ROBOT TRAJECTORY DESIGN USING GENETIC ALGORITHM IN MATLAB

I.Sekaj, A.Husár

Institute of Robotics and Cybernetics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic

Abstract

The contribution deals with the design of a robotic arm optimal trajectory calculation in the Matlab/Simulink environment. The design is based on inverse kinematics problem solving using robot model simulation with the optimization procedure based on genetic algorithm. The precision of the robot operating point reaching is maximized concurrently with minimization of all joint rotations.

1 Introduction

The optimization of the robotic arm trajectory is a frequent design problem in the industry. Hundred-thousands of robots make many thousands operation cycles each day to produce various products or services. Thus the optimization of the robot moving is a problem with a huge economic impact. Also small improvements can bring considerable energy savings.

In the presented approach, the robotic arm trajectory design is based on inverse kinematics problem solving (IKP). The IKP of the robot represents the dependency between the joint variables and the coordinates of the end effector (the end point of the arm). The transition between point *A* and *B* in the 3D working space has infinite number of possible and non-optimal solutions. An additional consideration of other optimization criterions as operation time, energy, joint rotations or others into the IKP becomes a very difficult task, almost insolvable using conventional methods. Conventional approaches entailed only suboptimal solutions. We propose the solving of the IKP with additional criterions by means of the Genetic Algorithm (GA). This approach optimizes the robot trajectory from point of view positioning precision concurrently with a chosen other criterion and it is able to reach or to converge to the global optimum of the task. Robot trajectory designs based on GAs were addressed by several authors as Davidor [1], Juang [2], in Pires [3] and others. A design procedure of optimal robot trajectory design was proposed in Sekaj [4], where a mathematical representation of ICP was used. In the presented contribution the mathematical transformation is replaced by a simulation model in Simulink/SimMechanics environment. The use of the robot model as introduced in this paper enables to consider dynamics of the manipulator, calculation of energy, operation time and other important properties, which were unknown in the simple kinematic model.

2 Robot trajectory optimal design

Consider a robot with *n* degrees of freedom. In case of the used robot ABB IRB 4600 *n* is equal to 6 (Fig.1). The joint rotation angles $\omega,\theta,\psi,\phi,\rho,\varepsilon$ performs $N=3$ operations from basic position P_0 to P_1, from P_1 to P_2 and from P_2 to P_3. The points are defined by the user in a clockwise Cartesian coordinate system $P_i[x_i;y_i;z_i]$. Each operating point, which is the end effector of the manipulator, is characterized by *n*-angles of the joint rotation

$$P_i[x_i;y_i;z_i] \leftrightarrow f(\omega_i,\theta_i,\psi_i,\phi_i,\rho_i,\varepsilon_i);i\in\{1,2,3\}. \quad (1)$$

According to (1), the aim of the inverse kinematics problem for the execution of a single robot cycle is the visiting of the desired *N* points by the effector of the robot. The goal is the search for a sequence of *N* vectors of *n* angles, which connects the desired operating points. The problem is that there is infinite number of possible trajectories for connection of the *N* points. But if we connect the trajectory design with another optimization criterion as energy, time or rotation, the problem has a global optimum.

Among the possible additional criterions the joint rotation minimization has been considered in our case. The positioning precision of a potential trajectory can be evaluated using the expression

$$D = \sum_{j=1}^{N} \sqrt{\begin{matrix}[x_{w,j} - x_j]^2 + [y_{w,j} - y_j]^2 \\ +[z_{w,j} - z_j]^2\end{matrix}} \qquad (2)$$

which represents the Euclidean distance of the desired coordinates (marked *w*) from the by model simulation reached coordinates in all required points P_j, $j=1,2,...,N$. Next the rotations are minimized by considering sum of all rotations of all joints during the operation cycle. The criterion for the movement between two points *a* and *b* is defined as

$$A_{p2p} = \sum_{i=1}^{n} (\alpha_{b,i} - \alpha_{a,i}) \qquad (3)$$

where α_i is each i-th particular joint angle. Movement between all *N* points is

$$A = \sum_{j=1}^{N} \sum_{i=1}^{n} (\alpha_{b,i,j} - \alpha_{a,i,j}) \,. \qquad (4)$$

The goal of the design is to find such a trajectory, which minimizes concurrently (2) and (4). Let the final criterion for the optimal trajectory design is in form

$$F=D+\beta A \qquad (5)$$

In the GA terminology the criterion function (*F*) is called "fitness". The positioning precision *D* should be in practice in range fractions of *mm* (say $D<0.001mm$). The *A* is the summary rotation of all joints during the operation cycle which is in range thousands of degrees. From that reason let us set the weight constant $\beta=10e^{-7}$.

Because of the high complexity of such problem, conventional optimization methods are able to find only some sub-optimal solution (local optimum). From that reason genetic algorithm has been used to solve this task.

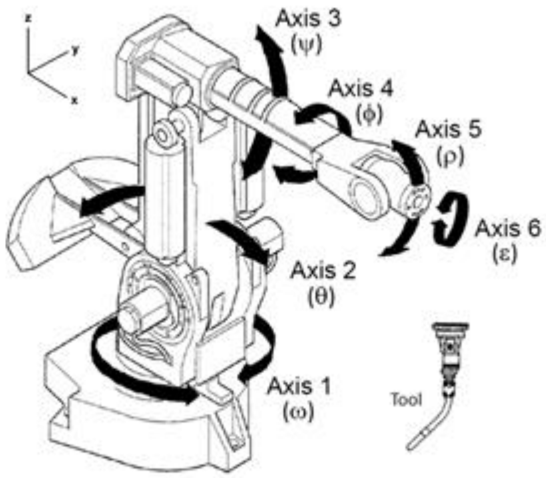

Figure 1: Robotic arm ABB IRB 4600 with 6 degrees of freedom

3 Robot model in Simulink

For the criterions (2) and (4) evaluation the model simulation of the robot has been used. Construction data of the manipulator ABB IRB 4600 from the producer were used to create the simulation model in the Simulink/SimMechanics environment (Fig.2). In Fig.3 the CAD model is shown, which has been used for animation reasons in Matlab/Simulink. For next possible editing of the model the Matlab/SimScape library can been used.

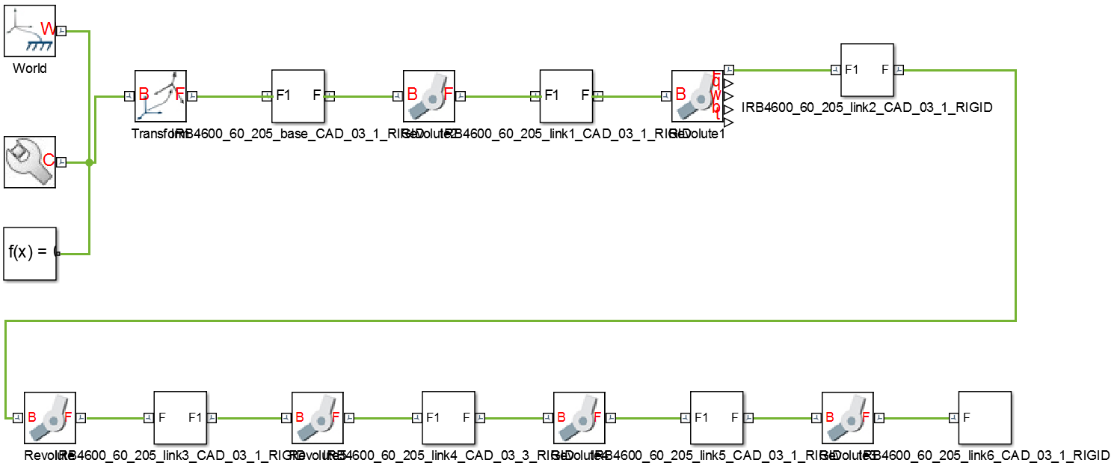

Figure 2: Model of the robot ABB IRB 4600 in Simulink/SimMechanics

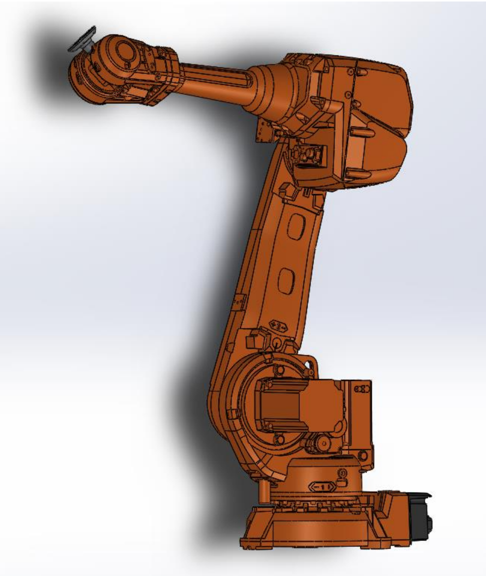

Figure 3: CAD model of the robot ABB IRB 4600

4 Genetic algorithm

The GA is an optimisation approach which mimics the natural evolution [5,6,7]. The algorithm operates over a set of potential solutions (a population). Each i-th potential solution (individual of the population) is represented as the vector in form

$$I_i=(\omega_1,\theta_1,\psi_1,\phi_1,\rho_1,\varepsilon_1,\omega_2,\theta_2,\psi_2,\phi_2,\rho_2,\varepsilon_2,\omega_3,\theta_3,\psi_3,\phi_3,\rho_3,\varepsilon_3)$$

where items of the vector are the rotations of each n joints during transition to each of the N operating points. A general scheme of the used GA can be described by following steps:

1. Initialisation of the population of individuals.
2. Fitness evaluation of the population (simulation and calculation of (5)).

3. End if terminal conditions are satisfied.
4. Selection of parent individuals *parents₁* for crossover and *parents₂* for mutation.
5. Selection of individuals *unchanged* which will not be changed, including the best fit individual.
6. Crossover of the $parents_1 = children_1$.
7. Mutation of the $parents_2 = children_2$.
8. Unification of $children_1$, $children_2$, and *unchanged* individuals as the new population.
9. Continue in step 2.

The crossover operator combines items (genes) of two parent individuals by random and produces two new childs. The mutation operator changes by random a parent individual to produce a new child individual.

5 Results

The population size used was 50 individuals. In Fig.4 the graph of the fitness function evolution is depicted. On the horizontal axis is the number of generations (computation cycles), on the vertical axis is the fitness value of the currently best fit individual of the population. After 9000 generations we got the value 0.0013 as the resulting fitness value. In Tab.1 the desired and obtained positions of the robotic arm are presented. In Fig. 5 the optimal trajectory is shown for all 3 motions.

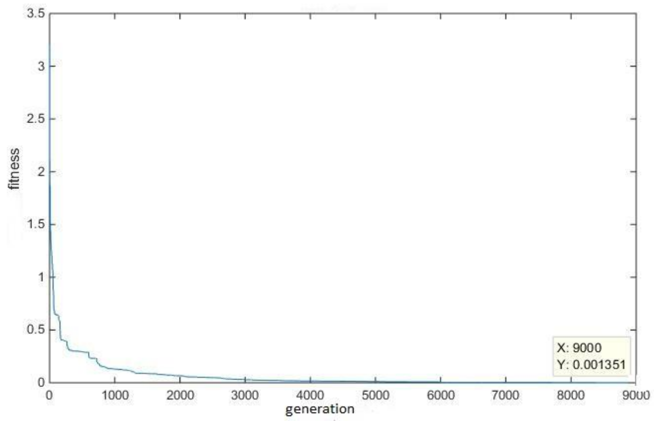

Figure 4: Graph of fitness function (5) evolution

Table 1. Positioning accuracy

point	desired position [m]	reached position [m]
P_1	0.3	0.29999
	1	1.00004
	1.5	1.49990
P_2	-1	-1.00018
	-0.5	-0.49997
	-0.5	-0.50001
P_3	1	0.99966
	1	0.99996
	1	0.99922

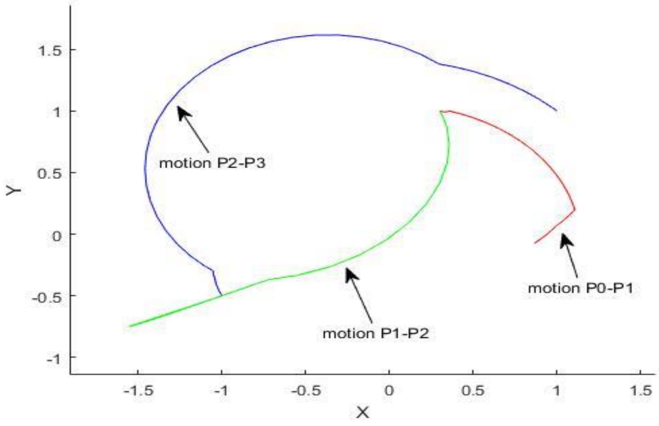

Figure 5: Graphs of the optimal trajectory for all 3 motions of the robot, projections into the XY plain

6 Conclusion

A genetic algorithm – based design procedure for a robotic manipulator was proposed. The method is able to design optimal trajectory of the manipulator, which operates in defined points in the Euclidean space. Various optimisation criterions can be considered in connection with the positioning precision as operating time, energy consumption, joint rotation or combination of all these aspects. The criterion evaluation, which represents the fitness function of the GA is based on simulation of the robot model in the Simulink/SimMechanics environment. The obtained positioning precision is sufficient and the trajectory is characterised by the optimal (minimal) summary rotation of all joints of the robot manipulator.

Acknowledgement

The work has been supported by the grant agency VEGA no. 1/0475/16. This support is very gratefully acknowledged.

References

[1] Y. Davidor. *Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization.* World Scientific, 1991.

[2] J. G. Juang. *Application of Repulsive Force and Genetic Algorithm to Multi-manipulator Collision Avoidance*. http://ascc2004.ee.mu.oz.au/proceedings papers/P142.pdf, 2004.

[3] S. Pires, J. A. Machado, P. B. De Oliveira. *Robot Trajectory Planning Using Multi-objective Genetic Algorithm Optimization.* In *GECCO, LNCS 3102*, pp. 615–626, Springer-Verlag Berlin Heidelberg, 2004.

[4] S. Števo, I. Sekaj, M. Dekan. *Optimization of Robotic Arm Trajectory Using Genetic Algorithm.* In Proceedings of the 19th World Congress of the International Federation of Automatic Control. IFAC, 2014.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addisson-Wesley, 1989.

[6] I. Sekaj I. *Evolučné výpočty a ich využitie v praxi*. Iris Bratilava, 2005

[7] I. Zelinka, Z. Oplatková, M. Šeda, P. Ošmera, F. Včelař. *Evoluční výpočetní techniky - Principy a aplikace*. BEN, Praha, 2009.

Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
e-mail: ivan.sekaj@stuba.sk

Adam Husár
Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
xhusara@is.stuba.sk