

Piotr GIERLAK¹

SINGULARITY ROBUST TRAJECTORY GENERATOR FOR ROBOTIC MANIPULATOR BASED ON GENETIC ALGORITHM WITH DYNAMIC ENCODING OF SOLUTIONS

In this paper a singularity robust trajectory generator for robotic manipulators is presented. The generator contains the procedure of solving the inverse kinematics problem. This issue is defined as an optimization problem, where a genetic algorithm is used for optimizing the fitness function. In order to avoid singularity problem, the generator is based on the direct kinematics problem. The trajectory generator allows to obtain generalized coordinates, velocities and accelerations. Simulation results show that the procedure generates a trajectory of manipulator even in kinematics singularities.

Keywords: inverse kinematics, genetic algorithm, singularity, robotic manipulator, trajectory generator

1. Introduction

One of the basic problems of robotics is generating a desired trajectory of manipulators' motion. The desired trajectory generator contains a procedure of solving the inverse kinematics problem (IKP), which enables to calculate generalized coordinates, velocities and accelerations in the manipulator joint space. All the variables are necessary to control manipulators because they are the desired trajectory of the control system (Fig. 1). However, authors of many publications which deal with IKP often consider only the motion from an initial point to a final point [1, 2] and do not deal with generation of generalized velocity and acceleration of motion. Some algorithms allow to obtain a solution to the singular IKP [3] but they are very complex and applicable to special structures of manipulators only. Moreover, the authors are concerned with the problem of generalized coordinates but they do not consider the problem of generalized velocities and accelerations. In the articles [4, 5] the authors presented methods based on artificial intelligence which allow to obtain generalized coordinates and velocities even in singularities. However, a solution to the problem of generalized accelerations has not been found due to too large computational complexity of the method.

¹ Corresponding author/autor do korespondencji: Piotr Gierlak, Politechnika Rzeszowska, 12 Powstańców Warszawy Ave., 35-959 Rzeszów, tel.: (17) 8651854, e-mail: pgierlak@prz.edu.pl

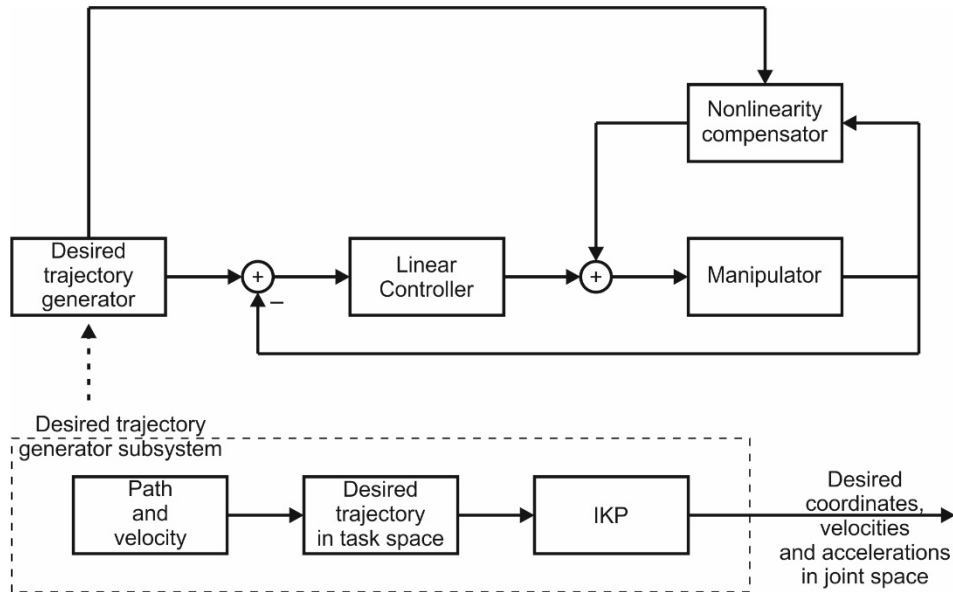


Fig. 1. Schema of manipulator control system with the trajectory generator

Kinematics' equation of manipulator may be written in a general form as follows:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \quad (1)$$

where: $\mathbf{x} \in \mathbf{R}^n$ – a vector of position and orientation of manipulator's end-effector in a frame of reference, which is connected with task space,

$\mathbf{q} \in \mathbf{R}^m$ – a vector of generalized coordinates,

$\mathbf{f} \in \mathbf{R}^n$ – a vector function, usually non-linear, the structure and parameters of which are known.

The IKP may be symbolically written as:

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}) \quad (2)$$

Because $\mathbf{f}(\mathbf{q})$ is non-linear many \mathbf{q} give the same \mathbf{x} , so the equation (2) may have more than one solution. Moreover, the inverse of the function \mathbf{f} may not exist. Typical disadvantages in the IKP solution are presented graphically in Fig. 2.

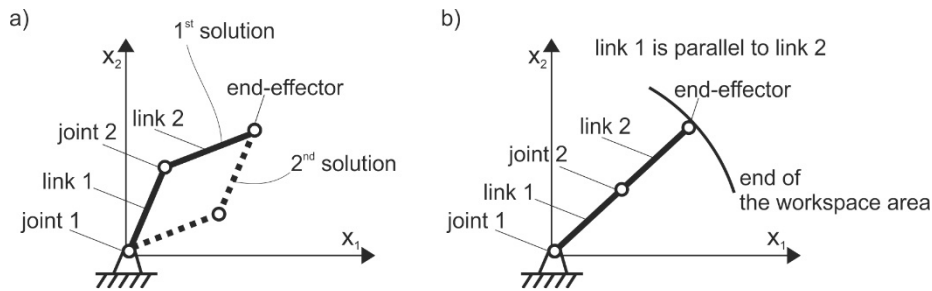


Fig. 2. Typical disadvantages in the IKP solution for a two-link manipulator: a) ambiguous solution, b) singular configuration

There is a classical method used to solve IKP, in which the inverse of a Jacobian matrix is used. Jacobian matrix is defined as

$$J(\mathbf{q}) = \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} \quad (3)$$

and depending on the dimensions of the joint space and task space it may be a square or rectangular matrix. The IKP solution is based on the numerical solution of the equation

$$\dot{\mathbf{q}} = J^{-1} \dot{\mathbf{x}} \quad (4)$$

While solving IKP by inversion of the Jacobian matrix a problem of singularity may occur. This is strictly a mathematical problem which occurs in configuration \mathbf{q} , in which an order of the Jacobian decreases. The most formal definition of singular configuration is as follows [3]: a configuration \mathbf{q} in which a rank of the geometric Jacobian is less than the dimension of the task space T , which can be written as

$$\text{rank}(\mathbf{J}(\mathbf{q})) < \text{dim}T \quad (5)$$

and is called a singular configuration.

In literature there are many publications which deal with the application of adaptive systems [6, 7] or genetic algorithm (GA) [8, 9] for solving the IKP. Especially the second category of methods allows to avoid the problem of singularities. In a very interesting work [10] the authors used continuous GA for trajectory generating. The procedure is singularity robust but only the problem of generalized coordinates is considered. Renner and Ekárt [11] presented the review of applying GA in path planning. Some of the algorithms are singularity robust but in the articles only the problem of generalized coordinates generation is presented. In recent years, there has been an increase in the number of publications devoted

to the IKP. This conclusion is based on the analysis of the number of articles devoted to IKP topics in the five-year period of 1997-2006 in one of major bases of scientific journals. In the last five years the number of articles has increased by about 30%. This is probably due to the fact that despite many attempts, we have failed to find general robust methods for solving the IKP for different manipulator structures.

In this paper the singularity robust trajectory generator for robotic manipulators is presented. It is singularity robust because it is based on the direct kinematics problem (DKP). In this method there is no overt information about singularities. This trajectory generator allows to obtain generalized coordinates, velocities and accelerations.

2. Optimizing problem formulation

Let us assume that the manipulator will be non-redundant and x describes only the position of the end-effector in the task space $x_1x_2x_3$ (Fig. 3).

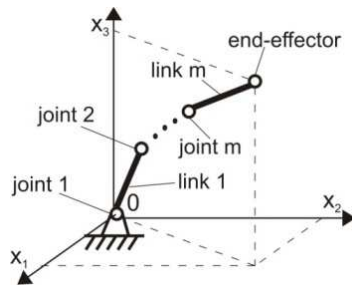


Fig. 3. Schema of manipulator

Projecting the equation (1) onto the axis of the frame of reference yields system of equations which describes the position of manipulator's end-effector. Calculating its i^{th} time derivatives yields the following description of the kinematics:

$$\mathbf{x}^{(i)} = \mathbf{f}(\mathbf{q}, \dots, \mathbf{q}^{(i)}), \quad i = 0, 1, \dots, p, \quad p = 3 \quad (6)$$

Equation (6) constitutes a mathematical model of DKP.

The IKP consists in calculating generalized coordinates \mathbf{q} , velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ for the desired path and desired velocity of manipulator's end-effector in the task space. For this trajectory and velocity in the task space we may calculate coordinates $\mathbf{x}^{(0)} = \mathbf{x} = [x_1, x_2, x_3]^T$, projections of velocity $\mathbf{x}^{(1)} = \dot{\mathbf{x}} = [\dot{x}_1, \dot{x}_2, \dot{x}_3]^T$, acceleration $\mathbf{x}^{(2)} = \ddot{\mathbf{x}} = [\ddot{x}_1, \ddot{x}_2, \ddot{x}_3]^T$ and acceleration's time derivative $\mathbf{x}^{(3)} = \dddot{\mathbf{x}} = [\dddot{x}_1, \dddot{x}_2, \dddot{x}_3]^T$ onto the axis of the frame of reference that is $\mathbf{x}^{(i)}$. These variables are connected with $\mathbf{q}^{(i)}$ by equation (6).

The IKP is defined as an optimization problem, where a fitness function (error function) for each trajectory point defined by (7) is optimized:

$$E = \sum_{i=0}^p w_i \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|^2 \quad (7)$$

where: $\hat{\mathbf{x}}^{(i)}$ – approximation of $\mathbf{x}^{(i)}$,
 $\|\cdot\|$ – Euclidean norm of a given vector,
 w_i – unit value weight.

By using (5) we finally obtain the fitness function in the following form

$$E(\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}}) = \sum_{i=0}^p w_i \|\mathbf{x}^{(i)} - \mathbf{f}(\hat{\mathbf{q}}, \dots, \hat{\mathbf{q}}^{(i)})\|^2 \quad (8)$$

where: $\hat{\mathbf{q}}^{(i)}$ – approximation of $\mathbf{q}^{(i)}$.

The optimization is carried out by a procedure based on the GA [11-15]. The GA searches for a minimum of the fitness function towards $\hat{\mathbf{q}}^{(i)}$ and values of $\mathbf{f}(\hat{\mathbf{q}}, \dots, \hat{\mathbf{q}}^{(i)})$ are calculated from DKP. The presence of terms $\hat{\mathbf{q}}$, $\hat{\dot{\mathbf{q}}}$ and $\hat{\ddot{\mathbf{q}}}$ in the fitness function is natural because the solution of the IKP consists of generalized coordinates, velocities and accelerations. The term $\hat{\ddot{\mathbf{q}}}$ in the fitness function makes that acceleration $\hat{\ddot{\mathbf{q}}}$ is smooth.

The idea of solving the problem results from the following reason. It is a well-known fact that if the desired trajectory in task space is a continuous function of time, the IKP has a solution for each moment of time but it may be ambiguous or non-continuous (Fig. 4). However, it is known that in the physical system continuous changes of coordinates, velocities and accelerations in task space are associated with continuous changes of coordinates, velocities and accelerations in joint space. A physical point cannot have two or more different coordinates, velocities and accelerations at the same time. Furthermore, coordinates, velocities or accelerations of a physical object belong to a bounded set. In other words, non-continuous solutions are caused by a mathematical description, not by specificity of a real object.

According to the above reasoning, there are practically not very large changes (Fig. 4b) of generalized coordinates, velocities and accelerations even in singularities, and solutions of IKP in neighbouring moments of time are similar (Fig. 4a). Therefore, we can assume the highest changes of generalized coordinates, velocities, accelerations and acceleration's time derivatives, i.e.: Δq_{jMAX} , $\Delta \dot{q}_{jMAX}$, $\Delta \ddot{q}_{jMAX}$ and $\Delta \ddot{\ddot{q}}_{jMAX}$ respectively, where $j = 1, 2, \dots, m$ denotes manipulator's joint. Their value may be associated with properties of manipulator or with constraints imposed by the user. Therefore, the GA may be used for searching a solution in a certain neighbourhood of the previous solution only. The neighbourhood is defined by the highest changes in parameters of motion. In the centre of this

region there is the previous solution. It ensures that the trajectory of manipulator will be continuous even in singularities because the solution is constrained (in this method there are not singularities, the expression „singularities” is used symbolically to define a region in which the singularities exists in classical methods). In other words, the GA searches the solution in a certain small region of search, and the region is changing in time and, in a way, moving in the space over the curve that is the appropriate solution. The current encoding of the population of solutions along the trajectory has been referred to by the author as *dynamic encoding*. Compared to *static encoding* performed for the whole task space, the dynamic approach allows for greater accuracy and smoothness of the solutions. This approach allows to obtain the applicable desired trajectory of the manipulator. It should be added that the procedure requires the initial condition for $\hat{\mathbf{q}}$. The initial conditions for $\hat{\mathbf{q}}$, $\hat{\dot{\mathbf{q}}}$ and $\hat{\ddot{\mathbf{q}}}$ will practically be zero. For each trajectory point a trajectory error $e_p(k)$ and a velocity error $e_v(k)$ in task space are defined as:

$$e_p(k) = \|\mathbf{x} - \hat{\mathbf{x}}\| \quad (9)$$

$$e_v(k) = \|\dot{\mathbf{x}}\| - \|\hat{\dot{\mathbf{x}}}\| \quad (10)$$

where k – a number of path points.

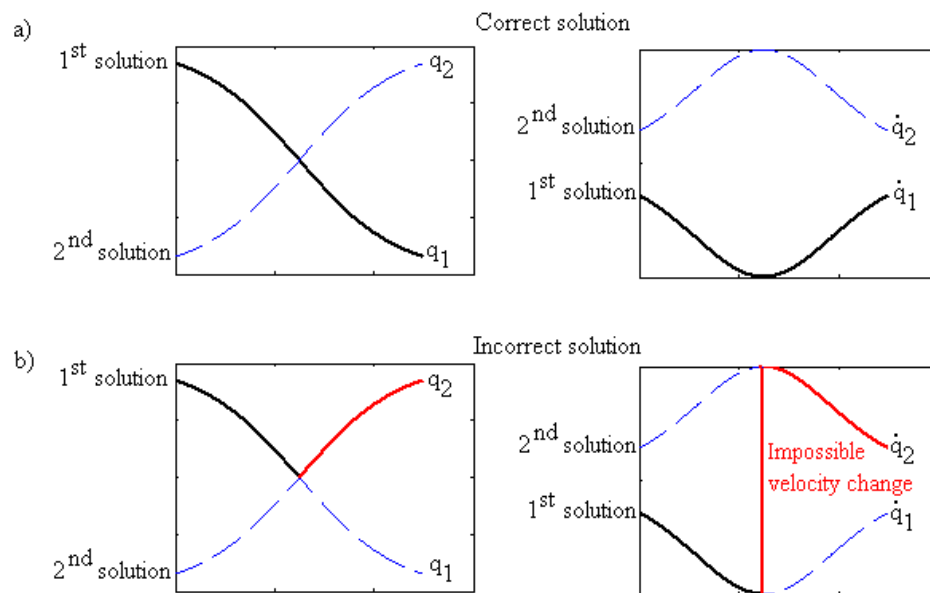


Fig. 4. The idea of IKP solution: a) physically realizable solution, b) physically unrealistic solution

Moreover, quality ratings are defined as:

$$E_P = h\sqrt{\sum_k [e_P(k)]^2} \quad (11)$$

$$E_V = h\sqrt{\sum_k [e_V(k)]^2} \quad (12)$$

where: k – a number of path points,

h – the time step.

This allows to evaluate quality of the solution.

3. Genetic algorithm implementation

In the optimizing problem the GA is used, which contains the following operations: encoding, initialization, evaluation, selection, crossover and mutation. The GA carries out operations on population of chromosomes which consist of genes. For a given path point k , encoding, initialization and evaluation should be carried out. After that, it should be checked whether the fitness is acceptable. If the fitness is sufficient, the result for a given path point is generated. If the fitness is unsatisfactory, selection, crossover and mutation are carried out. Next the evaluation is carried out and the cycle is repeated. Based on the numerical tests one change was made in the basic GA., namely, if the value of the fitness function of the best chromosome was larger than the prescribed threshold value E_T and it did not decrease during the optimization the mutation probability is temporarily increased. It improves optimization results.

In the first operation i.e. encoding, permissible solutions are encoded in the form of vectors Q_j^i which store values of variables $q_j^{(i)}$. For example, if the variable $q_1 \in \langle q_{1MIN}, q_{1MAX} \rangle$, a given set will be digitized with a constant step and the vector will store discrete values from the set. A number of discrete values depends on the number of genes in the chromosome which is allotted to encoding values of a variable. For example, if a number of genes allotted to encoding q_1 is equal to g_1^0 , the vector will store $2^{g_1^0}$ digitized value of q_1 . Values of variables are encoded in chromosomes. The binary system is used for encoding; therefore, each gene may store values 0 or 1. It is shown in Fig. 5.

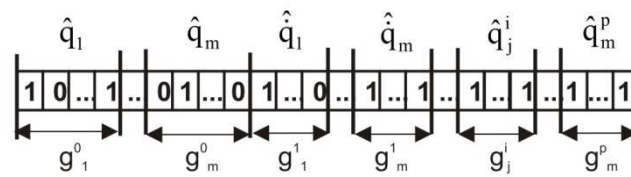


Fig. 5. Manner of encoding

In genes from 1 to g_1^0 the discrete value of a generalized coordinate q_1 is encoded. In the following genes discrete values of generalized coordinates, velocities, accelerations and accelerations' time derivatives are encoded. Therefore, a chromosome is a vector which is a candidate for representation solution of the IKP. For example, if a chain of genes which is allotted to encoding q_1 has a form of |1|0|0|1|1|, then the part of chromosome is associated with the value which is stored in the vector Q_1^0 on position $1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 25$.

A number of genes depends on the desired precision of solution and manipulator's degrees of freedom. Each chromosome has a number of genes described as:

$$g = \sum_{i=0}^p \sum_{j=1}^m g_j^i \quad (13)$$

where: g_j^i – a number of genes allotted to encoding i^{th} time derivatives of i^{th} generalized coordinates.

During initialization an initial population with even number of the chromosome H is generated. Values 0 or 1 are randomly generated and stored in genes.

In the next operation i.e. evaluation, the degree of adaptation to problem is checked. Information about them is taken from the fitness function $E(\hat{q}, \hat{q}, \hat{q}, \hat{q})$. If a value of fitness function for a given chromosome is lower, then the degree of chromosome adaptation to minimization fitness function is larger. With the aim of calculating the value of fitness function, values of generalized coordinates, velocities, accelerations and acceleration's time derivatives should be decoded. Then, these values first should be used for determining $f(\hat{q}, \dots, \hat{q}^{(i)})$ from the DKP, and next the value of fitness function $E(\hat{q}, \hat{q}, \hat{q}, \hat{q})$ for each chromosome should be calculated. A chromosome for which the fitness function has the lowest value is the best chromosome in a given group.

In order to create a new generation of chromosomes, a parental pool is selected from the current population. In order to do that, the tournament selection is used. In this method a sub-group of chromosomes is randomly chosen (with repetition) from population. In the articles [12, 14], sub-groups have 2 or 3 chromosomes, but results of numerical tests show that the procedure converges much faster if the size of the sub-group r is dependent on the size of population H in this way:

$$r = \text{round}(0.6\sqrt{H} + 0.4) \quad (14)$$

Then the chromosomes from the sub-group are evaluated and next one chromosome is selected and added to the parental pool. This selection should be performed H times. Then, the chromosomes from the parental pool are randomly coupled.

New chromosomes are created by using one-point crossover with the probability $P(c) = 1$. A crossover point for each pair of parents is generated randomly. New chromosomes have features like their parents because offsprings are their combination. In this way, the next generation of chromosomes is created from the parental pool. After that, an old population is replaced by a new population. Finally, mutation is carried out. In this operation values stored in the genes change from 1 to 0 or from 0 to 1. The mutation occurs with the probability m .

4. Simulation results

In this chapter kinematics of a two-degrees-of-freedom planar manipulator with rotational joints, which is schematically presented in Fig. 6, is considered. It is assumed that movement took place in the xy plane.

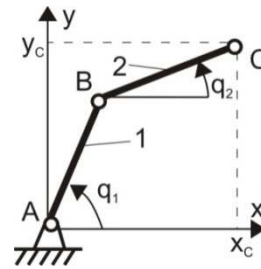


Fig. 6. The two-degrees-of-freedom planar manipulator with rotational joints

Point C is chosen as the particular manipulator's point. It is also referred to as an end-effector. Its coordinates are described by the system of equations:

$$\begin{cases} x_C = l_1 \cos q_1 + l_2 \cos q_2 \\ y_C = l_1 \sin q_1 + l_2 \sin q_2 \end{cases} \quad (15)$$

where: x_C, y_C – coordinates of the point C,

$l_1 = 0.22$ m, $l_2 = 0.22$ m – lengths of manipulator's link 1 and 2,

q_1, q_2 – joint angles of link 1 and 2 (generalized coordinates).

By calculating time derivatives (15), we obtain equations of velocities, accelerations and accelerations' time derivatives:

$$\begin{cases} \dot{x}_C = -l_1 \dot{q}_1 \sin q_1 - l_2 \dot{q}_2 \sin q_2 \\ \dot{y}_C = l_1 \dot{q}_1 \cos q_1 + l_2 \dot{q}_2 \cos q_2 \end{cases} \quad (16)$$

$$\begin{cases} \ddot{x}_C = -l_1 \ddot{q}_1 \sin q_1 - l_1 \dot{q}_1^2 \cos q_1 - l_2 \ddot{q}_2 \sin q_2 - l_2 \dot{q}_2^2 \cos q_2 \\ \ddot{y}_C = l_1 \ddot{q}_1 \cos q_1 - l_1 \dot{q}_1^2 \sin q_1 + l_2 \ddot{q}_2 \cos q_2 - l_2 \dot{q}_2^2 \sin q_2 \end{cases} \quad (17)$$

$$\begin{cases} \ddot{x}_C = l_1 \dot{q}_1^3 \sin q_1 - 3l_1 \dot{q}_1 \ddot{q}_1 \cos q_1 - l_1 \ddot{q}_1 \sin q_1 + \\ \quad + l_2 \dot{q}_2^3 \sin q_2 - 3l_2 \dot{q}_2 \ddot{q}_2 \cos q_2 - l_2 \ddot{q}_2 \sin q_2 \\ \ddot{y}_C = -l_1 \dot{q}_1^3 \cos q_1 - 3l_1 \dot{q}_1 \ddot{q}_1 \sin q_1 + l_1 \ddot{q}_1 \cos q_1 + \\ \quad - l_2 \dot{q}_2^3 \cos q_2 - 3l_2 \dot{q}_2 \ddot{q}_2 \sin q_2 + l_2 \ddot{q}_2 \cos q_2 \end{cases} \quad (18)$$

For this manipulator Jacobian of velocity has the form:

$$J = \begin{bmatrix} -l_1 \sin q_1 & -l_2 \sin q_2 \\ l_1 \cos q_1 & l_2 \cos q_2 \end{bmatrix} \quad (19)$$

whose determinant has the form:

$$\det(J) = l_1 l_2 \sin(q_1 - q_2) \quad (20)$$

Therefore, singularity occurs when $q_1 = q_2 + n\pi$, but practically $n = 0$.

The highest changes of generalized coordinates, velocities, accelerations and accelerations' time derivatives are supposed as $\therefore \Delta q_{jMAX} = \pm 0.005$ rad, $\Delta \dot{q}_{jMAX} = \pm 0.005$ rad/s, $\Delta \ddot{q}_{jMAX} = \pm 0.01$ rad/s², $\Delta \ddot{\ddot{q}}_{jMAX} = \pm 0.05$ rad/s³. Numbers of genes for encoding values of variables are supposed as $g_j^0 = 3$, $g_j^1 = 4$, $g_j^2 = g_j^3 = 5$. We set the value of the population size as $H = 600$, $E_T = 5 \cdot 10^{-8}$, and the value of the mutation probability as $m = 0.05$, which is temporarily increased ten times if fitness function did not decrease during the optimization.

In order to test the method, simulation is performed. We assume generalized coordinates $q_1(t)$ and $q_2(t)$ so generalized velocities, accelerations and accelerations' time derivatives are known. Generalized coordinates are selected in such a way that the case of singularity in the IKP ($q_1 = q_2$ – regions marked by circles in Fig. 7) occurs several times.

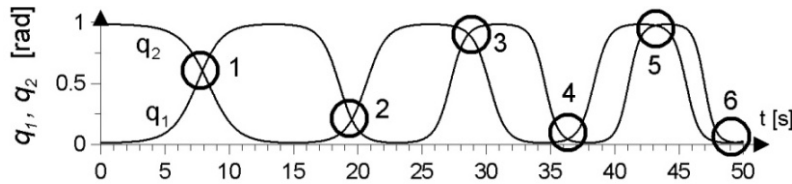


Fig. 7. Generalized coordinates and regions in which there occurs the case $q_1 = q_2$

Based on the relations (15)-(18), $\mathbf{x}^{(i)}$ is calculated, which means that mapping of the trajectories in the joint space into the trajectory in the task space is well known. Thus, the ideal solution to which the solution obtained by the proposed method is compared is known. Then, the IKP was solved. In Fig. 8 corresponding

supposed generalized movement parameters and their approximation calculated by the singularity robust trajectory generator are presented.

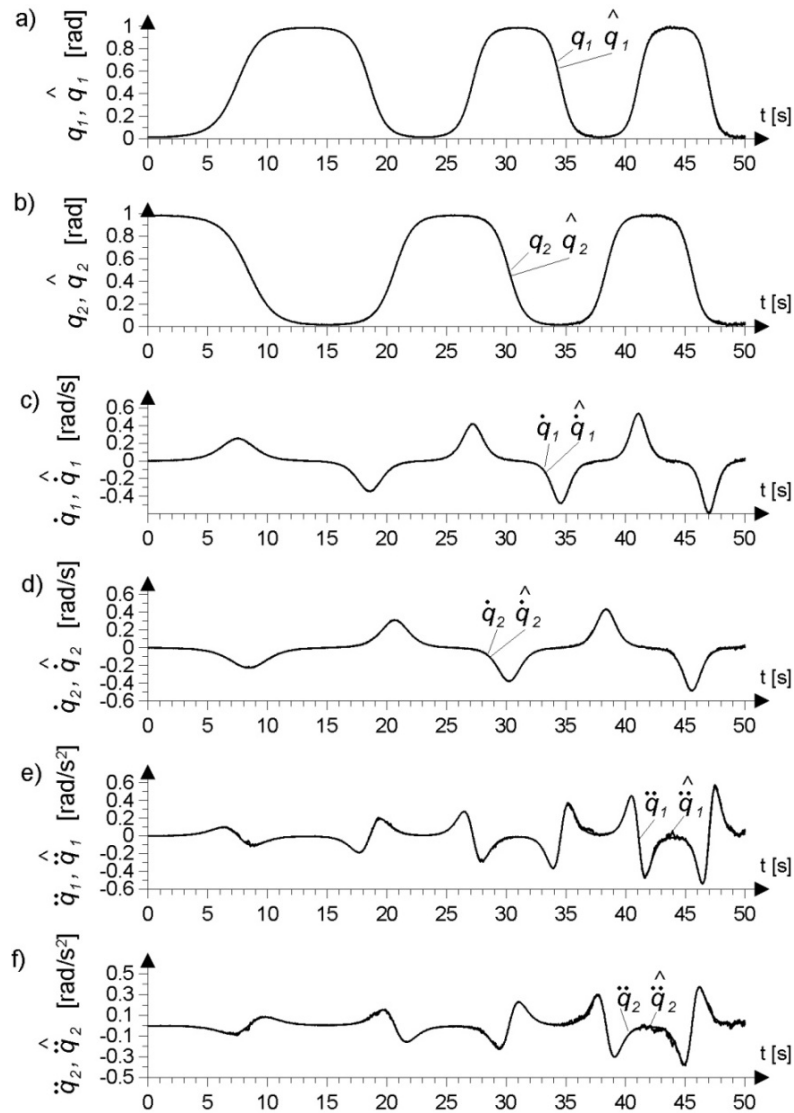


Fig. 8. The supposed generalized movement parameters and their approximation calculated by a trajectory generator

Particularly interesting are approximations of generalized movement parameters in cases when $q_1 \approx q_2$. In singularities 1, 2 and 3 the approximations are very smooth. In singularities 4, 5 and 6 ($t \approx 36$ s, 43 s, 50 s) approximations of generalized accelerations are oscillating. It is caused by very small values of

generalized velocities. When $q_1 \approx q_2$ and generalized velocities are nearly zero, then two possible solutions of IKP are very similar, and the procedure generates the trajectory with small oscillations.

In order to evaluate the solution of IKP, path error $e_p(k)$ (Fig. 9) and velocity error $e_v(k)$ (Fig. 10) are calculated. Path error defined by (9) is smaller than 0.001 m, and velocity error defined by (10) always belongs to the range $(-0.001; 0.001)$ m/s. The quality ratings for $h = 0.01$ have values: $E_p = 1.59 \cdot 10^{-4}$ m, $E_v = 1.14 \cdot 10^{-4}$ m/s. Such errors and the quality of solution are acceptable and this given trajectory may be used in the movement control of manipulator.

In the case of practical applications of a trajectory generator, a slightly different approach than the one presented in subchapter 4.1. *Test of trajectory generator* is required. In practice, a characteristic point C which belongs to the manipulator end-effector is chosen and it is defined how it should move within the workspace. For this purpose, the trajectory of the point and its velocity in the workspace is determined. Then, with the use of kinematic equations, the coordinates of the point C with regard to the frame of reference and its subsequent derivatives, that is, $x_C, y_C, \dot{x}_C, \dot{y}_C, \ddot{x}_C, \ddot{y}_C, \ddot{\ddot{x}}_C$ and $\ddot{\ddot{y}}_C$ are set. The final stage is to produce the IKP solution with the use of a trajectory generator.

Let us assume the path of the point C for the manipulator shown in Fig. 7 be

$$f_C(x_C, y_C) = 0 \quad (21)$$

The velocity of the point C in the xy reference frame put in vector notation is

$$\mathbf{v}_C = \begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} \quad (22)$$

whereas its value is defined by the equation:

$$v_C = \sqrt{(\dot{x}_C)^2 + (\dot{y}_C)^2} \quad (23)$$

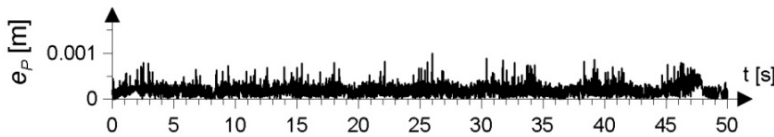


Fig. 9. Path error of the point C

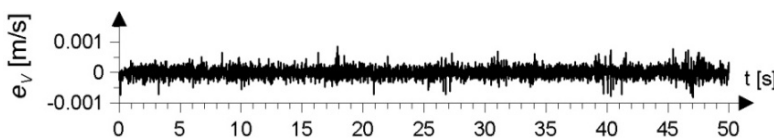


Fig. 10. Velocity error of the point C

In order to fulfil the velocity of the point C, the velocity vector must be tangent to the trajectory, thus the following equation must be satisfied:

$$\text{grad}f_C \mathbf{v}_C = 0 \quad (24)$$

Upon taking into consideration the equations (21) and (22), it shall have the form of:

$$\begin{bmatrix} \frac{\partial f_C}{\partial x_C} & \frac{\partial f_C}{\partial y_C} \end{bmatrix} \begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} = 0 \quad (25)$$

By determining $f_x = \frac{\partial f_C}{\partial x_C}$ i $f_y = \frac{\partial f_C}{\partial y_C}$ the equation:

$$f_x \dot{x}_C + f_y \dot{y}_C = 0 \quad (26)$$

was obtained. By solving the equation systems (23) and (26), the equation:

$$\begin{cases} \dot{x}_C = \mp \frac{f_y v_C}{\sqrt{f_x^2 + f_y^2}} \\ \dot{y}_C = \pm \frac{f_x v_C}{\sqrt{f_x^2 + f_y^2}} \end{cases} \quad (27)$$

was obtained. Solving the system of differential equations (27) enables to determine how velocity vector components and the coordinates of the point C change. Moreover, it enables to determine how higher derivatives of motion parameters for the defined path and velocity values of this point change.

In the following example, a trajectory in the form of an ellipse is assumed:

$$f_C = \left(\frac{x_C - x_0}{a}\right)^2 + \left(\frac{y_C - y_0}{b}\right)^2 - 1 = 0 \quad (28)$$

for which let $f_x = 2 \frac{x_C - x_0}{a^2}$ and $f_y = 2 \frac{y_C - y_0}{b^2}$, and it is assumed that $x_0 = 0$, $y_0 = 0.22$, $a = 0.16$, $b = 0.22$. The velocity value was determined as:

$$v_C = \frac{v_{Cmax}}{1 + \exp[-c(t-5)]} - \frac{v_{Cmax}}{1 + \exp[-c(t-25)]} \quad (29)$$

where: $v_{Cmax} = 0.3$ m/s – the maximum motion velocity,

$c = 3$ s⁻¹ – constant responsible for the velocity curve gradient during accelerating and stopping.

Having assumed the initial conditions of the trajectory as $x_C(0) = 0.16$ m and $y_C(0) = 0.22$ m, the equation system (27) was solved, considering the

equations (28) and (29). As a result, the kinematic motion parameters of the point C, that is, x_C , y_C , \dot{x}_C , \dot{y}_C , \ddot{x}_C , \ddot{y}_C , \dddot{x}_C and \dddot{y}_C (Fig. 11a-d) were obtained. Then, by using the afore-discussed trajectory generator, which uses the kinematic equations (15)-(18), the IKP solution was obtained (Fig. 11f-h).

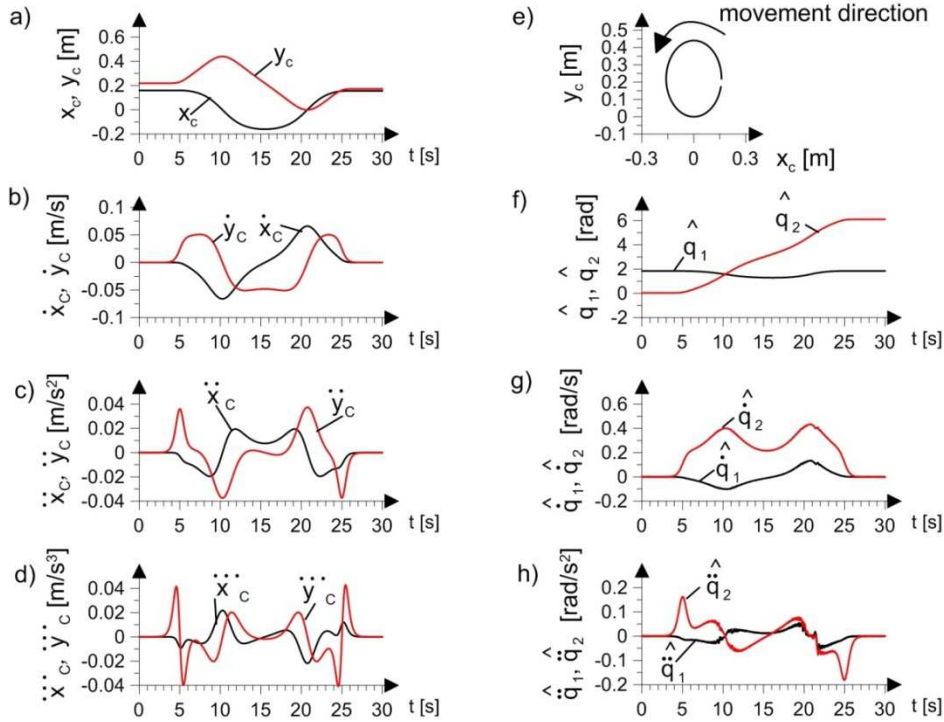


Fig. 11. The IKP solution with the use of a trajectory generator: a-d) kinematic motion parameters of the point C, e) point C path, f-h) the IKP solution

The chart in Fig. 11f shows that around 10 s we have situation when the angles of rotation of the manipulator joints are equal. This means that in the case of the classical approach, kinematic singularities would occur, whereas due to the trajectory generator which is free from singularities, arriving at the IKP solution became possible.

5. Conclusions

In this paper a trajectory generator for robotic manipulators is presented. This generator does not need the inverse of a Jacobian matrix, so it is singularity robust. Continuity of trajectory is assured by limiting changes of generalized coordinates, velocities, accelerations and accelerations' time derivatives. The IPK is formulated as an optimization problem because in this approach we may use the DKP

only. By limiting the region in which the optimization is carried out, the time of solving the IKP decreases.

The theoretical discussion is confirmed by an example in which generalized coordinates, velocities and accelerations of a two-degrees-of-freedom manipulator are generated correctly even in singularities. To the authors' knowledge, it is the first singularity robust trajectory generation algorithm that can be applied to determine generalized coordinates, velocities and accelerations of the manipulator in singularities.

This algorithm may easily be applied to IKP of other manipulators, also redundant manipulators. In such cases the fitness function should be appropriately defined.

References

- [1] de la Cueva V., Ramos F.: Cooperative genetic algorithms: A new approach to solve the path planning problem for cooperative robotic manipulators sharing the same work space, Proc. of the 1998 IEEE/RSJ Intl. Conf. Intell Robots and Sys., 1998.
- [2] Pires E.J.S., Oliveira P.B.M., Machado J.A.T.: Manipulator trajectory planning using a MOEA, Appl. Soft. Comput., 7 (2007) 659-667.
- [3] Dułęba I., Hossa R., Mazur A., Muszyński R., Tchoń K.: Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie, PLJ, Warszawa 2000.
- [4] Żylski W., Gierlak P.: Generator zadanej trajektorii ruchu obiektu dynamicznego, ZN PRz, seria: Mechanika, 74 (2008) 421-432.
- [5] Żylski W., Gierlak P.: Zadanie odwrotne kinematyki manipulatora w ujęciu Rozszerzonej Mapy Kohonena, Prace Naukowe Politechniki Warszawskiej, seria: Elektronika, 166 (2008) 495-504.
- [6] Hasan A.T., Hamouda A.M.S., Ismail N., Al-Assadi H.M.A.A.: An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot manipulator, Adv. Eng. Softw., 37 (2006) 432-438.
- [7] Vargas L.V., Leite A.C., Costa R.R.: Overcoming kinematic singularities with the filtered inverse approach, IFAC Proc. Vols., 47 (2014) 8496-8502.
- [8] El-Sherbiny A., Elhosseini M.A., Haikal A.Y.: A comparative study of soft computing methods to solve inverse kinematics problem, Ain Shams Eng. J., 2017, <http://dx.doi.org/10.1016/j.asej.2017.08.001>
- [9] Köker R.: A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, Inf. Sci., 222 (2013) 528-543.
- [10] Abo-Hammour Z.S., Mirza N.M., Mirza S.M., Arif M.: Cartesian path generation of robot manipulators using continuous genetic algorithms, Rob. Auton. Syst, 41 (2002) 179-223.
- [11] Renner G., Ekárt A.: Genetic algorithms in computer aided design, Comput. Aided Des., 35 (2003) 709-726.
- [12] Chakraborty M., Chakraborty U.K.: An analysis of linear ranking and binary tournament selection in genetic algorithms, Proc. ICICS'9 1997.

- [13] Rutkowska D., Piliński M., Rutkowski L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, PWN, Warszawa 1999.
- [14] Rutkowski L.: Metody i techniki sztucznej inteligencji, PWN, Warszawa 2005.
- [15] Zhong J., Hu X., Gu M., Zhang J.: Comparison of performance between different selection strategies on simple genetic algorithms, Proc. CIMCA-IAWTIC'05, 2005.

ODPORNY NA OSOBLIWOŚCI GENERATOR TRAJEKTORII ROBOTA MANIPULACYJNEGO OPIERAJĄCY SIĘ NA ALGORYTMIE GENETYCZNYM Z DYNAMICZNYM KODOWANIEM ROZWIĄZAŃ

Streszczenie

W niniejszym artykule przedstawiono generator trajektorii robota manipulacyjnego odporny na osobliwości kinematyki. Generator zawiera procedurę rozwiązywania zadania odwrotnego kinematyki. Problem ten jest zdefiniowany jako problem optymalizacyjny, w którym algorytm genetyczny służy do optymalizacji funkcji dopasowania opierającej się na błędzie rozwiązania zadania. Aby wyeliminować problem osobliwości, generator wykorzystuje zadanie proste kinematyki. Generator trajektorii umożliwia uzyskanie uogólnionych współrzędnych, prędkości i przyspieszeń manipulatora. Wyniki symulacji wskazują, że opracowana procedura generuje trajektorię manipulatora nawet w przypadku wystąpienia osobliwości kinematyki.

Słowa kluczowe: zadanie odwrotne kinematyki, algorytm genetyczny, osobliwości, robot manipulacyjny, generator trajektorii

DOI: 10.7862/rm.2018.40

Otrzymano/received: 7.06.2018

Zaakceptowano/accepted: 25.09.2018