location, within given tolerances. But, to ensure the prescribed quality, the downhand deviation is charged by reduction of the welding speed. For such settings, it is proposed a novel method for the kinematic control of a robot-positioner system and related optimisation algorithm for the cluster-level planning of the welding operations. By using this technique in combination with the existing CAD tools, it is possible essentially reduce the cycle time of the robotic welding station.

8. References

Ahmad, S. & Luo, S. (1989). Coordinated motion control of multiple robotic devices for welding and redundancy coordination through constrained optimization in Cartesian space. *IEEE Transactions on Robotics and Automation*, Vol. 5, No 4, (Aug. 1989), 409–417.

Bolmsjo, G. (1987). A kinematic description of a positioner and its application in arc welding robots, In: *Proc. of the 2nd Int. Conference on Developments in Automated and Robotic Welding*, pp. 77-84., Nov. 1987, London, UK.

Bolmsjo, G.; Olsson, M. & Cederberg, P. (2002). Robotic arc welding—trends and developments for higher autonomy. *Industrial Robot: An International Journal*, Vol. 29, No 2, 98–104.

Cary, H.B. (1995). *Arc welding automation ,* Marcel Dekker, New York.

Dolgui, A., & Pashkevich, A. (2006). Cluster-level operations planning for the out-of-position robotic arc welding. *International Journal of Production Research*, Vol. 44, No 4, (Febr. 2006), 675-702.

Everitt, B.S.; Landau, S. & Leese, M. (2001). *Cluster Analysis*, Oxford University Press, New York.

Fernandez, K. & Cook, G.E. (1988). A generalized method for automatic downhand and wire feed control of a welding robot and positioner, *NASA Technical Paper 2807*.

Fu, K. S.; Gonzalez, R. C. & Lee, C. S. G. (1987). *Robotics: Control, vision and intelligence.* McGraw-Hill, NewYork.

Fukuda, S. & Yoshikawa, K. (1990). Determination of welding sequence: a neural net approach, *Engineering Analysis with Boundary Elements*, Vol. 7, No 2, (June 1990), 78–82.

Grenestedt, J.L. (2003). Optimization of the weld path for overlay coatings. *Journal of Structural and Multi-Disciplinary Optimization*, Vol. 25, No 3, (Aug. 2003), 215–224.

Gutin, G. & Punnen, A.P. (Eds), (2002). *The Traveling Salesman Problem and its Variations*, Kluwer Academic Publishers, Dordrecht.

Henry-Labordere, A.L. (1969). The record balancing problem: a dynamic programming solution of a generalized traveling salesman problem. *Revue d'informatique et de recherche operationnelle* (*RIRO*), Vol. B-2, 43–49.

Kim, D.W.; Choi, J.-S. & Nnaji, B.O. (1989). Robot arc welding operations planning with a rotating/tilting positioner. *International Journal of Production Research*, Vol. 36, No 4, (Apr. 1998), 957–979.

Kim, H.J.; Kim, Y.D. & Lee, D.H. (2005). Scheduling for an arc welding robot considering heat-caused distortion. *Journal of the Operational Research Society*, Vol. 56, No 1, (Jan. 2006), 39 – 50.

Kim, K.Y.; Kim, D.W. & Nnaji, B.O. (2002a). Robot arc welding task sequencing using genetic algorithms. *IIE Transactions on Design and Manufacturing*, Vol. 34, No 10, (Oct. 2002), 865–880.

Kim, K.Y.; Norman, B. & Nnaji, B.O. (2002b). Heuristics for single-pass welding task sequencing. *International Journal of Production Research.*, Vol. 40, No 12, (Aug. 2002), 2769–2788.

Laporte, G. & Palekar, U. (2002). Some applications of the clustered travelling salesman problem. *Journal of the Operational Research Society*, Vol. 53, No9, (Sept. 2002), 972-976.

Nikoleris, G. (1990). A programming system for welding robots. *International Journal for the Joining of Materials*, Vol. 2, No 2, 55–61.

Pashkevich A. ; Dolgui A. & Semkin K.(2003). Kinematic aspects of a robot-positioner system in an arc welding application. *Control Engineering Practice*, Vol 11, No 6, (June 2003), 633–647.

Pires, J.N; Loureiro, A., Godinho, T., Ferreira, P., Fernando, B. & Morgado, J. (2003) Welding robots. *IEEE Robotics and Automation Magazine*, Vol. 10, No 2, (June 2003), 45- 55.

Ranky, P.G. (2004). Automotive robotics. *Industrial Robot: An International Journal;* Vol. 31, No 3, (June 2004), 252 – 257.

Renaud, J. & Boctor, F.F. (1998). An efficient composite heuristic for the symmetric generalized travelling salesman problem. *European Journal of Operational Research*, Vol. 108, No 3, (Aug. 1998), 571-584.

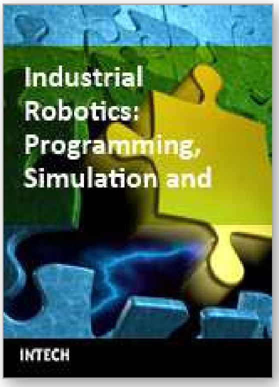
Rubinovitz, J. & Wysk, R.A. (1988). Task level off-line programming system for robotic arc welding – an overview. *Journal of Manufacturing Systems*, Vol. **7,** No 4, 293–299.

Tarn, T.J.; Chen, S.B. & Zhou, C. (Eds.), (2004). *Robotic Welding, Intelligence and Automation: Lecture Notes in Control and Information Sciences*, Vol. 299, Springer-Verlag, New York.

Tolinski, M. (2001). Getting to the Core of Welding Wire. *Forming and Fabricating*: *SME Journal*, Vol. 8, No 1, 42 –49.

Wu, L.; Cui, K. & Chen, S. B. (2000). Redundancy coordination of multiple robotic devices for welding through genetic algorithm. *Robotica*, Vol. 18, No 6, (Nov. 2000), 669 – 676.

Yagi, T. (2004). State-of-the-art welding and de-burring robots. *Industrial Robot: An International Journal*, Vol. 31, No 1, (Jan. 2004), 48-54.



Industrial Robotics: Programming, Simulation and Applications
Edited by Low Kin Huat

ISBN 3-86611-286-6
Hard cover, 702 pages
Publisher Pro Literatur Verlag, Germany / ARS, Austria
Published online 01, December, 2006
Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821