Optimization of Robotic Arm Trajectory Using Genetic Algorithm

Stanislav Števo. Ivan Sekaj. Martin Dekan.

Institute of Control and Industrial Informatics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(e-mail: stanislav.stevo@stuba.sk, ivan.sekaj@stuba.sk, martin.dekan@stuba.sk)

Abstract: The paper presents a genetic algorithm - based design approach of the robotic arm trajectory control with the optimization of various criterions. The described methodology is based on the inverse kinematics problem and it additionally considers the minimization of the operating-time, and/or the minimization of energy consumption as well as the minimization of the sum of all rotation changes during the operation cycle. Each criterion evaluation includes the computationally demanding simulation of the arm movement. The proposed approach was verified and all the proposed criterions have been compared on the trajectory optimization of the industrial robot ABB IRB 6400FHD, which has six degrees of freedom.

Keywords: Robotic arm trajectory, genetic algorithm, inverse kinematics problem, energy consumption minimization, operating time minimization, joint rotation minimization.

1. INTRODUCTION

The optimisation of the robotic arm trajectory is a frequent design problem. Because of the complexity of this task in the past, many of the proposed approaches entailed only a suboptimal solution. Due to that reason, previously, several authors have used evolutionary algorithms. Rana and Zalzala (1997) applied EA to the collision-free path planning of the robotic arm. In Garg & Kumar (2002), the formulation and application of Genetic Algorithm and Simulated Annealing for the determination of an optimal trajectory of a multiple robotic configuration is presented. In Park et al. (1999) a method for optimal trajectory control using the Evolution Strategy is proposed. In the first step, the optimal trajectory based on the cubic polynomials under certain physical constraints is determined. In the second step, the fuzzy controller is optimized to precisely track the determined trajectory. Davidor (1991) uses Genetic Algorithms with regards to the trajectory generation by searching the inverse kinematics solutions to pre-defined paths of end-effectors. In Juang (2004) a multi-manipulator collision avoidance using Genetic Algorithms is presented, the safety distance between objects is affected by the repulsive force gain and real-time manipulator collision avoidance control has been achieved. Multi-objective Genetic Algorithm for generating manipulator trajectories considering obstacle avoidance is proposed in Pires (2004). The results are presented for robots with two and three degrees of freedom (DOF), considering two and five optimization objectives. An overview of using evolutionary algorithms in controller design and robotics can be found in Sekaj (2011).

In the presented approach, the robotic arm trajectory design is based on inverse kinematics problem solving (IKP) with connection to further optimizations of selected criterions. The inverse kinematics problem of the robot – the dependency between the joint variables and the coordinates of the end effector (or the end point of the arm) represents a complex problem with infinite number of possible solutions. The more DOF the robot has the more complex the calculation of the IKP is. An additional consideration of other optimization criterions into the IKP becomes a very difficult task, almost insolvable using conventional methods e.g. Garg & Kumar (2002), Juang (2004), Vigrala et al. (2013). We propose the solving of the IKP using additional criterions, which make the problem solvable with a powerful optimisation approach - the Genetic Algorithm (GA). Three additional optimisation criterions (together with the positioning accuracy) are considered. These are the minimisation of energy consumption, minimisation of operation time as well as the criterion of minimal total angular changes of the robotic arm.

2. INVERSE KINEMATICS PROBLEM

A robot with n degrees of freedom (Fig.1) has the joint rotation angles α_1, α_2, ..., α_n and performs N operations in points P_1 to P_N defined in a clockwise cartesian coordinate system $P_i[x_i;y_i;z_i]$. Each operating point (end effector of the manipulator) is characterized by n-angles of the joint rotation

$$P_i[x_i;y_i;z_i] \leftrightarrow f(\alpha_{1i}, \alpha_{2i},...,\alpha_{ni}); i \in \{1,2,3,...,N\}. \qquad (1)$$

According to (1), the aim of the inverse kinematics task for the execution of a single robot cycle is the search for a sequence of N vectors of angles, which characterize the desired operating points. In general, using conventional design methods under consideration of additional criterions, it is not possible to solve the inverse kinematic task for a robotic arm with n degrees of freedom e.g. Pac et al. (2013), Vigala et al. (2013). To solve this problem it is possible to use other approaches as evolutionary algorithms. Because the number of search parameters in such formulation of the task is constant, the Genetic Algorithm can be used. Each potential solution of the problem represents a vector of values

$$S=\{\alpha_{11}, \alpha_{21},..., \alpha_{n1}, ... , \alpha_{1N}, \alpha_{2N},..., \alpha_{nN}\}, \tag{2}$$

Each potential solution (individual) represents the rotation of all robotic joints, which move in terms of the end effector between points $P_1$to P_N. The range of values for each parameter $\alpha_{j,i}$ is from the interval $\langle \lambda_{\min}, \lambda_{\max} \rangle$, which is the rotation range of each particular arm joint. The optimal solution is such a vector S^*, which minimises the selected criterion.

3. GENETIC ALGORITHM

The genetic algorithm (GA) e.g. Goldberg (1989), Eiben (2007) and others is a powerful, stochastic based search/optimization approach, which imitates biological evolution. It is based on the following steps:

1. initialization of the population (set of individuals),
2. fitness function (criterion) calculation of each individual of the population,
3. if termination conditions are met, then finish (in our case - predefined number of generations), else continue in step 4,
4. parent selection (more fit individuals have higher probability to be selected, in our case the stochastic universal sampling was used, 70% of individuals of the population were selected),
5. modification of parents by crossover and mutation = children,
6. completion of the new population (children + selected unchanged individuals),
7. continue in step 2.

An individual is a string containing parameters of the optimized object. In our case, the individual is in the form (2). Mutation is an operation, where a parent individual is randomly changed (mutation rate was 0.1). Crossover is an operation, where properties of two parent individuals are randomly combined to produce a child (crossover rate used was 0.7). The fitness function evaluation contains the calculation (or a simulation) of the robot movement and the cost function evaluation. The cost function in our case contains at least two particular criterions. The population size used in our case was set as quadruple of the gene number.

3.1 Particular optimisation criterions

The choice of the objective function will have a determining influence on the final solution. The robot positioning optimization has many aspects. Next, the selected optimisation criterions are explained: minimizing of the operation cycle time, energy consumption and sum of all robotic arm rotation angles during an operating cycle.

3.1.1 Energy

The Energy criterion represents the minimization of energy consumed by the robot handling a working tool (or a load). The rotations of arms are forced by motors with rated outputs of EP_1, EP_2 to EP_N. Energy consumed between two operational points counts as the energy consumed by motors during transition from one operating point to another. Energy consumed between two working points (p2p - point to point) P_a and P_b is then determined as

$$E_{p2p} = \sum_{i=1}^{n} [(\alpha_{b,i} - \alpha_{a,i})] * EP_{r,i} \tag{3}$$

where $EP_{r,i}$ is the required energy of i-th joint rotation per 1°.

Energy needed for the entire trajectory of the robot per cycle (with N points) is given as

$$E_{tr} = \sum_{j=1}^{N} \sum_{i=1}^{n} [(\alpha_{b,i,j} - \alpha_{a,i,j})] * EP_{r,i} \tag{4}$$

3.1.2 Operation time

The time criterion is minimizing the manipulator time, which is required for the complete working cycle realisation. The time between the two operation points *a* and *b* is defined as

$$T_{p2p} = \max_{i=1..n}([\alpha_{b,i} - \alpha_{a,i}] * TP_{r,i}) \tag{5}$$

where $TP_{r,i}$ is the rotation time of the i-th joint needed for the angle of 1°.

Time taken to pass the entire trajectory of one working cycle (with *N* points) is given as

$$T_{tr} = \sum_{j=1}^{N} \max_{i=1..n}[(\alpha_{b,i,j} - \alpha_{a,i,j})] * TP_{r,i} \tag{6}$$

3.1.3 Rotations

This criterion minimises the sum of all rotations of all joints during the operation cycle. The criterion for the movement between points *a* and *b* is defined as

$$A_{p2p} = \sum_{i=1}^{n} (\alpha_{b,i} - \alpha_{a,i}) \tag{7}$$

The sum of the angles in the transition across the trajectory of a single working cycle (with *N* points) is given as

$$A_{tr} = \sum_{j=1}^{N} \sum_{i=1}^{n} (\alpha_{b,i,j} - \alpha_{a,i,j}) \tag{8}$$

3.1.4 Positioning accuracy

This criterion represents the accuracy of the positioning of the robot end effector - the positioning error. It represents the euclidean distance between the desired and the calculated points in the 3-D space. This condition must be considered in each control strategy. The positioning error is defined as

$$D_{tr} = \sum_{j=1}^{N} \sqrt{\begin{matrix}[x_{w,j} - x_{GA,j}]^2 + [y_{w,j} - y_{GA,j}]^2 \\ +[z_{w,j} - z_{GA,j}]^2\end{matrix}} \qquad (9)$$

where *[x_w, y_w , z_w]* are the coordinates of the required points and *[x_{GA}, y_{GA} , z_{GA}]* are points calculated using GA. The minimising of this criterion maximise the robot positioning accuracy.

3.1.5 The used objective functions

The objective functions, which are used in the GA optimisation process, consist of at least two criterions. The first one is always aimed at achieving the defined operating points with the required accuracy (9). The second criterion will determine whether it will be: a) time-optimal (6), b) rotation-optimal (8) or c) energy-optimal (4) design:

a) $FF_{time} = D_{tr} + \beta.T_{tr}$ (10)

b) $FF_{angle} = D_{tr} + \gamma.A_{tr}$ (11)

c)$\mathrm{FF}_{\mathrm{energy}} = \mathrm{D}_{\mathrm{tr}} + \delta.\mathrm{E}_{\mathrm{tr}}$ (12)

where *β=20, γ=15* and *δ=20* are weighting coefficients (the coefficients were set after suboptimal solution investigation). In general, the objective function can contain more criterions. By combining of (10), (11) and (12), we can obtain a universal objective function

d) $FF_{combined} = D_{tr} + \beta_a E_{tr} + \gamma_a T_{tr} + \delta_a A_{tr}$ (13)

In our case the following weights were experimentally set

$FF_{combined} = D_{tr} + 5\ E_{tr} + 10\ T_{tr} + 5\ A_{tr.}$

4. CASE STUDY

The presented approach has been verified using the simulation model of the industrial robot ABB IRB 6400FHD (Fig. 1). The robot is defined by arm lengths R_1 = 0.188 m, R_2 = 1.175 m; R_3 = 1.3m; R_4 = 0.2m; tool: R_{6x} = 0.3; R_{6y} = 0.1 (Fig.2) and by the dynamic parameters listed in Table 1. Using the methodology described above we find the optimal trajectory - operational cycle which is given by ten operational points defined in (14). The robot performs this trajectory as a closed and repeating cycle. The cycle includes the return from P_{10} to P_1 as well.

Table 1. Specification of robot motors

Motor	Axis rotation[°]		Rotation velocity [°/s]	Rated output [W]	Energy consum. [W.s/°]
$E_1(\omega)$	360	-180 ; 180	90	2800	31.1
$E_2(\theta)$	140	-70 ; 70	90	1900	21.1
$E_3(\psi)$	165	-28 ; 105	90	2400	26.6
$E_4(\phi)$	600	-300 ; 300	120	1000	8.3
$E_5(\rho)$	155	-120 ; 120	120	600	5.00
E_6 (ε)	600	-300 ; 300	190	500	2.6

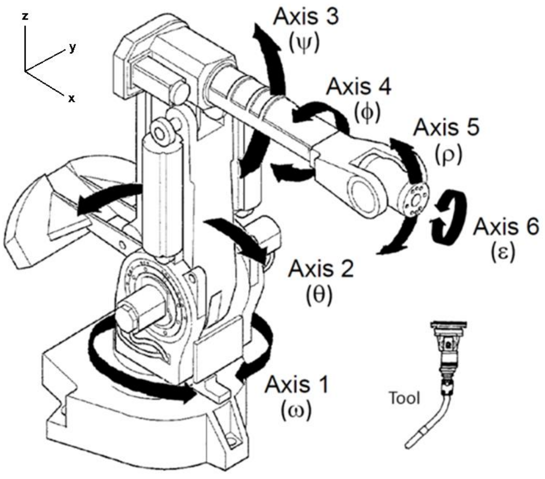

Fig. 1. Robot ABB IRB 6400FHD with 6 degrees of freedom

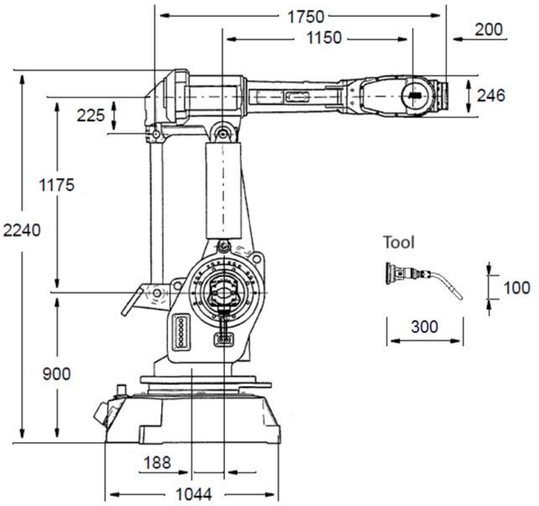

Fig. 2. ABB IRB 6400FHD, size in [mm]

The defined operating points of the robot are

$P_1 = [2.25;1.1;0.25]$
$P_2 = [0.9;1.5;0.25]$
$P_3 = [-0.85;1.14;2.22]$
$P_4 = [-1.8;1.25;1.17]$
$P_5 = [1.8;1.25;1.17]$
$P_6 = [-1.25;-1.1;0.25]$
$P_7 = [-2.25;-1.48;0.25]$
$P_8 = [0.45;-1.14;2.22]$
$P_9 = [0.8;-1.25;2.35]$
$P_{10} = [0.8;-1.25;-1.35].$ (14)

4.1 Direct kinematics

The robot end effector trajectory is described in a clockwise coordinate system using the transformation matrix (TM)

*TM = A*B*C*D*E*F*G*H*I* (15)

where the matrices *A, B, C, D, E, F, G, H, I* are defined in Table 2.

The coordinates *(x,y,z)* of the robot end-effector are given as

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = TM * \begin{bmatrix} R6x \\ 0 \\ R6z \\ 1 \end{bmatrix} \tag{16}$$

4.2 Individual representation

Each individual of the GA population is in form (2) and has 6 x 10 = 60 items (genes) which corresponds to 6 rotation angles for reaching of each working point. The ranges of all genes are defined in column 2 of Tab. 1. Note that a singular configuration cannot appear due to the definition of this task.

Table 2. Direct kinematics – transformation matrixes

Rotation	Movement
$A = \begin{pmatrix} \cos(\omega) & -\sin(\omega) & 0 & 0 \\ \sin(\omega) & \cos(\omega) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$B = \begin{pmatrix} 1 & 0 & 0 & R1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$C = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & R2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$E = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$G = \begin{pmatrix} 1 & 0 & 0 & R3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$H = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\rho) & 0 & \cos(\rho) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$I = \begin{pmatrix} 1 & 0 & 0 & R4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varepsilon) & -\sin(\varepsilon) & 0 \\ 0 & \sin(\varepsilon) & \cos(\varepsilon) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	

4.3 Results and discussion

The selected terminating condition of the GA represents the calculation of the predefined number of generations. Experimentally, the value has been set up to 750,000 generations. The obtained results for the time-optimal control are shown in Tab. 3-4, for the minimum-rotation control in Tab. 5-6, for the energy-optimal control in Tab. 7-8 and for the combined control in Tab. 9-10.

The goal of the time-optimal control is to achieve the shortest time of the operation cycle. The best result achieved was 7.33 seconds (see Tab.11). This solution is inefficient from the point of view of energy or the sum of rotation criterions. The sum of the rotation was 50.12 rad and the power consumption per cycle was 53.32 Ws.

In the case of the minimum-rotation control strategy, the best result achieved was the sum of all rotations amounting to 19.74 rad. This solution is close to the combined-optimal control case (Tab. 10).

Table 3. Joint rotations for the time-optimal control

	ω [rad]	θ [rad]	ψ [rad]	ϕ [rad]	ρ [rad]	ε [rad]
P1	0.345	0.720	-0.153	2.120	0.874	1.620
P2	1.083	0.029	0.603	3.128	0.229	1.147
P3	2.124	-0.142	-0.369	2.939	0.680	-2.327
P4	2.453	0.187	-0.149	2.502	0.264	-1.130
P5	0.398	0.526	-0.475	1.899	1.307	0.695
P6	-2.184	0.110	0.395	4.907	0.919	-4.978
P7	-2.464	0.824	-0.398	3.849	0.518	0.965
P8	-1.256	-0.269	-0.235	3.221	0.679	-1.798
P9	-1.001	-0.015	-0.489	3.385	0.591	-0.946
P10	-0.922	1.222	0.364	3.809	0.309	-0.435

Table 4. Objective functions for time-optimal control

	T_{p2p} [s]	A_{p2p} [rad]	E_{p2p} [Ws]	∑
P1→ P2	0.807	6.596	6.905	14.308
P2→ P3	0.481	5.420	5.806	11.708
P3→ P4	0.663	4.716	5.626	11.004
P4→ P5	0.210	2.427	2.657	5.293
P5→ P6	1.309	5.495	4.617	11.421
P6→ P7	1.643	11.806	13.418	26.867
P7→ P8	0.505	4.647	5.144	10.297
P8→ P9	0.769	4.618	5.260	10.647
P9→ P10	0.162	1.525	1.780	3.467
P10→ P1	0.787	2.875	2.111	5.773
∑	**7.336**	50.123	53.324	110.783

Table 5. Joint rotations for minimum-rotation control

	ω [rad]	θ [rad]	ψ [rad]	ϕ [rad]	ρ [rad]	ε [rad]
P1	0.473	0.592	-0.230	0.130	0.008	-0.617
P2	1.026	0.293	-0.008	0.132	1.155	-0.617
P3	2.086	-0.014	-0.270	2.890	1.155	-0.617
P4	2.523	0.179	-0.270	2.890	-0.440	-0.617
P5	0.597	0.179	-0.270	2.890	-0.440	-0.617
P6	-2.417	0.179	0.434	2.887	-0.665	-0.617
P7	-2.465	0.794	-0.459	1.342	-0.665	-0.617
P8	-1.087	-0.189	-0.462	0.324	-0.665	-0.617
P9	-0.951	-0.100	-0.462	0.130	-0.526	-0.617
P10	-0.966	1.188	0.215	0.130	0.008	-0.617

Table6. Objective functions for minimum-rotation control

	T_{p2p} [s]	A_{p2p} [rad]	E_{p2p} [Ws]	$\sum$
P1→ P2	0.817	1.830	0.963	3.610
P2→ P3	0.488	1.641	1.187	3.316
P3→ P4	1.174	3.238	1.249	5.661
P4→ P5	0.679	1.642	1.295	3.616
P5→ P6	1.093	1.421	0.328	2.842
P6→ P7	1.710	2.913	0.797	5.421
P7→ P8	0.658	2.290	1.224	4.171
P8→ P9	0.782	2.497	1.530	4.809
P9→ P10	0.083	0.412	0.255	0.749
P10→ P1	0.731	1.856	1.840	4.427
$\sum$	8.214	**19.740**	10.669	38.623

Table 7. Joint rotations for the energy-optimal control

	ω [rad]	θ [rad]	ψ [rad]	ϕ [rad]	ρ [rad]	ε [rad]
P1	0.302	0.788	-0.258	-1.032	-0.871	3.198
P2	0.766	0.246	0.437	-1.411	-0.956	3.198
P3	1.883	-0.065	-0.489	-1.411	-0.959	3.198
P4	2.318	0.415	-0.424	-1.410	-1.016	3.198
P5	0.689	0.415	-0.181	0.371	-1.044	3.198
P6	-2.312	0.415	0.741	0.371	-1.044	3.198
P7	-2.545	0.746	-0.351	0.370	0.010	3.198
P8	-1.194	-0.222	-0.489	-0.053	0.010	3.198
P9	-1.011	-0.116	-0.489	-0.205	0.010	3.198
P10	-1.011	0.788	0.575	-0.210	0.010	3.198

Table 8. Objective functions for energy-optimal control

	T_{p2p} [s]	A_{p2p} [rad]	E_{p2p} [Ws]	$\sum$
P1→ P2	0.836	3.849	1.115	5.800
P2→ P3	0.443	2.165	0.906	3.514
P3→ P4	0.711	2.359	0.681	3.751
P4→ P5	0.306	1.036	0.608	1.951
P5→ P6	1.037	3.681	0.761	5.479
P6→ P7	1.910	3.923	0.669	6.502
P7→ P8	0.695	2.709	1.232	4.636
P8→ P9	0.860	2.879	1.336	5.075
P9→ P10	0.117	0.441	0.175	0.733
P10→ P1	0.677	1.973	1.127	3.777
$\sum$	7.592	25.015	**8.609**	41.217

Table 9. Joint rotations for the combined control

	ω [rad]	θ [rad]	ψ [rad]	ϕ [rad]	ρ [rad]	ε [rad]
P1	0.459	0.590	-0.123	-2.397	-0.014	-0.588
P2	0.957	0.027	0.482	-2.397	-0.432	-0.588
P3	2.118	-0.180	-0.489	-2.397	-0.432	-0.588
P4	2.473	0.179	-0.221	-2.397	-0.432	-0.588
P5	0.547	0.179	-0.221	-2.395	-0.431	-0.588
P6	-2.356	0.179	0.661	-2.037	0.128	-0.588
P7	-2.519	0.756	-0.347	-2.037	0.128	-0.588
P8	-1.109	-0.286	-0.347	-2.037	0.128	-0.588
P9	-0.974	-0.186	-0.347	-2.037	-0.014	-0.588
P10	-0.971	0.590	0.661	-2.037	-0.014	-0.588

Table 10. Objective functions for the combined control

	T_{p2p} [s]	A_{p2p} [rad]	E_{p2p} [Ws]	$\sum$
P1→ P2	0.910	2.574	0.581	4.066
P2→ P3	0.385	2.084	1.267	3.736
P3→ P4	0.739	2.338	0.724	3.801
P4→ P5	0.229	0.982	0.586	1.797
P5→ P6	1.226	1.929	0.390	3.545
P6→ P7	1.848	4.702	1.325	7.875
P7→ P8	0.642	1.749	0.996	3.387
P8→ P9	0.898	2.452	1.594	4.944
P9→ P10	0.086	0.378	0.261	0.724
P10→ P1	0.642	1.786	1.213	3.641
$\sum$	**7.604**	**20.974**	**8.938**	37.517

The energy-optimal task has achieved 8.6089 Ws. In this case, it is possible to see the strong correlation between the energy and the time-optimal trajectory. The energy-optimal control is relatively fast (only about 3.5% slower than the fastest trajectory) and also relatively efficient in terms of sum of rotations (26.7% increase in comparison to the rotation-optimal solution).

The last case was aimed at finding a combined-control strategy. It includes all the components representing accuracy, operation time, sum of rotations and energy consumption. This solution is fast (only about 3.7% slower than the fastest trajectory), it obtains a good value of rotations (6.3% higher than the rotation-optimal case) and a low power consumption (3.8% higher in comparison with the energy-optimal solution).

The evolution of the objective functions of all optimization cases is depicted in Fig. 3, Fig.4 and Fig.5.

Table 11. Comparison of all objective functions

Objective function	T_{p2p} *[s]*	A_{p2p} *[rad]*	E_{p2p} *[Ws]*
Time	**7.3357**	50.1233	53.3236
Rotation	8.2136	**19.7401**	10.6689
Energy	7.5924	25.0154	**8.6089**
Combined	**7.6044**	**20.9742**	**8.9381**

In Fig. 5 the comparison of the evolution of the positioning accuracy in each optimization strategy is compared.

5. CONCLUSION

In the proposed robotic arm trajectory optimization, various design aspects have been considered, like operation-time minimization, robot rotation minimization, energy consumption minimization and combined optimization. Because the inverse kinematics task for a robot with many degrees of freedom is a complex problem, the genetic algorithm approach in combination with the robot simulation has been proposed. The results obtained have shown that such a design is able to provide very good results and it can achieve significant time savings, energy and wear and tear on the equipment.

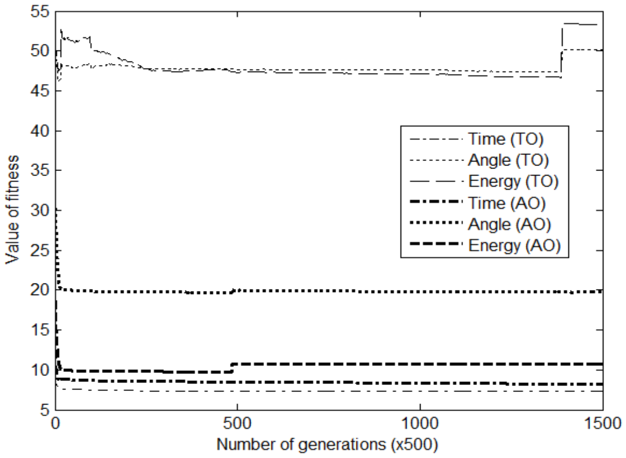

Fig. 3. Graphs of the objective function evolution for the time-optimal (TO) and rotation-optimal (AO) solution

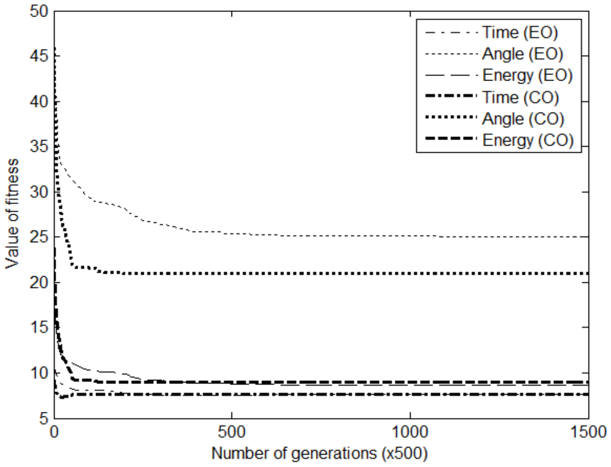

Fig. 4. Graphs of the objective function evolution for the energy-optimal (EO) and combined-optimal (CO) solution

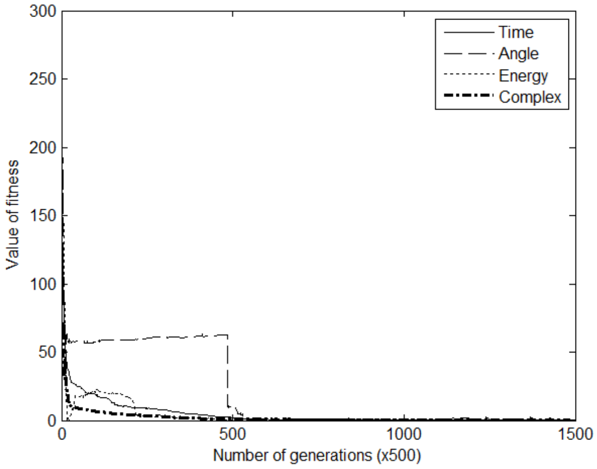

Fig. 5. Graphs of the positioning accuracy evolution of all optimization strategies.

ACKNOWLEDGEMENT

This project was supported by the Slovak research and grant agency, grant No. VEGA 1/0178/13 and VEGA 1/2256/12.

REFERENCES

Davidor, Y. (1991). Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization. *World Scientific*

Eiben, A.E., Smith, J.E. (2007). *Introduction to evolutionary computing*. Springer.

Garg, D.P., Kumar, M. (2002). Optimization techniques applied to multiple manipulators for path planning and torque minimization. *Engineering Applications of Artificial Intelligence 15*, Elsevier, 241–252

Goldberg, D.E. *(1989). Genetic algorithms in search, optimisation and machine learning.* Addison-Wesley.

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

Pac M., Rakotondrabe M., Khadraoui S., Popa D., Lutz P. (2013). Guaranteed Manipulator Precision via Interval Analysis of Inverse Kinematics. In *IDETC/CIE 2013* August 4-7, 2013, Portland, Oregon, USA

Park, J.H., Kim, H.S., Choy, Y.K. (1999). Optimal Trajectory Control for Robot Manipulators using Evolution Strategy and Fuzzy Logic. In *ICASE: Institute of Control, Automation and System Engineering*, Korea, June, 1999

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

Rana, A. S., Zalzala, A. (1997). Collision-free motion planning of multi-arm robots using evolutionary algorithms. In *Proceedings of the Institution of Mechanical Engineers* Part I, 211, 373–384.

Sekaj, I. (2011). Control algorithm design based on evolutionary algorithms. In: Chugo, D., Yokota, S. (ed.), *Introduction to Modern Robotics*. iC. Press, Hong Kong. iConcept Press Ltd., 2011. - ISBN 978-0980733068. - S. 251-266

Virgala I., Frankovský P., Kelemen M. (2013). Mathematical model of a planar manipulator. In: *Strojárstvo*. Strojárstvo Extra (In Slovak). - ISSN 1335-2938. - Vol. 17, NO. 5 (2013), pp. 140-144.