Pergamon

Mathl. Comput. Modelling Vol. 21, No. 5, pp. 105–115, 1995

0895-7177/95 $9.50 + 0.00

0895-7177(95)00017-8

Generating a Reference Trajectory with Defined Kinematics for the IRb-6 Manipulator

T. SZKODNY
Institute of Automation, Silesian Technical University
ul. Akademicka 16, 44-100 Gliwice, Poland

(Received January 1994; accepted April 1994)

Abstract—The fundamental problem in industrial robots control concerns algorithms generating reference trajectories.

References [1–4] suggest generating algorithms of a reference trajectory, which are based on an arbitrary discretization of the manipulator's internal coordinates. Each point of discretization in the external space approximating a reference trajectory corresponds to known discretized internal coordinates of the manipulator.

In [5–7], iteration methods of determining the internal coordinates corresponding to external coordinates of the reference trajectory point have been suggested. In this method of internal coordinates, determining the point of the reference trajectory is being approached in successive steps of an iterative computation. In [5], a modified iterative method of generation of a straight segment reference trajectory has been presented.

Analytic formulae, which are the solution of an inverse problem of manipulator kinematics, enable design of trajectory generating algorithms which compute, in one step only, the internal coordinates of points lying exactly on the reference trajectory, with the accuracy resulting from the computer register length.

In this paper, the author has presented an original **PLAN2** computer algorithm generating reference trajectories of motion for a task. The kinematics of those trajectories is defined at selected points through which a task is to be passed, the distances between them being optional. The algorithm is based on formulae which are analytic solutions to an inverse problem for an IRb-6 manipulator kinematics.

Keywords—Planning of reference trajectory of manipulators motion.

NOMENCLATURE

ΔT	discretization time along reference trajectory segment
Φ_{ref}, Θ_{ref}, Ψ_{ref}	reference external coordinates of orientation
k_4, k_5	transmission ratios in the fourth and fifth degree of freedom of an IRb-6 manipulator
l_6, λ_6	kinematic parameters describing the task
R_2	longest effective turning radius of the task
Θ_i	natural coordinate of i^{th} link
Θ_{ai}	natural coordinate of i^{th} actuator
T	reference time of the consecutive main *fulcrums*

1. INTRODUCTION

To simplify the description, the abbreviation IRM will be used for 'industrial robot manipulator.' The IRM external space is described by its external coordinates of position x_{ref}, y_{ref}, z_{ref}, and orientation Φ_{ref}, Θ_{ref}, Ψ_{ref} (Euler angles). The coordinates describe the task in relation to

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX a chosen reference coordinate system, irrespective of the IRM kinematic structure. The internal space of the IRM is described by its internal coordinates, which are the natural coordinates of links and actuators [8]. Reference values of actuator natural coordinates result either from education of a robot [9], or from trajectory computing tier, cooperating with a camera or another sensor of the scene.

One of the major problems of industrial robot control is designing of reference trajectory generating algorithms. In [1–4], reference trajectory generating algorithms, based on arbitrary discretization of the IRM internal coordinates have been proposed. A discretized description of the IRM external space is obtained from a discretized description of the IRM internal space. Reference trajectory in the external space is approximated by using the IRM external space description discretized in this way. Each discretization point in the external space approximating the reference trajectory corresponds to known discretized IRM internal coordinates. Some disadvantages of those algorithms are: a large memory is required; big sets, being a discretized internal space description, must be searched; and it is not possible to reduce the approximation error which results from arbitrary discretization of the IRM internal space.

In [5–7], iterative methods for determining internal coordinates corresponding to external coordinates of a reference trajectory point are proposed. In this method of determining internal coordinates, a reference trajectory point is being approached in consecutive steps of an iterative computation. A step of internal coordinates discretization, in a set of consecutive steps of the iterative computation, depends on the error of external coordinates in the previous step of iteration. Iterative methods do not require large memories because computing is carried out only for the reference trajectory approximating points. Although the error in the reference trajectory approximation may be reduced in these methods, the number of iterative computing steps increases. In [5], a modified iterative method of a reference trajectory rectilinear segment generation has been presented. The modification consists of arbitrarily accepting an error distribution in the external space, which reduces a number of iterative computing steps. However, such an error distribution is only useful for short segments. It is not possible to determine their length limits, which would guarantee the errors to be bound for an arbitrary IRM configuration. A disadvantage of the iterative methods is the necessity of multiple iterative computations. An advantage of both the methods, based upon arbitrary discretization of the internal spaces and iterative methods, is the simplicity of computing when only equations of IRM direct kinematics are used. The advantage may, however, turn out to be a trap for those computer programmers who did not consider the IRM kinematic singularities [8].

The analytic formulae, being solutions to an inverse problem of the IRM kinematics, afford possibilities for design of trajectory generation algorithms which compute the internal coordinates of the points situated precisely along the reference trajectory in one step and with the accuracy resulting from the computer registers length. The formulae force the programmers to provide alternative solutions for singular states of the IRM. The analytic formulae, solutions of an inverse problem of the six degrees of freedom IRM kinematics, are presented in [10–12]. The same formulae for the five degrees of freedom IRM were presented in [13,14]. However, a constraint equation for a working link [8] has not been mentioned in [14]. This implies that the link can realize reference trajectories with six degrees of freedom, which is impossible. The reviewed references [1–7,14] present models of kinematics which do not allow the design of accurate and simultaneously fast generation algorithms of the reference trajectory with a defined kinematics for the IRM less than six degrees of freedom.

This author has worked out the kinematics models [8] of the IRM with less than six degrees of freedom. In [8], the models of direct and inverse kinematics with constraint equations for the IRMs PR-02, IRb-6, and IRb-60 have been presented. The models have been presented in continuous and differential form; kinematic singularities being taken into account.

In Section 2 of this paper, an original generation algorithm of reference trajectories for the IRb-6 manipulator has been described. The algorithm determines internal coordinates of the points situated precisely along the reference trajectory in a single step. It also affords possibilities for defining reference kinematics in the form of external coordinates of points that are freely distant from each other. Section 3 presents a description of a numerical example resulting from running the generation algorithm described in Section 2. The algorithm was written in FORTRAN 77 and run on a μVAX-3800 computer. Section 4 contains conclusions.

2. REFERENCE TRAJECTORY GENERATION

The design of algorithms generating natural coordinates of actuators is based on formulae which are solutions to an inverse problem of IRM kinematics. The coordinates correspond to the manipulation object reference trajectory described in the robot internal space. The algorithms generating natural coordinates of actuators form a tier of reference trajectory computations, which is an element of the functional structure of an intelligent robot control system [15]. These algorithms may be the indispensable programming tools with which to interconnect a vision tier with a drive control tier [9].

This author has designed the PLAN2 computer algorithm generating trajectories for an IRb-6 manipulator task. The reference external coordinates of the points through which a generated trajectory will pass, will be called main *fulcrums* (or coordinates of the main via points). Generation requires a preliminary description of the trajectory, in the form of values of external coordinates at least two main *fulcrums*, optionally distant from each other. In case of a kinematic singularity occurrence, the PLAN2 algorithm announces the state, gives the values of acceptable natural coordinates for the links, and asks the user which of the given values are to be accepted. The PLAN2 algorithm generates additional *fulcrums* for, either defined or nondefined, kinematics between the successive main *fulcrums*.

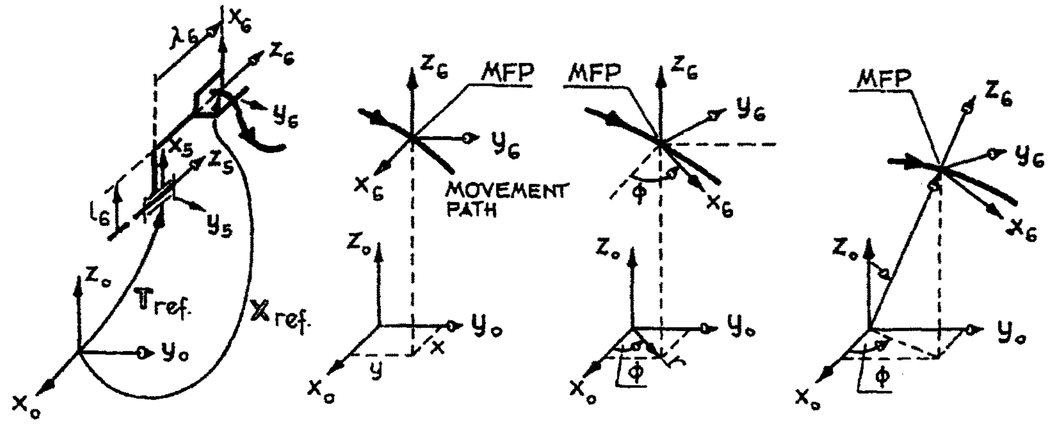
The algorithm for defined kinematics was described in [16], and the one for nondefined kinematics in [17]. The present paper will describe the PLAN2 algorithm for defined kinematics of a reference trajectory. The algorithm comprises four basic segments; the computation, from there, is transmitted to 21 ancillary segments. The basic segments are:

(a) a master segment,
(b) the ROZ1 segment,
(c) the ROZ2 segment, and
(d) the ROZ3 segment.

To simplify the description, the following abbreviations will be used: MRF—main *fulcrum* and AFP—additional *fulcrum*.

After starting, the PLAN2 algorithm asks about the l_6 and λ_6 parameters describing the task (see Figure 1). Then, it asks about the number (MFP$\leq$ 50) of external coordinates and the reference time T for the consecutive MFP, and whether the consecutive MFP orientation is defined. If the orientation is defined, the algorithm asks if it is to be computed. If so, the next question is in what coordinate system it is to be computed; Cartesian, cylindrical or spherical. After the required coordinate system has been set, Euler angles are determined, describing the orientation of a given MFP. The angles are shown in Figure 1. If the defined orientation is not to be computed, a question about MFP Euler angles appears. For a nondefined MFP orientation, the Ψ_{ref} angle is being set arbitrarily. The Φ_{ref} and Θ_{ref} angles are computed from the x_{ref}, y_{ref}, z_{ref} external coordinates (describing the present MFP position).

For the so determined x_{ref}, y_{ref}, z_{ref}, Φ_{ref}, Θ_{ref}, Ψ_{ref} external coordinates, describing the consecutive MFP, the algorithm determines the $\mathbf{T}_{5\text{ref}}$ matrix [8], checks whether the constraint equation is satisfied [8], and derives Θ_1'–Θ_5' natural coordinates from the formulae, which are an analytic solution to an inverse problem of IRb-6 IRM kinematics. Then, the algorithm asks about a coordinate system describing the shape of a trajectory segment between the consecutive MFP. For a rectilinear segment, a Cartesian coordinate system should be chosen; for a curvilinear segment, either a cylindrical or spherical system should be chosen. The next question the algorithm asks is



(a) Cartesian (b) cylindrical (c) spherical

Figure 1. Euler angles in the coordinate system.

about the discretization time ΔT along the present reference trajectory segment. After the shapes and discretization times along each segment between the consecutive MFP have been defined, the algorithm asks about the admissible DP error of position and the DF error of orientation of the task. If an optional task orientation was set earlier, the algorithm arbitrarily accepts DF$= 360°$.

The next question the algorithm asks concerns the kind of trajectory generation, which may be set as: free, rough or accurate. When either rough or accurate generation has been set, a question about an admissible DFW error of orientation is being asked. This parameter is necessary for the IRM internal space discretization. For a free generation, the algorithm determines the AFP external coordinates coming from the pre-set DP, DF, and ΔT parameters, which ensure the declared shape of the trajectory segment in the external space. So determined AFPs are illustrated in Figure 2. If the length of a trajectory segment between the $(i-1)^{\text{th}}$ and the i^{th} MFP is denoted by l, then Δl_j, the length of a trajectory segment between the consecutive AFPs may be expressed as follows:

$$\Delta l_j = \min\left[\frac{2\,\mathrm{DF}}{|\Phi_{i,\mathrm{ref}} - \Phi_{i-1,\mathrm{ref}}|}\,l,\ \frac{2\,\mathrm{DF}}{|\Theta_{i,\mathrm{ref}} - \Theta_{i-1,\mathrm{ref}}|}\,l,\right.$$
$$\left.\frac{2\,\mathrm{DF}}{|\Psi_{i,\mathrm{ref}} - \Psi_{i-1,\mathrm{ref}}|}\,l,\ \frac{2\,\Delta T}{|T_i - T_{i-1}|}\,l,\ \Delta l\left(d \le \frac{1}{2}\,\mathrm{DP}\right)\right], \tag{1}$$

where $\Phi_{i,\mathrm{ref}}$, $\Phi_{i-1,\mathrm{ref}}$, $\Theta_{i,\mathrm{ref}}$, $\Theta_{i-1,\mathrm{ref}}$, $\Psi_{i,\mathrm{ref}}$, $\Psi_{i-1,\mathrm{ref}}$ are the Euler angles of the i^{th} and $(i-1)^{\text{th}}$ MFP; Δl and d, parameters illustrated in Figure 2; $\Delta l(d)$ is the length of a trajectory segment between consecutive AFPs dependent on the d distance, illustrated in Figure 2; d is the distance of the Δl trajectory segment center from the straight line connecting the $(j-1)^{\text{th}}$ and j^{th} AFP.

It follows from Figure 2 that the smaller d, the smaller Δl. With such a length of trajectory segments, it may be assumed that none of the trajectory points described by the coordinates within the ranges corresponding to neighboring AFPs will go beyond the "tube" of diameter DP, formed by the successive cylinders illustrated in Figure 2. To determine the AFP orientation, the algorithm assumes a linear change of two Euler angles along the trajectory. The third angle results from the constraint equation [8]. The AFP time is linearly dependent on the length l. When the consecutive MFPs differ in orientation only, the function of length l is taken over by

$$\max\left(|\Phi_{i,\mathrm{ref}} - \Phi_{i-1,\mathrm{ref}}|, |\Theta_{i,\mathrm{ref}} - \Theta_{i-1,\mathrm{ref}}|, |\Psi_{i,\mathrm{ref}} - \Psi_{i-1,\mathrm{ref}}|\right).$$

Then, d is the distance between the center of the trajectory segment connecting consecutive MFPs, and the center of a sphere of diameter DP, coinciding with MFP. Formula (1) is still

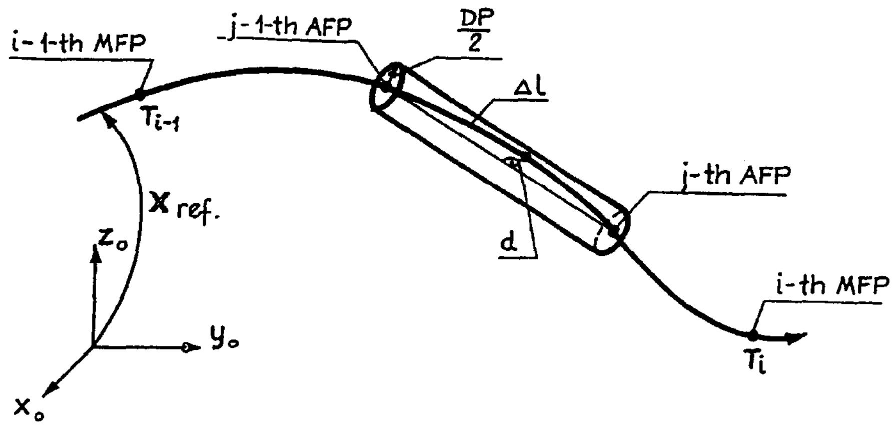


Figure 2. Trajectory segment between the $(i-1)^{\text{th}}$ and i^{th} MFPs. Δl: length of a trajectory segment between the $(j-1)^{\text{th}}$ and j^{th} AFPs; d: distance between the center of a trajectory segment of length Δl and a straight line connecting the $(j-1)^{\text{th}}$ and j^{th} AFPs; T_{i-1} and T_i: times corresponding to the above illustrated MFP.

useful. After having computed the AFP external coordinates, the algorithm checks whether this AFP belongs to the working space. In case that the checked AFP goes beyond the working space, its position, orientation, and time will be corrected. Then, the natural coordinates of the Θ_1'–Θ_5' links and Θ_{a1}–Θ_{a5} actuators corresponding to the AFP are being computed.

For moderate demands regarding trajectory generation, the values of the DP, DF, and ΔT parameters are relatively high. Those values correspond to relatively long trajectory segments between neighboring AFPs. Then, with the trajectory generation set free, the assumption that all the points are situated within the "tube" mentioned before may be wrong. It follows from the fact that the trajectory is realized within the internal coordinates. To prevent the trajectory from going beyond the "tube" at relatively high DP, DF, and ΔT values, the user can set a trajectory rough generation, as well as a DFW parameter. In that case, the algorithm determines two groups of *fulcrums*: additional external *fulcrums* and additional internal *fulcrums*. The additional external *fulcrums* are determined in the same way as in the case of trajectory free generation, and will be denoted as before, using the abbreviation AFP. Additional internal *fulcrums* result from the division of actuator natural coordinates (within ranges corresponding to consecutive AFPs), and will be denoted using the abbreviation AIFP. Figure 3 shows the way in which the AIFPs between the $(j-1)^{\text{th}}$ and j^{th} AFP are computed. Each range of actuator natural coordinates corresponding to consecutive AFPs is divided into $N+1$ parts, where N is described by formula (2). It is assumed in the formula that the minimal angular ratio for orientation errors is equal to $|k_4\,(1-k_5)|$ ($k_4=-128$, $k_5=19/32$, see [8,18]):

$$
\begin{aligned}
N &= E\left(\frac{2\,\mathrm{XX}}{|k_4\,(1-k_5)\,\mathrm{DFW}|}\right), \\
\mathrm{XX} &= \max\left(|\Theta_{s1k}-\Theta_{s1p}|,\dots,|\Theta_{s5k}-\Theta_{s5p}|\right),
\end{aligned}
\tag{2}
$$

where E is the total part of an argument; k_4 and k_5 are transmission ratios in the fourth and fifth degree of freedom of the IRb-6 manipulator. After the actuator natural coordinates of the first AIFP have been determined, the algorithm determines its external coordinates along the trajectory segment between the $(j-1)^{\text{th}}$ and j^{th} AFPs, and checks if it is inside the cylinder of diameter DP (illustrated in Figure 3). If this AIFP is outside the cylinder, then the number N is increased by 2 and the external coordinates of the first AIFP are redetermined. The number N is increased by 2 until the AIFP finds itself within the cylinder. For such modified number N, the algorithm determines the external coordinates of the next AIFP. If the neighboring AFPs differ in orientation only, then the function of the cylinder (illustrated in Figure 3) is taken over by a sphere of diameter DP and the center coinciding with the AFP. The algorithm checks the position

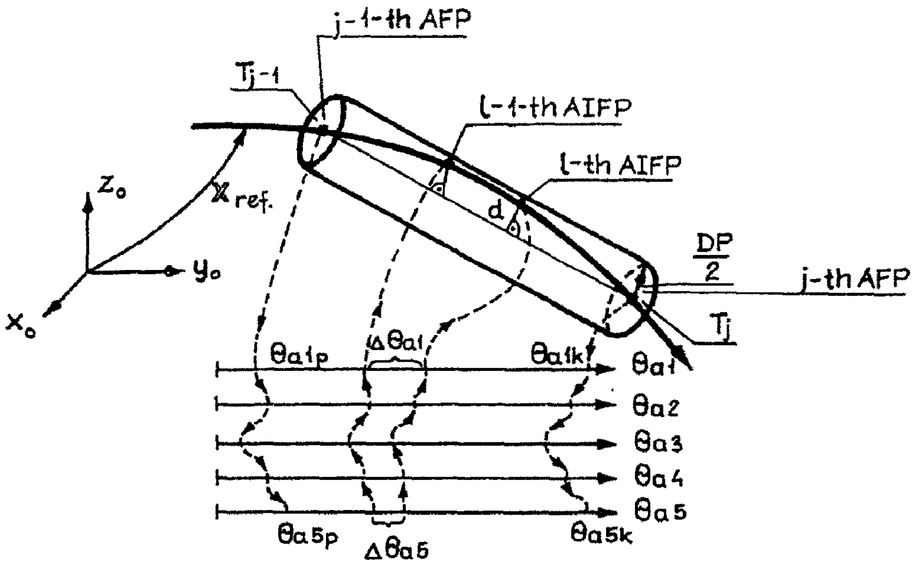


Figure 3. Trajectory segment between the $(j-1)^{\text{th}}$ and j^{th} AFPs. d: distance between the AIFP and a straight line connecting neighboring AFPs; Θ_{a1p}–Θ_{a5p}: actuator natural coordinates corresponding to the $(j-1)^{\text{th}}$ AFP; Θ_{a1k}–Θ_{a5k}: actuator natural coordinates corresponding to the j^{th} AFP; T_{j-1}, T_j: times corresponding of the $(j-1)^{\text{th}}$ and j^{th} AFPs.

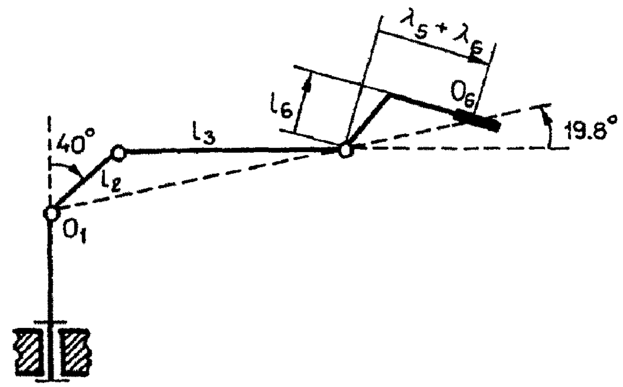


Figure 4. Configuration of the IRb-6 manipulator, with which the task effective turning radius $R_2 = O_1O_6$ is the largest.

of every AIFP in relation to the working space. If a checked AIFP goes beyond the working space, the correction of its position, orientation and actuator natural coordinates follows. To compute the AIFP time, the algorithm assumes its linear dependence on the natural coordinates of the actuators.

In rough trajectory generation, while computing the actuator natural coordinates, the algorithm considers only parameter DFW (see formula (2)). After accurate trajectory generation has been set, each range of the actuator natural coordinates (illustrated in Figure 3) is divided as in rough generation. The only difference is the way of computing the initial value of N divisions. In accurate trajectory generation, this number results from DP and DFW parameters in the following way:

$$N = E\left(\frac{\mathrm{XX}}{\mathrm{XY}}\right), \qquad \mathrm{XY} = \min\left[\frac{1}{2}|k_4\,(1-k_5)\,\mathrm{DFW}\,|, \frac{\mathrm{DP}}{2\,R_2}\right],$$
$$R_2 = \left[(l_2\,\cos 40^\circ)^2 + (l_2 \sin 40^\circ + l_3)^2\right]^{1/2} + \left[(\lambda_5+\lambda_6)^2 + l_6^2\right]^{1/2}, \tag{3}$$

where E is the total part of an argument; R_2 is the longest effective turning radius of a task shifting by DP/2 in the external space. This is an arc length the task describes for $d\Theta_1' = d\Theta_3' = d\Theta_4' =$

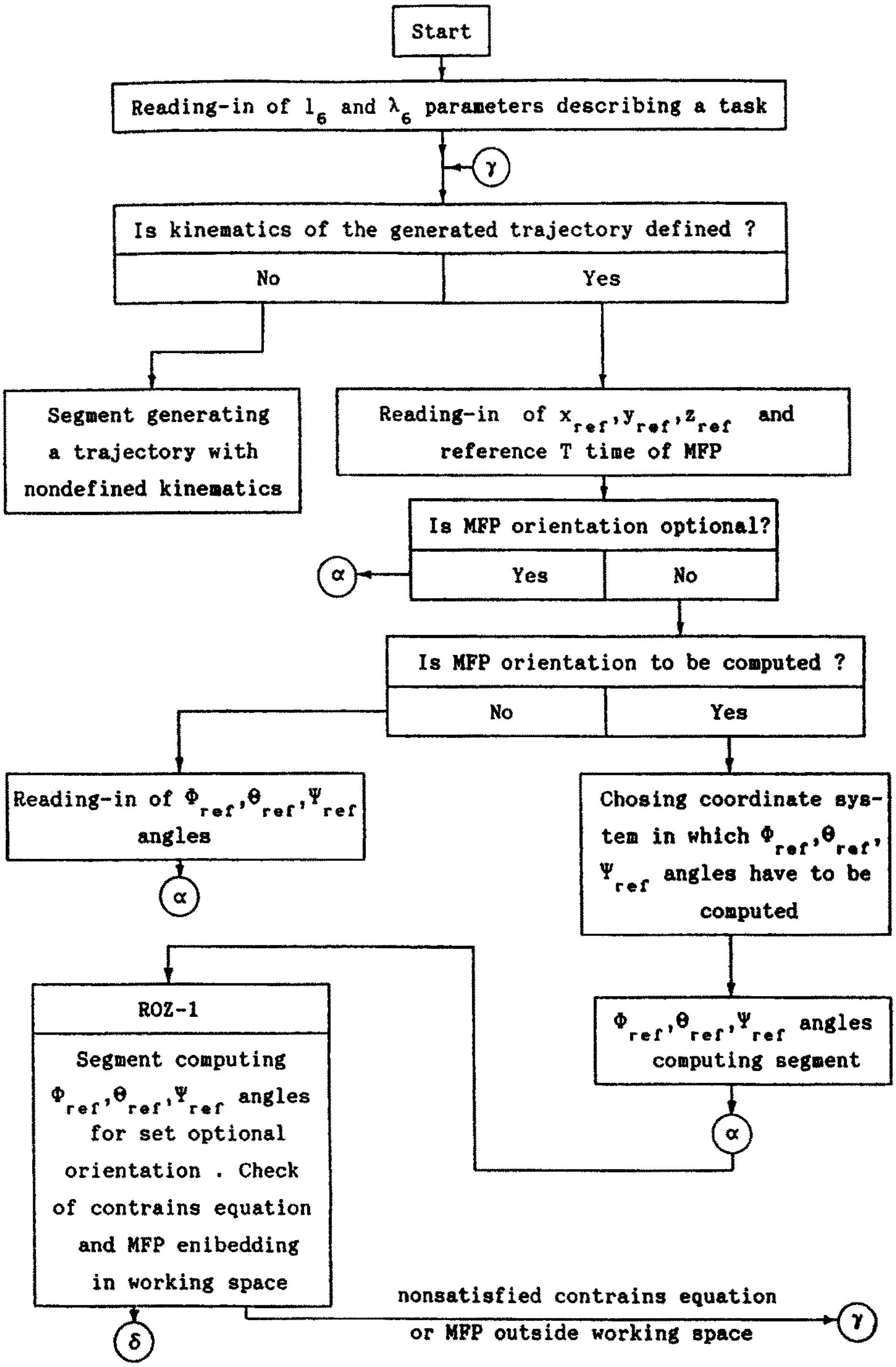


Figure 5. (Part 1) PLAN2 algorithm block diagram.

$d\Theta_5' = 0$ and $d\Theta_2' \simeq \mathrm{DP}/(2R_2)$, with the manipulator configuration as in Figure 4. Further corrections of N as well as the AIFP generation is continued identically as for the previously described rough trajectory generation.

The final result of the algorithm is to generate sets describing all the IRb-6 manipulator internal and external coordinates.

Figure 5 illustrates a block diagram of the PLAN2 algorithm for the defined kinematics of the task reference trajectory.

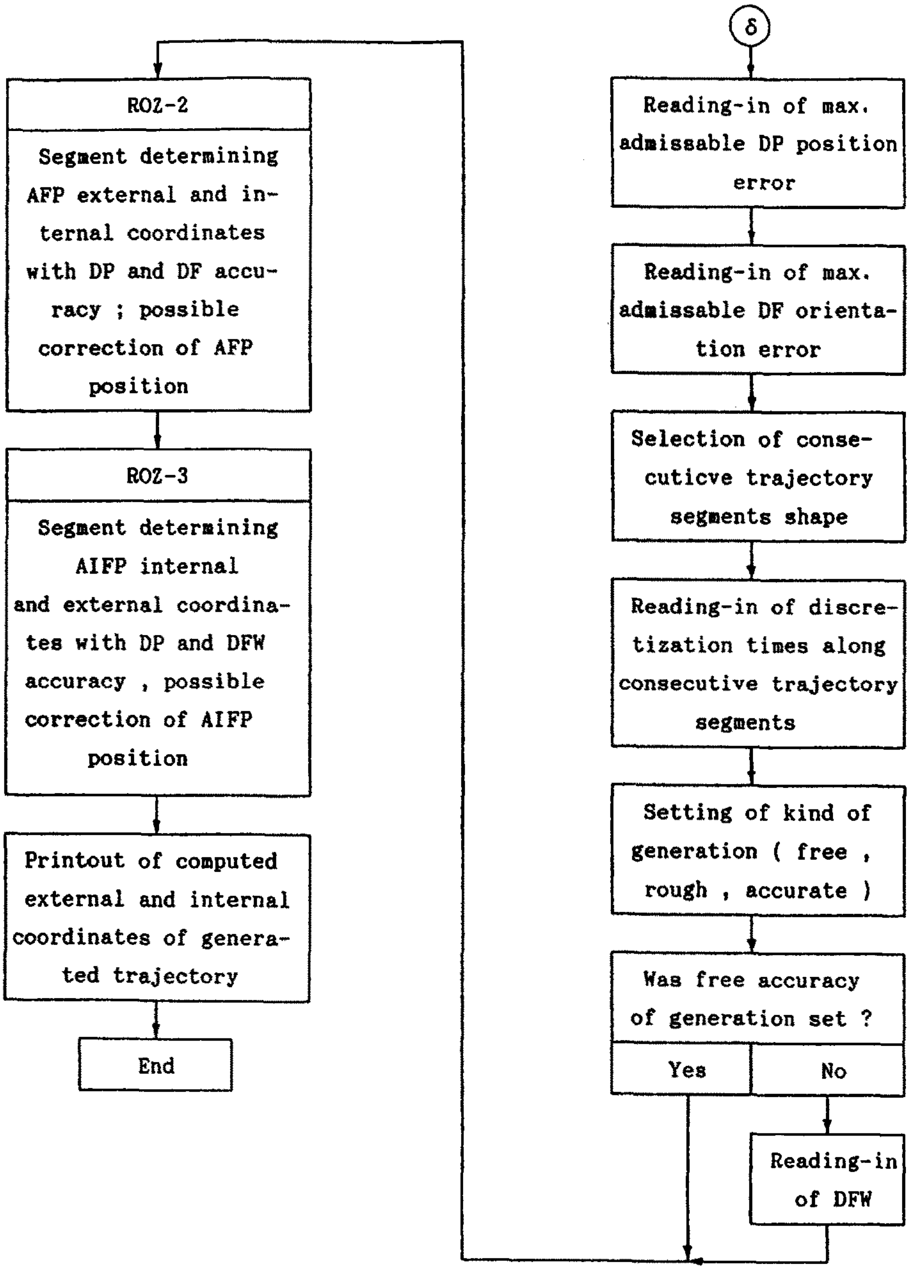


Figure 5. (Part 2) PLAN2 algorithm block diagram.

3. EXAMPLE

Figure 6 illustrates an exemplary rectilinear task reference trajectory, confined between the initial point P and the final point K. Coordinates of these points are as follows: $x_p = -0.60\,\mathrm{m}$, $y_p = 0.60\,\mathrm{m}$, $z_p = 1.0\,\mathrm{m}$, $\Phi_p = 135°$, $\Theta_p = 359°$, $x_K = -0.65\,\mathrm{m}$, $y_K = 0.60\,\mathrm{m}$, $z_K = 1.0\,\mathrm{m}$, $\Phi_K = 137.29$, $\Theta_K = 1°$, $\Psi_K = 180°$. The following PLAN2 algorithm input parameters will be assumed to define the kinematics of the trajectory as in Figure 6: task parameters, $l_6 = 0$ and $\lambda = 0.16\,\mathrm{m}$; MFP number $= 2$ (points P and K); MFP external coordinates of points P and K; the first MFP time, $T_p = 0\,\mathrm{sec}$, the second MFP time, $T_K = 1.0\,\mathrm{sec}$; trajectory: shape-straight line; kind of generation: rough; discretization time, $\Delta T = 0.004\,\mathrm{sec}$; DP$=0.0002\,\mathrm{m}$; DF$=60°$, DFW$=2°$. The generation has resulted in 1283 AFPs illustrated in Figures 7 and 8.

The y_{ref} and z_{ref} coordinates generated by means of the PLAN2 algorithm have an error less than $10^{-4}\,\mathrm{cm}$. It follows from Figure 7b that the angle Ψ_{ref} changes abruptly for $t \simeq 0.5\,\mathrm{sec}$. As

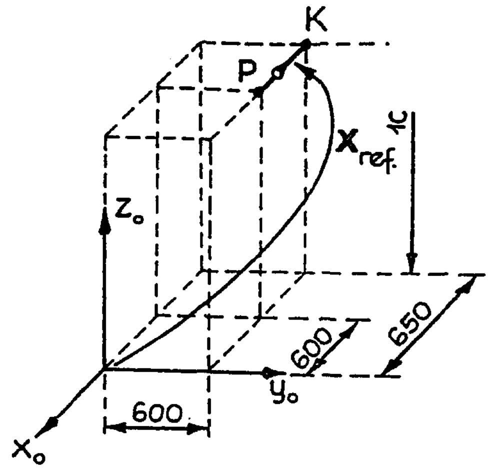


Figure 6. $\mathbf{X}_{ref}$ reference trajectory of the task.

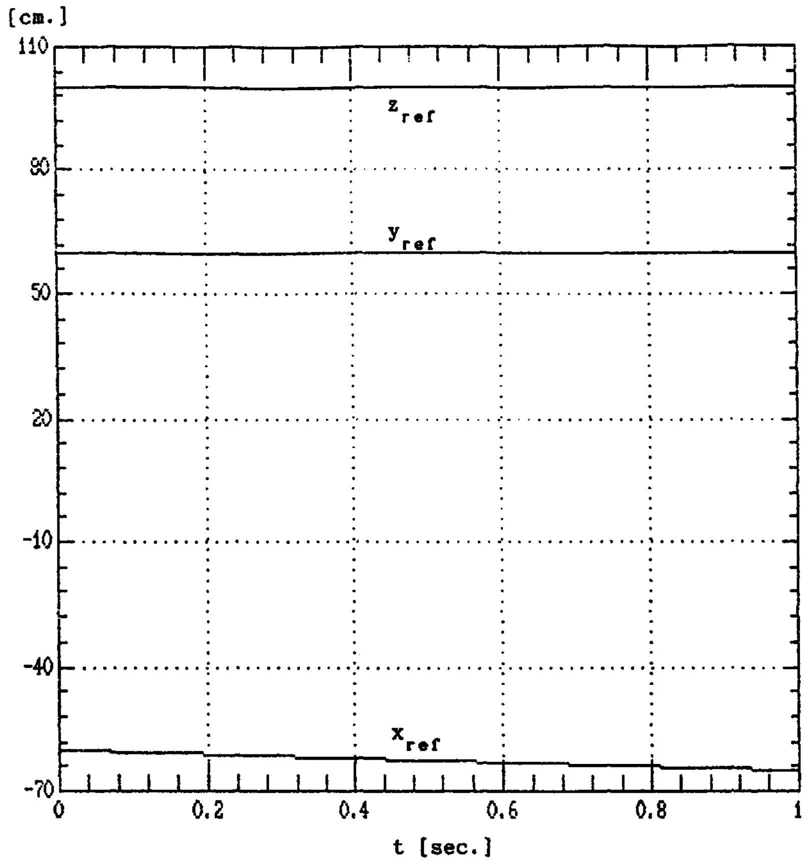


Figure 7a. Task position reference coordinates $x_{ref}(t)$, $y_{ref}(t)$, $z_{ref}(t)$.

it is illustrated in Figure 8, the jump is caused by an abrupt change of the fifth actuator natural coordinate at that time. An abrupt change of Θ_{a5ref} occurs between the AFPs numbered 377 and 900, from $-20,357.896°$ to $6,805.735°$, causing the change of the Θ'_{5ref} natural coordinate from the minimal to the maximal boundary value [8,18]. This jump of Θ'_5 equals 360°, because that is the difference between Θ'_{5ref} minimal and maximal boundary values.

The example shows that a linear change of the angle Ψ_{ref} between the reference trajectory between points P and K is not possible, as in Figure 6. This is caused by singularities of the IRb-6 IRM kinematic structure.

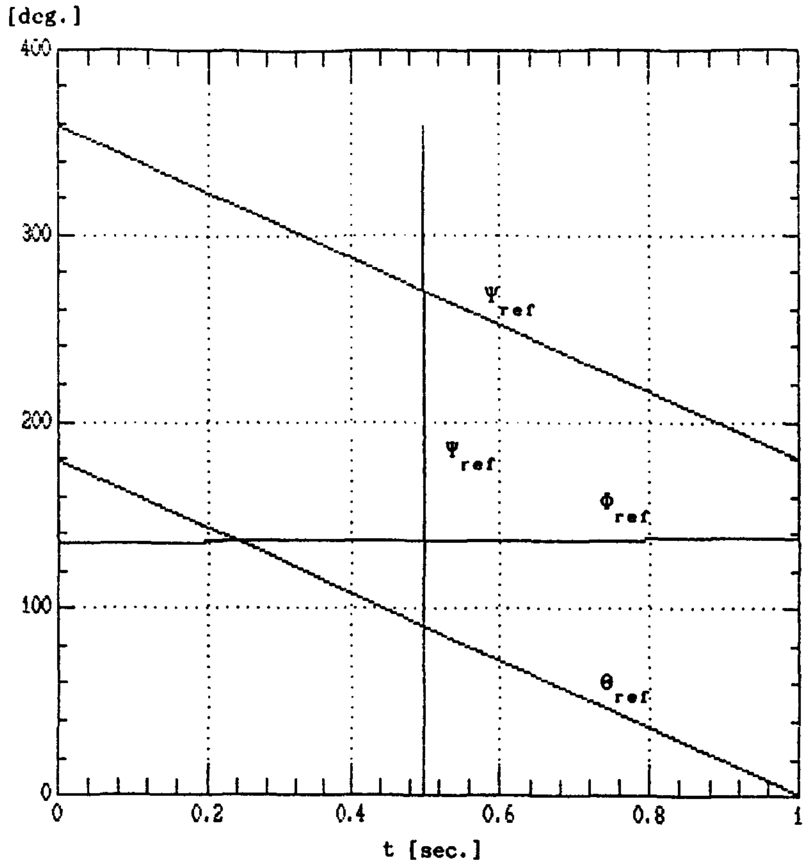


Figure 7b. Task orientation reference coordinates $\Phi_{\mathrm{ref}}(t)$, Θ_{ref}, $\Psi_{\mathrm{ref}}(t)$.

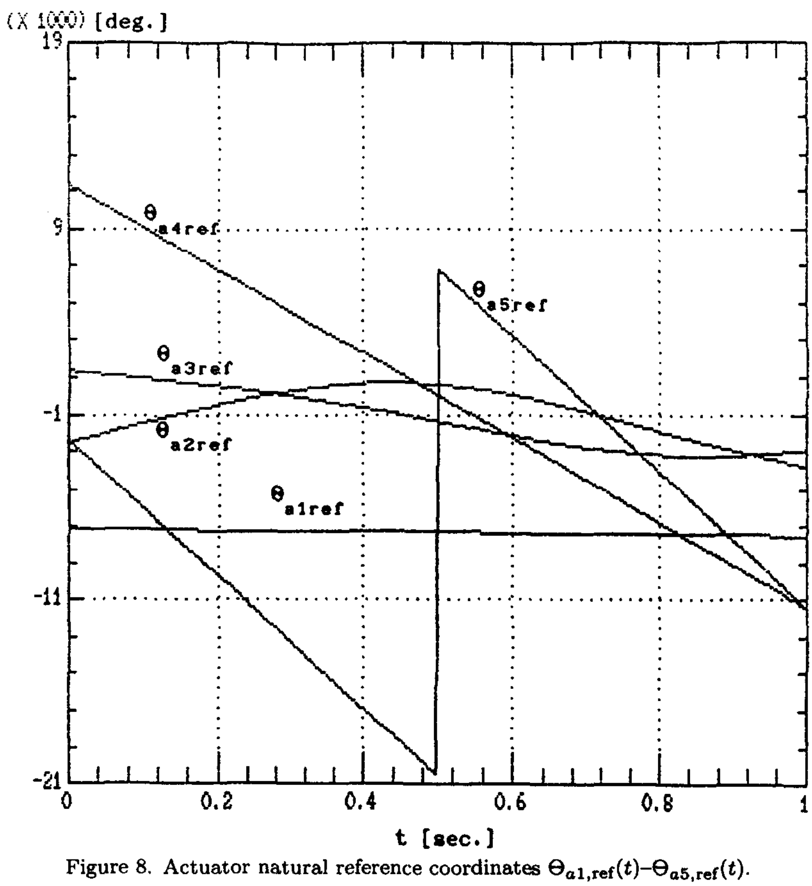


Figure 8. Actuator natural reference coordinates $\Theta_{a1,\mathrm{ref}}(t)$–$\Theta_{a5,\mathrm{ref}}(t)$.

4. CONCLUSIONS

The algorithm presented in this paper may be used as a computer tool for:

(a) planning the IRb-6 IRM task reference trajectory, with a selected motion path shape and either a defined or optional orientation;
(b) analysis of the kinematic possibilities of realization of trajectories set by the IRb-6 IRM, being a result of a designed robots work station;
(c) generation of actuator natural coordinates corresponding to task reference trajectories, being also reference values of the IRb-6 IRM servo-control; and
(d) design of a trajectory computing tier for IRb-6 and IRp-6 robots with identical manipulators.

This algorithm is susceptible to modification: any obstacle can be avoided through modification of an ancillary segment defining the working space.

REFERENCES

1. W. Gerke, Collision-free and shortest path for industrial robots found by dynamic programing, *Robotosysteme* **1**, 43–52 (1985).
2. L. Gouzenes, Strategies for solving collision free trajectories, Problem for mobile and manipulator robots, *Intern. J. Robotics Res.* **3** (4), 51–65 (1984).
3. W. Jacak, Strategies of searching for collision-free manipulators motions: Automata theory approach, *Robotica* **7** (2), 129–138 (1989).
4. T. Lozano-Perez, Spatial planning: Configuration space approach, *IEEE Trans. on Computers* **32** (2), 108–120 (1983).
5. I. Dulęba and B. Łysakowska, R.M. Taylor's modified method for parametrization of straight IRb-6 robot movement (in Polish), *Scientific Papers, Institute of Technical Cybernetics, Technical University of Wrocław*, Volume 2, *III National Conference on Robotics*, pp. 61–66, (1990).
6. W. Jacak and B. Łysakowska, Discrete methods of a robot's kinematics modelling (in Polish), *Scientific Papers, Institute of Technical Cybernetics, Technical University of Wrocław*, Volume 2, *III National Conference on Robotics*, pp. 176–183, (1990).
7. R.M. Taylor, *Planning and Execution of Straight-Line Manipulator Trajectories, Robotmotion: Planning and Control*, MIT Press, Cambridge, MA, (1983).
8. T. Szkodny, Industrial robot manipulators, Mathematical models (in Polish), *Silesian Techn. Univ. Publ. Comp.*, No. 1530, Gliwice, (1990).
9. A. Niederliński, *Industrial Robots* (in Polish), School and Pedagogical Camp. Publ., Warsaw, (1983).
10. J. Knapczyk and I. Kisiel, Vector method of determining of the 6 rotational kinematic pairs manipulator links movement (Inverse problem of kinematics) (in Polish), *Silesian Techn. Univ. Publ. s.Mechanics*, No. 86, Gliwice, (1987).
11. R.P. Paul, *Robot Manipulators: Mathematic, Programming and Control*, MIT Press, Cambridge, MA, (1983).
12. T. Szkodny, Kinematic decomposition of movement of IRb-60 robot of the effector (in Polish), *Scientific Papers, Institute of Technical Cybernetics, Technical University of Wrocław, III National Conference on Robotics*, Volume 2, pp. 247–254, (1990).
13. J.J. Craig, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, New York, (1989).
14. J. Knapczyk and A. Stępniewski, Kinematic and dynamic analysis of the 5 rotational kinematic pairs manipulator using matrix method for the reference trajectory (in Polish), *Silesian Techn. Univ. Publ. Comp., s.Mechanics*, No. 86, Gliwice, (1985).
15. L.M. Camarinha-Matos, Plan generation in robotics, *IEEE J. Robotics and Automation 3* **RA-3** (1987).
16. T. Szynawa, Computer planning of minitimes trajectories with reference kinematics for IRb-6 manipulator (in Polish), Dypl. Work, Automation Institute, Silesian Techn. Univ., Gliwice, (1991).
17. T. Baraniec, Planning of minitime trajectories with nondefined kinematics for IRb-6 robot (in Polish), Dypl. Work, Automation Institute, Silesian Techn. Univ., Gliwice, (1991).
18. T. Szkodny, Dynamics of industrial robots manipulators, (unpublished work).