We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Path Coordination Planning and Control in Robotic Material Handling and Processing

Xuan F. Zha[1], Xiaoqi Chen[2]
[1]*National Institute of Standards and Technology & University of Maryland*
Gaithersburg, MD 20899
Email: zha@cme.nist.gov,xfzha@ieee.org
[2]*Singapore Institute of Manufacturing Technology*
71 Nanyang Drive, Singapore 638075
Email : xqchen@simtech.a-star.edu.sg

1. Abstract

This chapter presents a unified approach to coordination planning and control for robotic position and orientation trajectories in Cartesian space and its applications in robotic material handling and processing. The unified treatment of the end-effector positions and orientations is based on the robot pose ruled surface concept and used in trajectory interpolations. The focus of this chapter is on the determination and control of the instantaneous change laws of position and orientation, i.e., the generation and control of trajectories with good kinematics and dynamics performances along such trajectories. The coordination planning and control is implemented through controlling the motion laws of two end points of the orientation vector and calculating the coordinates of instantaneous corresponding points. The simulation and experiment in robotic surface profiling/finishing processes are presented to verify the feasibility of the proposed approach and demonstrate the capabilities of planning and control models.

Keywords: Robot pose ruled surface, Unified approach, Trajectory planning and control, Off-line programming, Robotics polishing

2. Introduction

Motion coordination planning and control play a crucial role in robot applications to Cartesian task operations with the consideration of kinematics and dynamics constraints. To effectively carry out the design and application of robots, an efficient algorithm for motion planning needs to be developed by generating, simulating and evaluating, and optimizing robot trajectories in a virtual CAD off-line programming environment (Zha 1993, Zha et al 1994, Zha and Du 2001). Many joint-space and Cartesian space based planning schemes have been made out by proposing new algorithms or improving the existing algorithms to describe and plan robot motions and trajectories (Thompson and Patel 1987; Bobrow et al 1985; Shin and McKay 1985; Pfeiffer and Johanni 1987; Yang and Jin 1988; Slotine and Yang 1989; Shiller and Lu 1990; Patel and Lin 1995; Konjovic and Vukobratovic 1994; Lee 1995;

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

Seereeram and Wen 1995). Some criteria are identified for comparing and evaluating both trajectories and trajectory planners of robot (Thompson and Patel 1987):

(1) Trajectories should be effective both to compute and to execute.
(2) Trajectories should be predictable and accurate, and they should not degenerate unacceptably near a singularity.
(3) The position, velocity, and acceleration, and even the rate of change of acceleration, called the jerk, should be smooth functions of time.
(4) Trajectory planner should be possible to determine efficiently whether a proposed trajectory requires the robot end effector to move to a point outside its workspace or move with a velocity or acceleration that is physically impossible. Both of these are controlled with a good model.

The coordination planning and control for robot motions in Cartesian space has long been recognized as a most interesting but difficult research field not only for motion control but also for advanced virtual/real design and planning. Some of the existing methods are, for the most part, based on kinematics considerations and geometric flavor. They may render to have limited capabilities for handling some cases where the limits of maximum acceleration and maximum deceleration along the solution curve are no longer met, or where singular points or critical points of robot configuration exist. Another problem with existing methods to plan trajectories is that the unmanageable complexity could occur both in computation cost and storage.

This chapter aims to present a unified approach to coordination planning and control of pose trajectories for robot end effector in Cartesian space. The organization of the chapter is as follows. Section 2 introduces some issues related to robot task analysis and control, including the evaluation of robot kinematics performance; Section 3 proposes a new approach to robot pose trajectory planning by generating robot pose ruled surface. Section 4 deals with the optimization solutions to the problem of pose trajectory planning. Section 5 provides simulation of the proposed approach in virtual environment and examples. Section 6 demonstrates the industrial application for robotic polishing. Section 7 summarizes the chapter with concluding remarks.

3. Analysis and Control for Robot Manipulators

There are different levels of abstraction at which a robot task can be defined. For simplicity, the lowest level task description is adopted so that a robot task is specially defined as a collection of working points to be followed by the end-effector in the task space. If the robot is to follow a prescribed path, this task can be approximated by a set of points along the path. Other task specifications such as obstacle avoidance, position accuracy, and static capability at task points can also be included as part of the definition of a task. In the following context, the task point is referred to as the pose (position and orientation) of the end-effector, which is dependent on the configuration of robot manipulator.

3.1 Robot Pose and Configuration

The robot configuration in Cartesian space can be described by the position reference point, $\boldsymbol{P}$, and the orientation vector $\boldsymbol{\Phi}$ on the line $\boldsymbol{S}$ passing through the point $\boldsymbol{P}$. Thus, the configuration equation is expressed as follows (Makino et al 1982),

$$\boldsymbol{\chi}=[\boldsymbol{P},\ \boldsymbol{\Phi}]^{T} \tag{1}$$

The continuous motion of configuration in three-dimensional Cartesian space forms a ruled surface, called robot pose ruled surface (Zha 1993). Two end points of the orientation vector $\mathbf{\Phi}$ are supposed to be $\boldsymbol{P}$ and $\boldsymbol{Q}$ on the $\boldsymbol{S}$, and thus the starting and ending points of robot motion are P_s and Q_s, P_e and Q_e, as shown in Figure 1, respectively. The two base curves of the robot pose ruled surface, i.e., the robot pose trajectories P_sP_e and Q_sQ_e can be expressed as the vector equations of a function of joint variable with respect to time parameter t, as follows,

$$\begin{cases} \mathbf{P}(\mathrm{t}) = \mathbf{r}_1(\mathrm{t}) \\ \mathbf{Q}(\mathrm{t}) = \mathbf{r}_2(\mathrm{t}) \\ \mathbf{\Phi}(\mathrm{t}) = \mathbf{r}_2(\mathrm{t}) - \mathbf{r}_1(\mathrm{t}) \end{cases} \tag{2}$$

where, $\mathrm{t} \in [\mathrm{t}_1,\mathrm{t}_2]$, t_1 and t_2 are the start and end time of motion respectively. Using a standard mathematical equation to express the robot pose ruled surface, Eq.(2) can be rewritten as (Zha and Du 2001)

$$\mathbf{r}(t,\lambda) = \mathbf{r}_1(t) + \lambda[\mathbf{r}_2(t) - \mathbf{r}_1(t)] = \mathbf{r}_1(t) + \lambda\mathbf{\Phi}(t) \tag{3}$$

where, $\lambda \in [0,1]$.

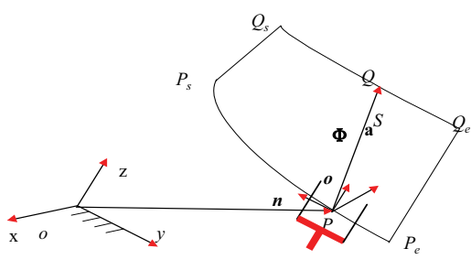


Fig. 1. Robot configuration and pose ruled surface.

From Eq.(3), when $\lambda = 0$, $\mathbf{r}(t,0) = \mathbf{r}_1(t)$ represents a pure position trajectory; while $\lambda = 1$, $\mathbf{r}(t,1) = \mathbf{r}_2(t)$ represents a pure orientation trajectory. The equation of the robot pose ruled surface is therefore flexible to represent its motion (both for position and orientation) trajectories. The orientation vector can be either an equivalent angular displacement vector or any other representational orientation vectors, e.g., an Euler angular vector (θ, ϕ, ψ). The properties of the equivalent angular displacement vector used as the orientation vector are different from those of normal vectors in mathematics (Makino et al 1982). It is noted that the representation for robot motions using the pose ruled surface is exceptional to handle the case when the orientation vector is in line with or parallel to the position vector as in these two cases no ruled surface can be formed or generated.

3.2 Motion Analysis

From the configuration equation, Eq.(1), the Cartesian velocity and acceleration, $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$, of the robot can be obtained by use of the first-order and the second-order derivatives, respectively, as follows

$$\dot{\mathbf{X}} = [\,\dot{\mathbf{P}}\,,\,\dot{\Phi}\,]^{T} = \boldsymbol{J}\ \dot{\mathbf{q}}\ = [\,\dot{\mathbf{r}}_{\mathbf{1}}\,(t),\ \dot{\mathbf{\Phi}}(t)\,]^{T} \tag{4}$$

$$\ddot{\mathbf{X}} = [\,\ddot{\mathbf{P}}\,,\,\ddot{\Phi}\,]^{T} = \mathbf{J}\,\ddot{\mathbf{q}}\ + \dot{\mathbf{J}}\dot{\mathbf{q}}\ = [\,\ddot{\mathbf{r}}_{\mathbf{2}}(t)\,,\,\ddot{\mathbf{\Phi}}(t)\,]^{T} \tag{5}$$

where, $\dot{\mathbf{P}}$ and $\ddot{\mathbf{P}}$ are the Cartesian linear velocity and acceleration of the robot position; $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the joint velocity and acceleration; $\dot{\Phi}$ and $\ddot{\Phi}$ represent the orientation velocity and acceleration; and $\mathbf{J}$ is a Jacobian matrix. Based on the theory of velocity addition, $\dot{\Phi}$ can be written as

$$\dot{\Phi}\ = \sum_{j=1}^{n}\ \dot{q}_j\,\delta_j\,\overline{e}_j \tag{6}$$

where, $\overline{e}_j = (\prod_{i=0}^{j} {}^{E}\,\hat{e}_i^{\delta_i\ q_i})\,\hat{\overline{e}}_j$ is a unit vector of j-th joint axis in the base coordinate system; $\hat{\overline{e}}_j$ is a unit vector of j-th joint axis in dynamic coordinate system; δ_j has the same meaning as δ_i mentioned in (Makino et al 1982). Hence, the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ can be expressed as

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_w \end{bmatrix} = \begin{bmatrix} \dfrac{\partial P}{\partial q_1} & \dfrac{\partial P}{\partial q_2} & \cdots & \dfrac{\partial P}{\partial q_n} \\ \delta_1\overline{e}_1 & \delta_2\overline{e}_2 & \cdots & \delta_n\overline{e}_n \end{bmatrix} \tag{7}$$

where, $\mathbf{J}_v$ and $\mathbf{J}_w$ are velocity Jacobian and angular Jacobian corresponding to $\mathbf{V}$ and $\dot{\Phi}$ respectively.

3.3 Motion Performances Evaluation

The performance constraints are defined to ensure the feasibility of a robot configuration while performing the given task. Great advances have been made in recent years in the study of robotic kinematics performance. For example, in order to obtain an energy-saving motion mode, an efficient way is to minimize the joint motion, i.e., $\min(\sum_{i=1}^{n} |\, q_i(t+\Delta t) - q_i(t)\,|^2)$ or the sum weighted distance or path (Section 6). However, it is not sufficient to consider only the position and orientation of manipulators for task planning. In fact, in many cases, it is required to satisfy certain constraints for not only the position and orientation but also the velocity and even the acceleration, i.e., kinematics and dynamics and control constraints.

3.3.1 Manipulability

The most important and commonly used concept is the robot manipulability, which is a measure of manipulating ability of robotic mechanisms in positioning and orientating the end effector which is beneficial for design and control of robots and for task planning. The measure for manipulability was first defined by Yoshikama (1985), as

$$W = \sqrt{\det(\mathbf{J}\mathbf{J}^{T})} \tag{8}$$

Since W is equal to zero at the singular positions, $W \neq 0$ and its increase will enlarge the dexterity of the system and assure avoiding "avoidable" singularities. For a non-redundant manipulators, i.e., $m=n$, the measure W reduces to $W=|\det(\mathbf{J})|$. The manipulability constraint is only used to detect whether a manipulator is at or nearby a singular pose when its end-effector reaches each of the task points, and thus it is a local manipulability index. For convenience, the *condition index* (CI) can also be employed to formulate the manipulability constraint (Angeles 1992):

$$\sigma_{\min} / \sigma_{\max} \geq \varepsilon \tag{9}$$

where, ε is the user defined lower bound of CI for singularity avoidance, which usually takes a small value, 0.001 for instance.

3.3.2 Area of Pose Ruled Surface and its Change Ratios

The area of the robot pose ruled surface with two base trajectory curves P_sP_e and Q_sQ_e can be obtained (Zha 1993, Zha 2002), as follows,

$$A = \int_{P_s}^{P_e} \int_{Q_s}^{Q_e} dA \tag{10}$$

where dA is the differential of area of ruled surface. From Eq (3), the area of robot pose ruled surface A can be further written as

$$A = \int_0^1 d\lambda \int_1^t \left| \dot{\mathbf{r}}_1(t) + \lambda \dot{\mathbf{\Phi}}(t) \right\| \mathbf{\Phi}(t) | dt \tag{11}$$

With respect to time t, the first-order and the second-order change ratios of the pose ruled surface, dA/dt and d^2A/dt^2, can be obtained respectively as

$$dA/dt = \int_0^1 | \dot{\mathbf{r}}_1(t) + \lambda \dot{\mathbf{\Phi}}(t) | \varphi(t) d\lambda \tag{12}$$

$$d^2A/dt^2 = d \left(\int_0^1 | \dot{\mathbf{r}}_1(t) + \lambda \dot{\mathbf{\Phi}}(t) | \varphi(t) d\lambda \right) / dt \tag{13}$$

From Eqs (11-13), both the area of the robot pose ruled surface and its change ratio are functions of the pose trajectory equations, $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$, and the velocity equations, $\dot{\mathbf{r}}_1(t)$ and $\dot{\mathbf{r}}_2(t)$. When a robot operates in a certain speed, the area of the robot pose ruled surface and its change ratios can indicate the kinematics and dynamics performances of the robot manipulator.

3.4 Dynamics Analysis and Control

The acceleration control can be considered as an extension of the velocity control, which controls the robot moving along the given motion parameters by imposing force and/or torque on each joint of robot. Thus, the main task of dynamics control is to determine the generalized force imposing on the robot joint to obtain the given acceleration. This can be achieved from the dynamics equation of the robot, as follows

$$\boldsymbol{\tau} = \boldsymbol{I}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{V} \dot{\mathbf{q}} + f(\mathbf{q}, \dot{\mathbf{q}}) + g(\mathbf{q}) \tag{14}$$

where, $\boldsymbol{\tau}$ is the jont driving force; $\boldsymbol{I}(\mathbf{q})$ is the $n \times n$ inertia matrix; $\mathbf{V}$ is the $n \times n$ damping matrix; $f(\mathbf{q}, \dot{\mathbf{q}})$ is the n- dimensional nonlinear function of certrifugal and Coriolis terms; $g(\mathbf{q})$ is the gravitational terms. Furthermore, the corresponding dynamics equations in the Cartesian space can be derived as follows:

$$\boldsymbol{F}=\boldsymbol{V}(q)\,\ddot{\mathbf{X}}+\boldsymbol{U}(\mathbf{q},\dot{\mathbf{q}})+\boldsymbol{P}(\mathbf{q}) \tag{15}$$

where, $\boldsymbol{F}$ is the generalized operational force in Cartesian space; $\boldsymbol{V}(\mathbf{q})$ is $m\times m$ kinetic matrix, i.e., inertia matrix in Cartesian space; $\boldsymbol{U}(\mathbf{q},\dot{\mathbf{q}})$ is the nonlinear function of certrifugal and Coriolis terms in Cartesian space; $\boldsymbol{P}(\mathbf{q})$ is the gravitational terms in Cartesian space.

As discussed above, the problem of planning robot pose trajectories is equivalent to that of the generation of robot pose ruled surface. It means that the motion locus of configuration can be planned for the robot end effector in task space by generating the robot pose ruled surface. The change laws of robot pose can be determined by fitting or interpolating key or knot pose points obtained from artificial teaching-by-showing or measurement and even by the latest technologies (e.g. data glove) in virtual environment. Thus, the corresponding points of the entire motion path can be calculated by interpolation.

4 Coordination Planning for Robot Trajectories

4.1 Trajectory Generation

When a robot operates in task space, it must meet some requirements and constraints. Constraints for robotic motion trajectories are dependent on that the application requires zero-order (C^0), first (C^1)- or second (C^2)-order continuity, and kinematics or dynamics performances to yield a position and orientation continuous curve. Given a serial manipulator with prismatic and revolute joints operating in task space, consider the class of trajectories in Cartesian space, $\boldsymbol{X}$ (t)=[$\mathbf{r}_1$(t), $\boldsymbol{\Phi}$ (t)]^T => [$\mathbf{r}_1$(t), $\mathbf{r}_2$(t)]^T, which satisfy the following kinematics, control and dynamics constraints and requirements :

(a) The robot end effector is desired to pass through the m specified or given knot pose points $\mathbf{r}_1(t_i)$, $\mathbf{r}_2(t_i)$, i=1,2,3,...m, accurately in workspace;

(b) When the robot moves along the planned trajectory in workspace, its motion is expected to be smooth with C^2 continuity and even small jerk, i.e., the existence of $\dot{\mathbf{r}}_1(t)$, $\ddot{\mathbf{r}}_1(t)$, $\dot{\mathbf{r}}_2(t)$ and $\ddot{\mathbf{r}}_2(t)$, or sometimes, $|\dddot{\mathbf{r}}_1(t)|\le J_{P_{\max}}$, $|\dddot{\mathbf{r}}_2(t)|\le J_{Q_{\max}}$, where, $J_{P_{\max}}$ and $J_{P_{\max}}$ are maximum pose jerks.

(c) The robot must avoid the singularity , W≠0, i.e., det($\mathbf{JJ}^T$)≠0 or CI ($\sigma_{\min}/\sigma_{\max}\ge\varepsilon$);

(d) The robot motion cannot exceed the joint range limits and maximum joint velo1city range, i.e., $|q_{i_{\min}}|\le q_i\le|q_{i_{\max}}|$, $|\dot{q}_i|\le\dot{q}_{i_{\max}}$.

(e) The robot motion cannot exceed the maximum joint acceleration and the maximum joint driving forces / torques, i.e., $|\ddot{q}_i|\le\ddot{q}_{i_{\max}}$, $|\tau_i|\le\tau_{i_{\max}}$, ($i$=1,2,3,…,$n$)

(f) The robot must have better kinematics performances, e.g., maximum velocity-space or shortest motion path, or minimum area of pose ruled surface, or minimum change ratios of area of ruled surface.

There are many methods for fitting or interpolating key pose points to generate robot motion trajectories. These include segment interpolation (e.g. straight-line segment and transition segment), polynomial curves, cubic curves (e.g., Hermite cubic splines, Bezier curve, B-spline, NURBS), and so on (Boehm 1985, Patel and Lin 1988, Dierckx 1993, Ge and Ravani 1994, Ge and Kang 1995, Gerald Farin 1991). Suppose that P_sP_e and Q_sQ_e are formulated by one of the methods mentioned above, as shown in Figure 2, and they are described as follows:

$$\begin{cases} \mathbf{P} : \mathbf{r}_1(t) = x_1(t)\mathbf{i} + y_1(t)\mathbf{j} + z_1(t)\mathbf{k} \\ \mathbf{Q} : \mathbf{r}_2(t) = x_2(t)\mathbf{i} + y_2(t)\mathbf{j} + z_2(t)\mathbf{k} \\ \mathbf{\Phi} : \mathbf{\Phi}(t) = \mathbf{r}_2(t) - \mathbf{r}_1(t) \end{cases} \tag{16}$$

where, $t \in [t_1, t_2]$. This means that both the position and orientation of robot vary from P_s to P_e and from Q_s to Q_e on the curves, respectively. If $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ or $\mathbf{r}_1(t)$ and $\mathbf{\Phi}$(t) are determined, the trajectory planning process is accomplished, and the coordinates of corresponding points on the pose trajectories can be calculated.

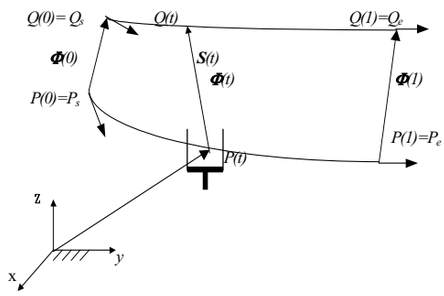


Fig. 2. Robot trajectory generation with constraints.

Here, we discuss the case when parametric cubic splines are used to interpolate pose data (Zha 2002). The pose trajectory curves, P_sP_e and Q_sQ_e, are assumed to be k-order space polynomial curves which can be explicitly expressed as

$$\begin{cases} \mathbf{P} : \mathbf{r}_1(t) = (\sum_{i=0}^{k} a_{1i}t^i, \sum_{i=0}^{k} b_{1i}t^i, \sum_{i=0}^{k} c_{1i}t^i)^T \\ \mathbf{Q} : \mathbf{r}_2(t) = (\sum_{i=0}^{k} a_{2i}t^i, \sum_{i=0}^{k} b_{2i}t^i, \sum_{i=0}^{k} c_{2i}t^i)^T \\ \mathbf{\Phi} : \mathbf{\Phi}(t) = [\sum_{i=0}^{k} (a_{2i} - a_{1i})t^i, \sum_{i=0}^{k} (b_{2i} - b_{1i})t^i, \sum_{i=0}^{k} (c_{2i} - c_{1i})t^i]^T \end{cases} \tag{17}$$

where, (a_{1i}, b_{1i}, c_{1i}) and (a_{2i}, b_{2i}, c_{2i}) $(i=0,1,2,...,k)$ are polynomial coefficients of 3D coordinates of $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ respectively. The pose trajectory curves can also be represented as matrix equation as follows (Zha 2002)

$$\begin{bmatrix} \mathbf{r}_1(t) \\ \mathbf{r}_2(t) \end{bmatrix} = C_{P-Q}T = \begin{bmatrix} a_{10} & a_{11} & a_{12} & \cdots & a_{1k} \\ b_{10} & b_{11} & b_{12} & \cdots & b_{1k} \\ c_{10} & c_{11} & c_{12} & \cdots & c_{1k} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2k} \\ b_{20} & b_{21} & b_{22} & \cdots & b_{2k} \\ c_{20} & c_{21} & c_{22} & \cdots & c_{2k} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \\ \vdots \\ t^k \end{bmatrix} \tag{18}$$

where, $C_{P\text{-}Q} \in R^{6\times(k+1)}$ is the polynomial coefficients matrix to be determined, and $T = (1, t, t_2, ..., t_k)^T$. Substituting Eq.(17) or Eq. (18) into Eqs (10-13), the area function and its change ratios can be determined. When the control points and constraints are determined, the polynomial coefficients, (a_{1i}, b_{1i}, c_{1i}) and (a_{2i}, b_{2i}, c_{2i}) $(i=0,1,2,...,k)$, i.e., the coefficient matrix $C_{P\text{-}Q}$,

can be obtained through the corresponding fitting or interpolating algorithms, and then the pose trajectories can therefore be obtained.

4.2 Trajectory Optimization

The optimal trajectory planning of robot manipulator is to formulate a task-oriented optimization model that consists of three parts: trajectory design parameters, objective function, and constraints. Trajectory design parameters and constraints are discussed for trajectory generation above. The objective function is related to performance evaluation, i.e., "goodness" of a robot trajectory. The straightforward way to define the objective function is to select one of performance measures or some of them in a weighted sum manner such as manipulability, reachability, joint and/or velocity range or space availability, motion jerks, and even the area of pose ruled surface and it change ratios.

Based on the above discussions, the area of robot pose ruled surface and its change ratios can indicate the kinematics performance. During the course of trajectory planning or task planning, the manipulability control must be considered. Therefore, the objective function can be defined as the area of pose ruled surface with good performance

$$F= \int_{P_s}^{P_e} \int_{Q_s}^{Q_e} \frac{1}{PI} dA \tag{19}$$

where,; P_s and P_e are the starting point and end point of position trajectory respectively; Q_s and Q_e are starting point and end point of orientation trajectory respectively; dA is a differential of area of ruled surface; PI is performance index, W, manipulability measure. Based on Eq.(11), Eq. (19) can be rewritten as

$$F= \int_0^1 d\lambda \int_{l_1}^{l_2} \frac{1}{PI} \left| \dot{\mathbf{P}}[q(t)] + \lambda \dot{\mathbf{\Phi}}[q(t)] \right\| \mathbf{\Phi}(t) \left| dt \right. \tag{20}$$

After all the optimization variables or coefficients are determined, the optimal pose trajectories are obtained. The models discussed above are suitable for the generation of not only the position but also orientation trajectory of robot.

In some cases, a simplified model could be used. For example, for the position trajectory planning, λ =0 and the area differential dA of robot ruled surface is changed into arc length differential dS, and then the objective function becomes the shortest path, which can be specified and simplified as,

$$F= \int_{P_s}^{P_e} \frac{1}{PI} dS = \int_{l_1}^{l_2} \frac{1}{PI} \left| \dot{\mathbf{P}}(t) \right| dt \tag{21}$$

The objective function can also be the sum weighted distance as described in Section 6, The optimal trajectories can be found with a commercial optimization software package such as MATLAB optimization toolbox (1998) in which the most two typical optimization methods, the grid method and the stochastic constraint method, were employed. Details about the simulation and optimization will be discussed in Sections 5 and 6.

5. Coordination Control for Robot Trajectories

In order to control robot conveniently, curve length variables l_1 and l_2 are often selected as the path parameters of P_sP_e and Q_sQ_e,. In general case, coordinates of corresponding points on the robot pose trajectories can be determined by interpolation algorithms, as follows:

$$\begin{cases} \mathbf{P}: \mathbf{r}_1(l_1) = x_1(l_1)\mathbf{i} + y_1(l_1)\mathbf{j} + z_1(l_1)\mathbf{k} \\ \mathbf{Q}: \mathbf{r}_2(l_2) = x_2(l_2)\mathbf{i} + y_2(l_2)\mathbf{j} + z_2(l_2)\mathbf{k} \\ \mathbf{\Phi}: \mathbf{\Phi}(l_1, l_2) = \mathbf{r}_2(l_2) - \mathbf{r}_1(l_1) \end{cases} \tag{22}$$

where, l_1=l_1(t), l_2=l_2(t). This means that robot pose trajectories can be controlled by path variables, l_1 and l_2, which obey a specified motion laws. The conditions to be satisfied for the coordination control of pose trajectories can be derived as follows,

$$\Delta l_1 = \sqrt{\Delta x_1^{\,2} + \Delta y_1^{\,2} + \Delta z_1^{\,2}}\ ,\ \Delta l_2 = \sqrt{\Delta x_2^{\,2} + \Delta y_2^{\,2} + \Delta z_2^{\,2}} \tag{23}$$

$$l_1 = l_1(t) = \int_1 \Delta l_1 dt\ ,\ l_2 = l_2(t) = \int_1 \Delta l_2 dt\ ,\ \begin{cases} p_s = 0 \\ p_e = \int_1^2 \Delta l_1 dt \end{cases},\ \begin{cases} q_s = 0 \\ q_e = \int_1^2 \Delta l_2 dt \end{cases} \tag{24}$$

As discussed above, the trajectory coordination and the calculation of corresponding pose point coordinates are dependent on and controlled by the motion laws of path parameters, such as uniform motion, constant acceleration, uniform and constant deceleration motion, etc. Two typical motion laws of path parameters were discussed in (Zha and Chen 2004). According to Eq.(22), the corresponding pose coordinates can be calculated for each pair of possible curves, such as line-line, line-arc, arc-line, arc-arc, high-order polynomials, etc. Consequently, the pose and its velocities and accelerations at any time can be determined or controlled. The problem determining the control laws of pose in Cartesian coordinate space is thus solved.

6. Planning and Control Simulation

To verify the proposed models above, simulation for coordinated planning and control should be carried out in an integrated environment. In this section, the simulation of the proposed approach is discussed.

6.1 Simulation Process

Using an optimization method, the optimal trajectory planning can be fulfilled (Zha and Du 2001). After the pose trajectories or the pose ruled surface are determined, the system calculates the corresponding position and orientation point coordinates based on the specified motion laws of path parameters. Finally, the system carries out the motion animation and outputs the joint angles for robot controller, and the simulation process is thus finished. The flowchart of the planning and simulation process in an integrated environment can be described in Figure 3.

The simulation environment is developed using Robot Toolbox (Corke 1999), Spline Toolbox and Optimization Toolbox, which are all based on the MATLAB package (1999). The task evaluation consists of two parts: performance constraint detection and fitness value computation. If a pose trajectory or a pose ruled surface can satisfy all the kinematics, dynamics and control constraints such as reachability, joint range availability, and manipulability, joint torques, discussed in Section 3, its fitness can be calculated by Eq.(23); otherwise the fitness is assigned to an infinite large number indicating such a trajectory or a pose ruled surface is infeasible.

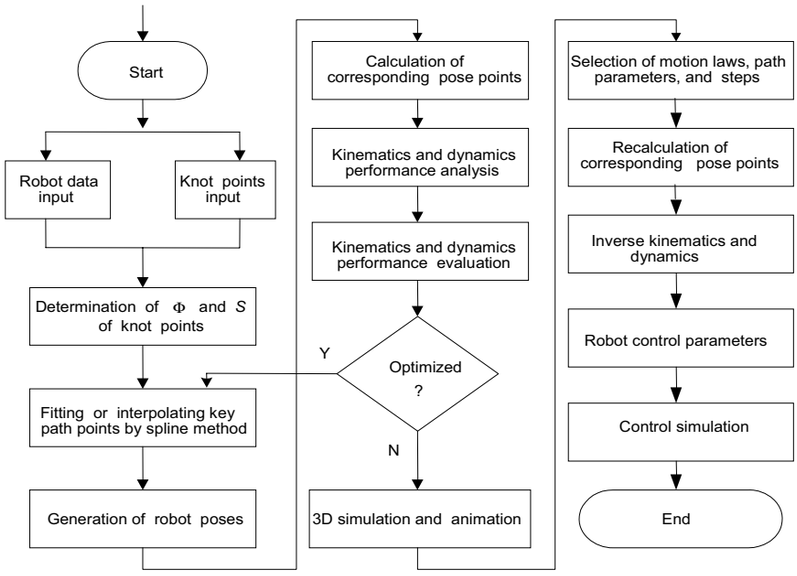


Fig. 3. Flowchart of robot trajectory planning and control simulation.

5.2 Two Simulation Examples

The first example is a 3-DOF planar robot manipulator (Zha 2002), as shown in Figure 4. The link length of this mechanism is represented by l_1, l_2, and l_3, where l_1=l_2=0.400, and l_3=0.200, respectively. The robot end effector is supposed to move along a trajectory in task space from P_s (-0.400,0.200,0)^T to P_e(0.400,0.200,0)^T with a constant orientation and operation force defined as Φ =(0,0,-$\frac{\pi}{2}$)^T and F= (1,1,1)^T, respectively. The motion time is required to be within T=60 seconds, which is the same as the time taken by the robot end effector to move from the start point to the end point. From the configuration of robot and geometric relationships, the position trajectory in workspace can be given by

$$r_1(t) = (l_1c_1+l_2c_{12}+l_3c_{123}, l_1s_1+l_2s_{12}+l_3s_{123}, 0)^T \tag{25}$$

where, c_1=cosθ_1,s_1=sinθ_1, c_{12}=cos(θ_1+θ_2), s_{12}=sin(θ_1+θ_2), c_{123}=cos(θ_1+θ_2+θ_3), s_{123}= sin(θ_1+θ_2+θ_3), and $\theta_1+\theta_2+\theta_3=\frac{3\pi}{2}$. By derivative of Eq (25), the following is obtained

$$\dot{r}_1(t)=[-l_1\dot{\theta}_1 s_1-l_2(\dot{\theta}_1+\dot{\theta}_2)s_{12}, -l_1\dot{\theta}_1 c_1+l_2(\dot{\theta}_1+\dot{\theta}_2)c_{12}, 0]^T = J\dot{q} \tag{26}$$

where, $\dot{q} = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)^T$, and J is a Jacobian matrix as

$$J = \begin{bmatrix} -l_1s_1 - l_2s_{12} & -l_2s_{12} & 0 \\ l_1c_1 + l_2c_{12} & l_2c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

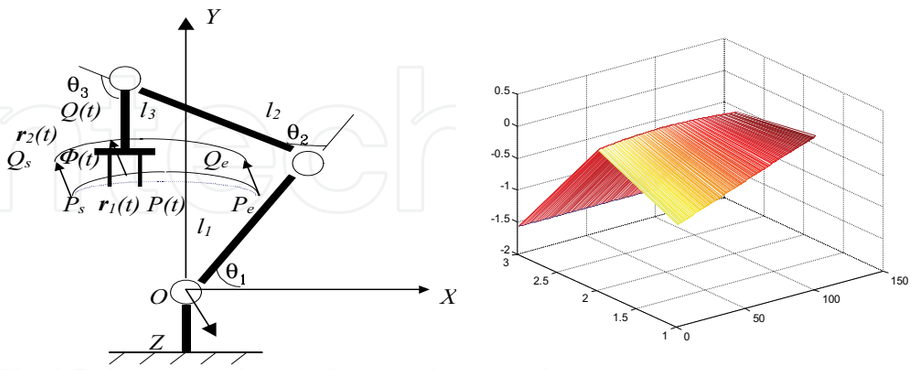


Fig. 4. Trajectory planning for a 3-dof manipulator (with redundancy).

According to Eq (27), the manipulability measure of robot can be written as

$$W=\sqrt{\det[JJ^T]}=|\det J|=l_1 l_2|s_2| \tag{28}$$

The pose trajectories P_sP_e and Q_sQ_e for the manipulator are supposed to be 6-order space polynomial curves respectively, i.e., as follows

$$r_1(t)=(\sum_{i-0}^{6} a_{1i}t^i,\sum_{i=0}^{6} b_{1i}t^i,0)^{\mathrm{T}} \tag{29}$$

$$r_2(t)=(\sum_{i-0}^{6} a_{1i}t^i,\sum_{i=0}^{6} b_{1i}t^i,-\pi/2)^{\mathrm{T}} \tag{30}$$

where, $t\in[0,1]$ are corresponding to P_s and Q_s ;P_e and Q_e respectively. Thus, the optimization objective function or fitness function can be chosen as

$$\mathrm{F}=\frac{\pi}{2l_1l_2}\int_0^1\frac{1}{|s_2|}\sqrt{(\sum_{i=1}^{6} ia_{1i}t^{i-1})^2+(\sum_{i=1}^{6} ib_{1i}t^{i-1})^2}\,dt \tag{31}$$

Based on the fact that t=0 and t=1 is corresponding to P_s ,Q_s and P_e and Q_e respectively, the optimization constraints are expressed as

$$a_{10}=-0.400,\quad b_{10}=0.200,\quad \sum_{i=0}^{6} a_{1i}=0.400,\sum_{i=0}^{6} b_{1i}=0.200$$

Using the traditional optimization method, stochastic constraints in MATLAB optimization toolbox or the genetic algorithm, the coefficients (a_{1i},b_{1i}) (i=0,1,2,...,6) for the optimal trajectory P_sP_e with the shortest path and maximum flexibility can be obtained (Table 1).

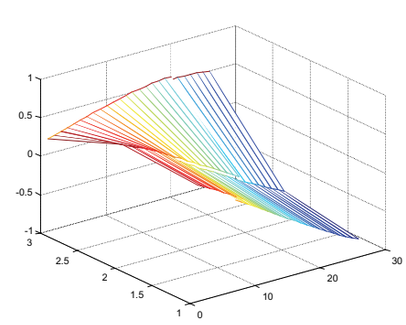
i	0	1	2	3	4	5	6
*a_{1i}	-0.4000	0.7903	-0.0022	0.1673	-0.2364	0.07275	-0.0083
*b_{1i}	0.2000	0.2770	-0.2544	0.0483	-0.2511	0.1826	-0.0023

Table 1. Optimal coefficients of trajectory equations for 3-DOF robot manipulators.

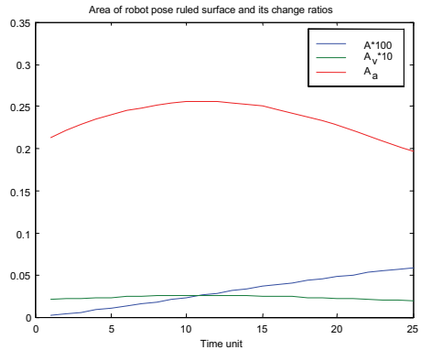
The second example is for PUMA 560 robot used for arc welding (Zha 2002). Assume that the end effector of robot PUMA 560 is required to move along a trajectory passing through the configuration points in task space. These key pose points are supposed to be fitted by 6-order space polynomial curves. Using the proposed model and approach, the optimal polynomial coefficients for the trajectory equations satisfying the kinematics and dynamics constraints, with the minimum path or pose ruled surface area and maximum flexibility, can be obtained. Table 2 lists the optimized coefficients for the trajectories. Figures 5-7 demonstrate the simulation and animation results for PUMA 560 robot moving along the optimized trajectories.

i	0	1	2	3	4	5	6
$*a_{1i}$	0.8000	0.0000	-3.7909	48.2443	-167.4964	198.5294	-79.3606
$*b_{1i}$	0.4000	0.0000	16.8877	-54.3584	39.1373	17.4540	-19.8360
$*c_{1i}$	0.2000	0.0000	-15.4005	66.5505	-103.0331	63.3715	-14.2356
$*a_{2i}$	0.6000	0.0000	17.5912	-58.9944	28.1370	43.7332	-32.8612
$*b_{2i}$	0.4000	0.0000	15.0318	-28.7452	-49.2458	121.6272	-60.6053
$*c_{2i}$	0.4000	0.0000	-25.8490	101.9592	-126.8347	48.5275	0.8091

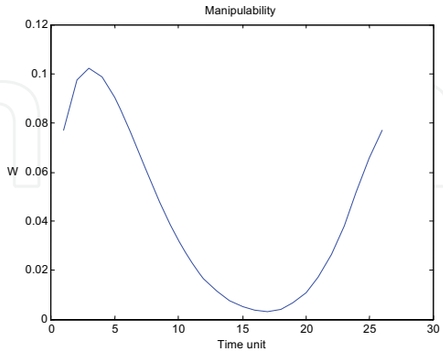
Table 2. Optimal coefficients of trajectory equations for PUMA 560 robot.



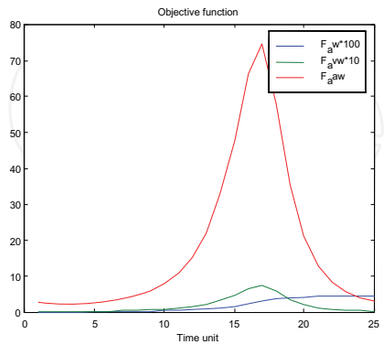
(a) Robot pose ruled surface



(b) Area of pose ruled surface and its change ratios

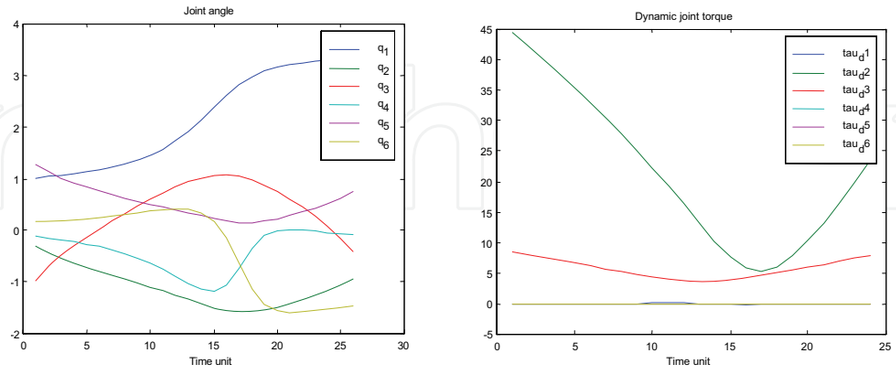


(c) Manipulability



(d) Objective function values (PI=W)

Fig. 5. The optimized performance indexes and objective functions for PUMA 560 robot.



(a) Joint angle (b) Joint torque

Fig. 6. Kinematics and dynamics analysis for PUMA 560 robot moving along the optimized trajectories.

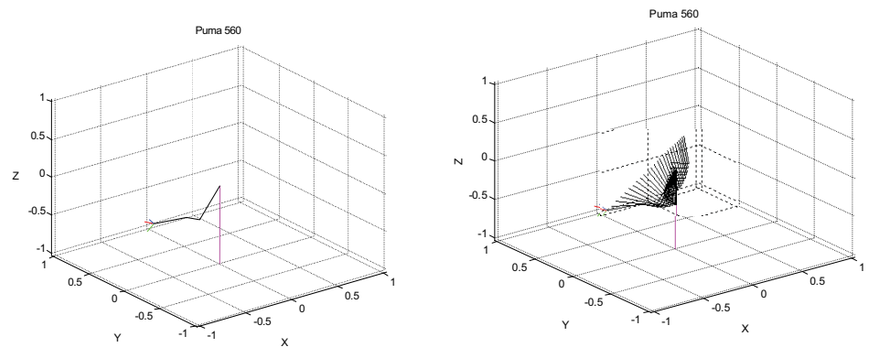


Fig. 7. Motion animations for PUMA 560 robot moving along the optimized trajectories.

7. Industrial Application for Robotic Polishing

7.1. Profile Reconstruction for Distorted Surface

One challenging task for robotic applications lies in precise materials surface processing. A robot is required to mimic an operator to manipulate a processing tool to remove materials from free-form surface. Fig. shows the schematics of a high pressure turbine (HPT) vane. The vane consists of an airfoil having concave and convex surfaces, an inner buttress and an outer buttress. After operating in a high-temperature and high-pressure environment, vanes incur severe distortions as large as 2 mm in reference to the buttress. On the airfoil surface there are hundreds of cooling holes. After a number of operational cycles, defects such as fully or partially blocked cooling holes, micro cracks and corrosions begin to occur. Because of the high cost of the components, it is common practice to repair these parts instead of scrapping them. The repairing process starts with cleaning and covering the defective areas with the braze material. The purpose of brazing is to fill up the defects, but unavoidably, the brazed areas will be higher than the original surface.

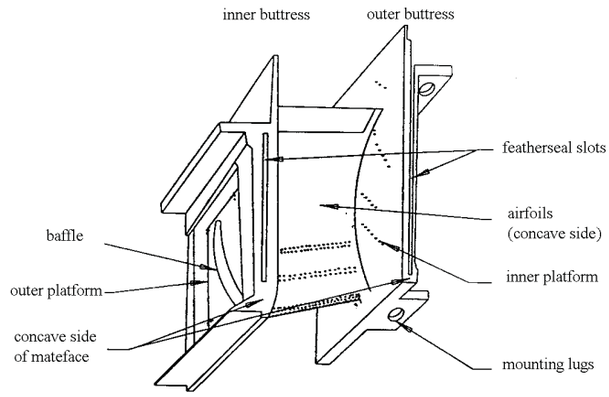


Fig. 8. Schematics of a HPT Vane.

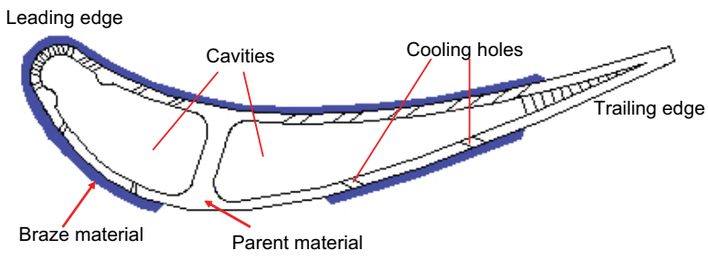


Fig. 9. Turbine airfoils repaired with braze material.

Fig. shows a cross section of the airfoil brazed with a repair material. The grinding and polishing process is intended to remove the excessive materials and make the brazed area flush with the original surface within a tight tolerance about 100 µm. Current manual polishing uses belt machine to remove the braze without undercuts and overcuts. For a robot to carry out such a delicate task, dexterous motion and control is needed to achieve compliant contact and contact force between airfoil and contact wheel.

The turbine airfoils to be repaired have severe distortions and twists after operations in the high-temperature and high-pressure environment. A teach-and-play robot cannot cope with the distorted profile. Neither can a commercial off-line programming system, which generates robot path according to the design data. It is absolutely critical and necessary to have profile sampling and distortion compensation system in this specific application to deal with part-to-part variations. Before any distortion compensation, the actual profile has to be sampled. A Linear Variable Differential Transducer (LVDT) has been integrated into the robotic system to carry out profile measurement. After gripping the part, the robot approaches the measurement probe, as shown in Fig. 10.

The sensor has good resolution and accuracy and reliability. As compared with off-line measurement, in-situ measurement enables the robot to act as a measurement instrument. Consequently, common fixturing and datum can be used for both measurement and polishing. This is advantageous in minimising fixture errors.

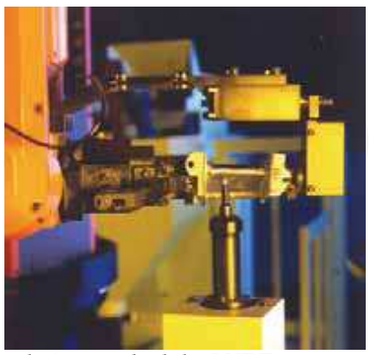


Fig. 10. Robot holding the workpiece probed the LVDT.

Selected points on the vane airfoil, many of which are covered by the brazing material, are sampled by the sensor. Approximation is made to offset the measured points to the prior-to-braze airfoil surface. The sensory readings only give the displacements in Z axis. In order to obtain the true coordinates of the measured points, corresponding robot coordinates have to be used for computation in conjunction with displacement readings.

The Optimal Profile Fitting (OPF) algorithm (Chen, X.Q. et al, 2002) fits a template to the actual measurement points with minimum sum of errors. The sectional template profiles are established based on design data using Cubic Spline Interpolation. It ensures that not only the interpolation is continuously differentiable (C^0, C^1) on the interval, but also has a continuous second derivative (C^2) on the interval.

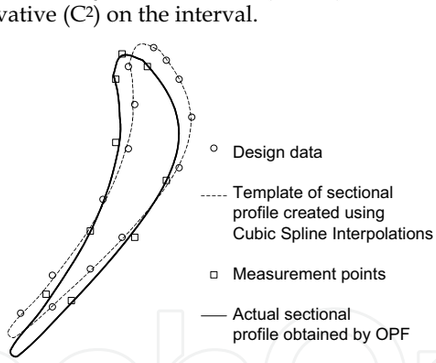


Fig. 11. Actual 2D sectional profile obtained by OPF.

The measurement data are first transformed from global coordinate system to local (tool) coordinate system. The OPF is carried out in the tool coordinate system on each 2D cross sectional profile. Three 2D sectional profiles are taken as templates to do the fitting. The template for each section is the design profile at the respective section, which is obtained through cubic spline interpolations. We assume that a template is a rigid 2D profile. Therefore there are three degrees of freedom in the optimal fitting, namely X axis shift, Y axis shift, and rotation around a certain point. Fig illustrates the concept of OPF.

Each cross section is computed individually. The complete 3D airfoil profile is obtained through interpolating the cross sections**.** The goal of the optimal fitting is to find the optimal X-Y shift and

rotate values for the template, so that after the transformation, the sum distance from the template to the measures points at the given height is minimum. In other words, the optimal fitting problem is a multi-dimensional minimization problem and the sum distance is used as the performance index function in the minimization process. We have developed and implemented the Search-Extend-Focus (SEF) minimization algorithm (Chen, X.Q. et al, 2002). SEF finds the optimal point at which the index function is minimum.

By varying the definition of the performance index function (i.e. the sum distance), different optimal fitting results can be obtained. Here, the following definition of the sum distance is used:

$$d_{sum} = w_h h^2 + \sum w_i d_i^2 \tag{32}$$

where d_{sum} is the sum distance defined, d_i is the distance from ith measure point to the profile, h is distance from the given height to the profile, w_i and w_h are the respective weights in the sum distance. When different weights w_i and w_h for different measure points are chosen, different priorities are given to different measure points. By choosing high weights for groups of measure points in turn, multiple optimal fitting can be carried out for one sectional optimal fitting. The weights in the optimal profile fitting are given in Table 3.

	Weight for Convex Side Measure Points					**Weight for Concave Side Measure Points**					**Weight for height**
	1	2	3	4	5	6	7	8	9	10	
Convex Fitting	10.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0	1.0	1.0	10.0
Concave Fitting	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	10.0	10.0

Table 3. Weights for Measure Points and Height in Optimal Fitting.

Note: Measure Point 1 and 10 are on the trailing edge of the profile without brazing material on the surface, and the other measure points are all on the profile with brazing material on the surface.

Combining the multiple optimal fitting results, the following fitting objectives are achieved:

1. All portions of the profile are optimally fitted to the measure points.
2. Whole profile is smooth.
3. No distortion of the profile shape at each portion. Minimum distortion exits only at adjacent areas of two portions of the profile for smooth transition.

7.2. Robot Polishing Path Planning for 3D Robotic Surface Finishing

Having an accurate description of the airfoil profile is not the ultimate aim. The computed profile based on the sensory data must be used to automatically generate the robotic polishing path. This process has to be repeated for every part.

The robot polishing path is a point-to-point motion in the Cartesian coordinate system. With the Euler angle computation and the coordinate transformation, the robot end-effector location (position and orientation) in Cartesian coordinates are generated from the contact points in the tool coordinate system. Along a curve between any two points, the robot automatically moves using the cubic spline line motion. Thus, we only concern with the formalism of deriving the robot coordinates (X, Y, Z, A, B, C) of the path points which the robot must travel along in the Cartesian coordinate system. The coordinates of every path point are calculated based on the robot kinematics model. In addition, certain system setup

parameters, such as polishing wheel size and global locations of all tool heads, have been built into the model. With this feature, intuitive re-calibration of tooling can be done rapidly and accurately. Fig. 12 shows one path point computed by ARP.

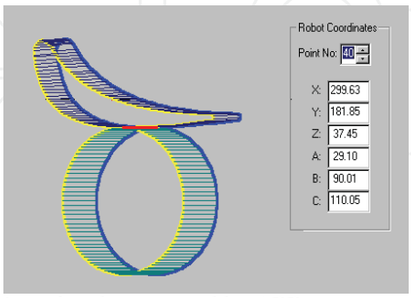


Fig. 12. Computer simulation of path generated by ARP.

The space curve that the robot end-effector moves along from the initial location (position and orientation) to the final location is called the *path* (Fu, K. S.; Gonzalez, R. C. & Lee, C. S. G, 1987). It deals with the formalism of describing the desired the robot end-effector motion as sequences of points in space (position and orientation of the robot end-effector) through which the robot end-effector must pass, as well as the space curve that it traverses.

Path planning scheme generates the desired path by a series of points (the endpoints and intermediate points) in Cartesian coordinates. They are specified in Cartesian coordinates not in joint coordinates because it is easier to visualize the correct configurations in Cartesian coordinates than in joint coordinates. In this work, the robot polishing path is a point-to-point motion in Cartesian coordinate system. With the Euler angle computation and the coordinate transformation, the robot end-effector location (position and orientation) in Cartesian coordinates can be generated from the local coordinates of the polishing path points on the surface of work piece in robot end-effector's coordinate system for polishing process. Each path knot point of the robot end-effector is described by six robot coordinates (*X, Y, Z, A, B, C*), where coordinates (*X, Y, Z*) specify the robot end-effector position and coordinates (*A,B,C*) specify the robot end-effector orientation.

For the space curve between any two points, the robot automatically moves using the cubic spline line motion. Thus, we only concern with the formalism of deriving the robot coordinates(*X, Y, Z, A, B, C*) of the points which the robot must .traverses in Cartesian coordinate system.

There are many different types of Euler angle representations. The scheme we adopted is illustrated in Fig. 1. The resultant Euler rotation matrix is given by:

$$R_{A,B,C} = R_{Z,A}R_{Y',B}R_{Z'',C} = \begin{bmatrix} \cos A\cos B\cos C - \sin A\sin C & -\cos A\cos B\sin C - \sin A\cos C & \cos A\sin B \\ \sin A\cos B\cos C + \cos A\sin C & -\sin A\cos B\sin C + \cos A\cos C & \sin A\sin B \\ -\sin B\cos C & \sin B\sin C & \cos B \end{bmatrix} \quad (33)$$

With the above defined Euler angle rotation matrix, the orientation of the robot end-effector can be derived with respect to the reference global coordinate system. Then plus the translation of the robot end-effector, the exact location of the robot end-effector (path point) after moving robot coordinates (*X,Y,Z,A,B,C*) in global coordinate system is developed.

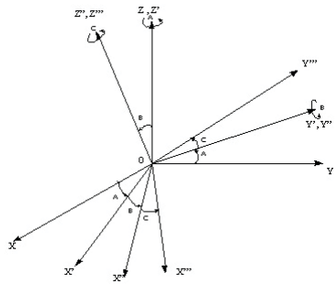


Fig. 13. Euler angles system.

A robot path is a sequence of points to polish the surface of the work. Each point can be described in terms of robot coordinates (*X, Y, Z, A, B, C*) which can be recognized by the robot controller. An illustration about robot path generation is given in Fig. 1. There are four 3D coordinates systems:

(1) Coordinate system A: This is the global coordinate system.
(2) Coordinate system B: Robot end-effector coordinate system.
(3) Coordinate system C: Tool coordinates system.
(4) Coordinate system D: Part coordinate system constructed for each polishing position as shown in Fig. 15.

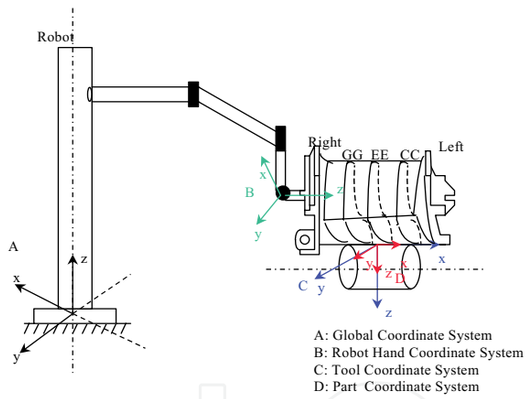


Fig. 14. Robot Polishing Path Generation.

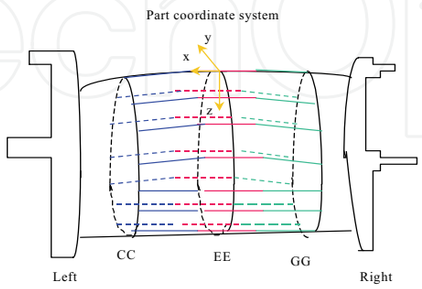


Fig. 15. Part coordinate system.

After the part's 3D profile is obtained by using the Optimal Template Fitting Method, the part coordinates system is constructed on the 3D profile for each polishing point. The part and tool coordinate systems are constructed such that, for each polishing point, the part is at desired polishing position when the two coordinate systems are overlapped. Based on this, we can derive the position of robot end-effector's coordinate system B by Coordinate System Transformation, and further more, the robot coordinates (*X, Y, Z, A, B, C*).

The robot path generation is the inversion process of coordinates transformation by translation and Euler angles rotation. The following is the procedure to compute the robot coordinates of a polishing point:

Step 1. Compute the polishing points in robot end-effector coordinate system.

Step 2. Construct a part coordinate system for a polishing point.

Step 3. Assume that the part coordinate system is overlapped with tool coordinate system, compute the robot end-effector coordinate system by using the coordinates system transformation.

Step 4. From the position of robot coordinate system in the global coordinate system, derived the robot coordinates (*X, Y, Z, A, B, C*).

Step 5. Repeat Step 2 to Step 4 to obtain a series the robot coordinates (*X, Y, Z, A, B, C*).

Through above steps, a series of robot coordinates (*X, Y, Z, A, B, C*) are developed for each desired polishing positions, which forms the robot polishing path.

7.3. Integrated Mechatronic Control of Robotic Surface Finishing

In robotic applications based on position control, path information is sufficient. In the sophisticated constrained application like surface finishing, contact force, compliance and tool wearing become indispensable. Process model including polishing tool geometry, contact stiffness, pre-load, Z displacement, robot travel speed and direction, etc. have been obtained from extensive laboratory prototyping on process optimization. These optimum process parameters are encapsulated in the process knowledge base. Operational parameters such as Z-axis offset, robot travel speed are associated with each path point to generate desired airfoil profile. Tool wear compensation is automatically done by associating process properties to the polishing path. Process development and optimization (Huang, H. et al, 2002) is critical to achieving an integrated mechatronic solution to the 3D polishing application.

Fig. 16 shows the architecture of the Knowledge-Based Adaptive Robotic System for 3D profile polishing. It comprises three inter-related hierarchical layers, namely, Device and Process, Knowledge-Based Process Control (KBPC), and Data-Driven Supervisory Control (DDSC). The Supervisory Controller is driven by product and process data including scheduling, system configuration, product design data, cross sectional data (template), and tool coordinates.

The Device/Process Control sub-system controls actuators and sensors to fulfil the required processes, and coordinates the process flow. In addition, it also acquires measurement data and exchanges data and information with the Supervisory Controller. Process Control relies on the following process knowledge bases:

Historical Process Database. It holds records of individual parts, such as measurement data, processing time, measurement data of finished profile (optional), breakdowns, uptime, and downtime.

Tool Compensation Knowledge Base. It stores tool compensation parameters such as the abrasive belt speed and the workpiece feed rate.

Path Optimisation Knowledge Base. It contains optimum process parameters such as Z-axis offset, approaching angle, robot travel speed.

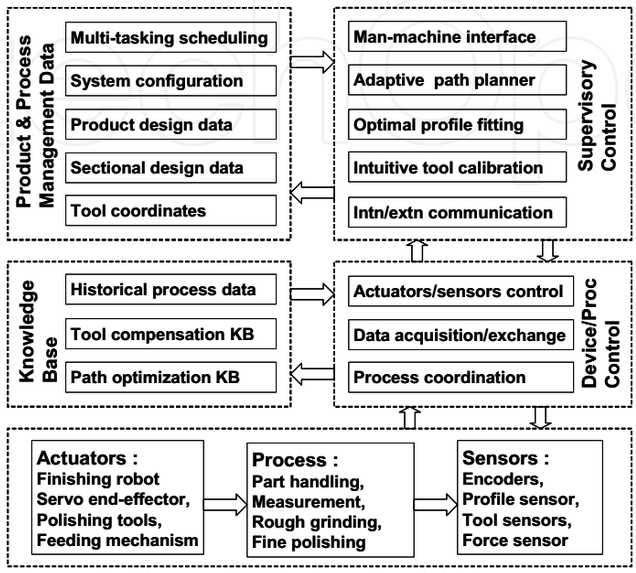


Fig. 16. Control System Architecture for robotic surface finishing.

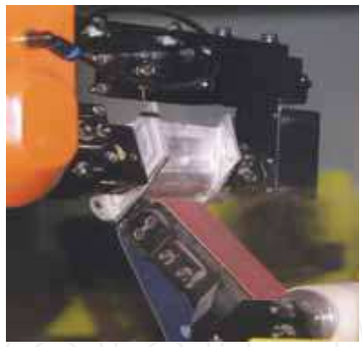


Fig. 17. Robot holding workpiece in contact with the polishing belt.

Figure 17 shows Robot holding workpiece in compliant contact with the polishing belt. It carries out polishing with the planned tool path and optimum process parameters. Optimum parameters can be inferred based on the part conditions: curvature, braze thickness, leading edge height, and trailing edge distortions. As a result, a smooth finish profile, free of transition lines, overlapping marks and burning marks, can be obtained.

The Supervisory Controller contains the following control functions:

- **Internal / External Communication**. Internal communication involves information and data exchange between control modules in DDSC. External communication allows data exchange with KBPC.

- **Intuitive Tool Calibration (ITC)**. During machine re-calibration, set-up or re-installation, position data of tool stations, measurement stations, and index table can be manually clocked. These data are keyed into the database. The mathematical model of workspace and robot kinematics is automatically generated.
- **Optimal Profile Fitting (OPF)**. It generates the actual profile based on the in-situ measurement data. The robust fitting algorithm uses the sectional data as templates, and maps them with the measurement points. 3D free form surface is generated through interpolation of cross sectional profiles.
- **Adaptive Path/Strategy Planner (ARP)**. It automatically generates the optimum tool path based on individual part conditions, and furthermore synthesises the robot programs from the resultant path points.
- **Human-Machine Interface (HMI)**. It allows the user to change system parameters, configure system, select product configurations, enter data, and make queries.

Through adaptive robot polishing path planning and knowledge based process control, very smooth airfoil finishing profiles were achieved by. Vanes before and after polishing are shown in Fig. 18. Further visual inspection shows no visible transition lines from the non-brazed area to the brazed one, no visible polishing marks in the cutting path overlap areas, and no burning marks. The curvature transition from the concave to convex airfoil is very smooth and more consistent than one generated by manual polishing.

(a) (b)

Fig. 18. Vanes before (a) and after (b) robotic grinding and polishing.

8. Summary and Conclusions

This chapter presented a unified approach to robot trajectories coordination planning and control in Cartesian space. The unified treatment of the robot end-effector's position and orientation is based on the pose ruled surface concept and used in trajectory interpolations. The generation and control of pose trajectories for robot end effector could be carried out by generating the 3D robot pose ruled surface or path and determining the minimum or maximum value of its area function with kinematics and dynamics constraints.

The models and algorithms as demonstrated in this chapter are generally reliable and effective if control parameters are appropriately selected. Comparing with existing methods, the proposed planning and control models and algorithms are more generic, and can be used as an alternative to reduce the computing cost and storage in robot

planning and control. During the planning and control of pose trajectories, the position, velocity and acceleration, and even the jerk and the torque limits of the robot joints are taken into consideration as constraints with smooth functions of time. It can be ensured that the robot moves along the planned trajectories with good kinematics and dynamics performances. Thus, the obtained trajectories are efficient both to compute and to execute, and the planned motions are uniform and can rule out bang-bang trajectories. The simulation and experimental study in arc welding and surface finishing process show that the planned trajectories are optimal with respect to the performance indexes, and can be realized through animation without violating any constraints of the robot actuators. Potentially, the proposed model and approach are also useful for computer graphics, CAD-based off-line programming, CAD/CAM (e.g., CNC, wire-EDM) in virtual/real design environments.

9. Disclaimer

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors. No approval or endorsement for any commercial product, service or company by the National Institute of Standards and Technology is intended and implied.

10 References

Angeles, J. (1992). The design of isotropic manipulator architectures in the presence of redundancies, *International. Journal of Robotics Research*, 11(3), pp.196-201

Barr, A., Curin, B. Gabriel, S. and Hughes, J. (1992). Smooth interpolation for orientations with angular velocity constraints using quaternions, *Computer Graphics*, Vol.26, pp.313-320

Bokeberg,E.H., Hunt, K.H. and Ridely, P.R. (1992). Spatial motion-I: acceleration and the differential geometry of screws, *Mech. Mach. Theory*, Vol.27, No.1, pp: 1-15

Bobrow, J.E., Dubowski, S. and Gibson, J.S. (1985). Time-optimal control of robotic manipulators along specified paths, *International Journal of Robotics Research.*, Vol.4, No.3, pp.3-17

Boehm,W. (1985). Curvature continuous curves and surfaces, *Computer-Aided Geometric Design*, Vol.2, pp.313-323

Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*, Addison Wesley Longman, ISBN 0-201-09528-9, Singapore.

Chen X. Q., Gong Z. M., Huang H., Ge S. S. , and Zhu Q. (1999). Development of SMART 3D polishing system, *Industrial Automation Journal*, Vol. 8, no. 2, pp. 6-11

Chen X. Q., Chang H., Chan K., Fong A. M., Kwek T. Y., Kerisnan B., Tan C. S., Lim P. H. (1998). Design and implementation of multi-sensory autonomous welding system for high quality precision welding of large boxed structure, *Proceedings of the Fifth International Conference on Control, Automation, Robotics and Vision*, pp.15-19, Singapore

Chen, X. Q.; Gong, Z. M.; Huang, H.; Ge, S. Z. & Zhou, L. B. (2002). Adaptive robotic system for 3D profile grinding and polishing, In: *Advanced Automation Techniques in Adaptive Material Processing*, Chen, X. Q.; Devanathan, R. & Fong, A. M. (Ed.), pp. 55-90, World Scientific, ISBN 981-02-4902-0, Singapore.

Corker, P. I. , Robot Toolbox for MATLAB Reference Guide, 1993-1999

Fu, K. S.; Gonzalez, R. C. & Lee, C. S. G (1987). *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Inc., ISBN 0070226253, New York.

Ge, Q.J. and Ravani, B. (1994). Geometric construction of Bezier motions, *ASME Journal of Mechanical Design*, Vol.116, No.3, pp.749-755

Ge, Q.J. and Kang, D.L. (1995). Motion interpolation with G^2 composite Bezier motions, *Transactions of ASME, Journal of Mechanical Design*, Vol.117 ,No.3, pp.520-525

Gong Z. M., Chen X.Q., and Huang H. (2000). Optimal profile generation in surface finishing, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 1557-1562, San Francisco, pp14-28

Gerald Farin (editor) (1991), NURBS for Curve & Surface Design, Society for Industrial & Applied Mathematics

Grudic, G.Z. and Lawrence, P.D. (1993). Iterative inverse kinematics with manipulator configuration control, *IEEE Trans. on Robotics and Automation*, Vol.9, No.4, pp. 476-483

Huang, H.; Gong, Z. M.; Chen, X. Q., and Zhou, L. B. (2002). Robotic Grinding/Polishing for Turbine Vane Overhaul. *Journal of Materials Processing Technology*, Volume 127 (2002), pp. 140-145.

Hunt, K.H. (1978). Kinematics Geometry of Mechanisms, Clarendon Press, Oxford

Jun, Bong-Huan, Lee, Jihong, and Lee, Pan-Mook (2006), Repetitive Periodic Motion Planning and Directional Drag Optimization of Underwater Articulated Robotic Arms, *International Journal of Control, Automation, and Systems*, vol. 4, no. 1, pp. 42-52, February 2006

Konjovic,Z., Vukobratovic, M. and Surla, D. (1994). Synthesis of the optimal trajectories for robotic manipulators based on their complete dynamic models, *International Journal of Robotics and Automations*, Vol.9, No.1, pp. 36-47

Kyriakopoulos, K.J. and Saridis, G. N. (1994). Minimum jerk for trajectory planning and control, *Robotica*, 12, pp.109-113

Lee, J.H. (1995). A dynamic programming approach to near minimum-time trajectory planning for two robots, *IEEE Trans. on Robotics and Automations*, Vol.11,No.1, pp.160-164,1995

Leu, M.C., and Singh, S.K. (1989). Optimal planning of trajectories robots, *CAD Based Programming for Sensory Robots*, B. Ravani (ed.), Springer-Verlag

Kim, J. Y. (2004), CAD-Based Automated Robot Programming in Adhesive Spray Systems for Shoe Outsoles and Uppers, *Journal of Robotic Systems*,21(11), 625–634 (2004)

Manocha,D. and Canny, J. F. (1994). Efficient inverse kinematics for general 6R manipulators, *IEEE Trans. on Robotics and Automations*, Vol.10, No.5, pp.648-657

Mitsi,S., Bouzakis, K.-D., and Mansour, G. (1995). Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joint limits, *Mech. Mach. Theory*, Vol.30,No.5, pp.653-663

Makino,H., Xie, C.X. and Zhen, S.X. (1982). Spatial mechanisms and robotic mechanisms, *China Mechanical Industry Press*, Beijing

MATLAB 5.3 Reference Guide, The MathWorks Inc., 1998

Osman Gursoy (1992). Some results on closed ruled surfaces and closed space curves, *Mech. Mach. Theory*, Vol.27,No.3, pp:323-330

Patel, R.V. and Lin, Z. (1988). Collision-free trajectory planning for robot manipulators, *Proceedings of 1988 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Vol.2, pp.787-790

Paul Dierckx (1993), Curve and Surface Fitting with Splines, *Oxford University Press*

Pfeiffer,F. and Johanni, R. (1987). A Concept for manipulator trajectory planning, *IEEE Journal of Robotics and Automation*, Vol.RA-3, No.3, pp.115-123

Pachidis, T.P. Tarchanidis, K.N., Lygouras, J.N., Tsalides, P. G. (2005), Robot Path Generation Method for a Welding System Based on Pseudo Stereo Visual Servo Control, EURASIP *Journal on Applied Signal Processing*, 2005:14, 2268–2280

Ridely,P.R., Bokelberg, E.H. and Hunt, K.H. (1992). Spatial motion-II: acceleration and the differential geometry of screws, *Mech. Mach. Theory*,Vol.27,No.1,pp:17-35

Roth, B. (1967). On the screw axes and other special lines associated with spatial displacements of a rigid body, *ASME Journal of Engineering for Industry*, Series B, Vol.89,No.1,pp.102-110

Seereeram, S. and Wen, J.T. (1995). A global approach to path planning for redundant manipulators, *IEEE Trans. on Robotics and Automation*, Vol.11, No.1, pp.152-160

Shin, K.G. and McKay, N.D. (1985). Minimum-time control of robotic manipulators with geometric path constraints, *IEEE Trans. Automat. And Cont*., Vol. AC-30, No.6,pp.531-541

Slotine, J.J.E and Yang, H.S. (1989). Improving the efficiency of time-optimal path-following algorithms, *IEEE Trans. Robotics and Automation*, Vol.5, pp.118-124

Shiller,Z., and Lu, H.-H. (1990). Robust computation of path constrained time optimal motions, 1990 *Proc. IEEE Int. Conf. Robotics and Automation*,pp.144-149

Tabarah, E. , Benhabib, B., and Fenton, R.G. (1994). Optimal motion coordination of two robots-a polynomial parameterization approach to trajectory resolution, *Journal of Robotic Systems*, 11(7), pp.615-629

Thompson, S.E. and Patel, R.V. (1987). Formulation of joint trajectories for industrial robots using B-splines, *IEEE Trans. Industrial Electronics*, Vol,.34, No.2, pp.192-199

Yang, T. and Jin, W.M. (1988). Study on the kinematics performance of robot manipulators using the theory of velocity space, *Proceedings of 1988 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Vol.2, pp.1364-1366

Yoshikawa,T. (1985). Manipulability of robotic mechanisms, *Int. J. of Robotics Research*, Vol.4 , No.2, pp. 3-9

Zeid,I. (1993). CAD/CAM Theory and Practice, *McGraw-Hill International Editions*, Computer Science Series

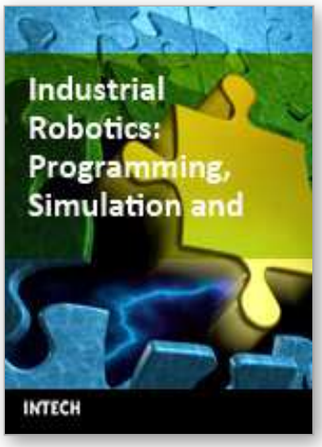
Zha, X.F. (1993). On the harmonious planning and simulation of the trajectories of position and orientation for robot end effector, *Proceedings of the 9th Int. Conf. on Computer Aided Production Engineering*, pp.296-301, Nanjing, China

Zha, X.F., Wang, M., and Choi, A.C.K. (1994). Optimal trajectory planning for robot manipulators, *Proceedings of the 2nd Asian Conference on Robotics and Applications*, International Academics Publishers, pp.569-575, Beijing, China

Zha, X.F., Du, H. (2001). Generation and simulation of robot trajectories in a virtual CAD-based off-line programming environment, *International Journal of Advanced Manufacturing Technology*, Vol.17:610-624

Zha, X.F. (2002). Optimal Pose Trajectory planning for robot manipulators, *Mechanism and Machine Theory*, Vol.37: 1063–1086

Zha, X.F., Chen, X.Q. (2004). Trajectory coordination planning and control for robot manipulators in automated material handling and processing, *International Journal of Advanced Manufacturing Technology*, Vol.23:831-845



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

IntechOpen