BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LX No. 3A/2008	131 - 142	Seria Tehnică

Robot Optimization by Means of a Hybrid Genetic Algorithm

Sevasti Mitsi, Konstantin-D. Bouzakis, Dimitrios Sagris, Gabriel Mansour

Laboratory for Machine Tools and Manufacturing Engineering, Mechanical Engineering Department, Aristoteles University of Thessaloniki, 54124, Greece

Abstract

In this paper some problems of robot task optimization based on hybrid genetic algorithm are presented. The main optimization criteria are the total travel time, the avoidance of singular configurations, the path smoothness, the collision avoidance, as well as the joint angles limits. The optimization problem is solved through a hybrid method that combines a genetic algorithm, a quasi-Newton algorithm and a constraints handling method, using a multi-objective function and various constraints. The feasibility of the proposed algorithm is verified in three applications of the robots used in design or manufacturing procedures.

Key words: *Robot manipulator, Path planning, Hybrid genetic algorithm, Optimization method*

Introduction

The productivity of a given manipulator used into a workcell, mainly depends on the cycle time of the end-effector. The cycle time is affected of many parameters, such as the placement of the robot base relative to the task [1, 2], the maximum velocities and accelerations of the actuators, the configurations of the robot on the path that obtains a collision free movement, etc. Several methods have been used in order to optimize different parameters of robot with respect to the task, considering various criteria [3, 4, 5, 6].

In the present paper a hybrid optimization method is developed to solve some robot tasks. The optimized performance index includes the positioning accuracy, the travel time, the collision avoidance, as well as the avoidance of singular configurations. Furthermore the smoothness of the path and the normal distribution of the intermediate poses are taken into account. The developed method is demonstrated to six, two or five degrees of freedom manipulators, in some numerical examples, where different tasks are involved.

Mathematical Formulation

In the present paper the manipulator is considered as an open space chain with revolute joints. A reference frame P_i attached at each link i is considered. The relative position between two successive frames is described using the 4x4 homogeneous transformation matrices and the Denavit-Hartenberg parameters [7].

For the task points it is very important to avoid the singular configurations of the robot. This can be assured by the maximization of the robot manipulability [8]. Also the cycle time of the robot must be minimized to improve the productivity of the robot. Other criteria take into account the collision avoidance, the limits of joint angles etc.

Therefore, it could be formulated as an optimization problem, where the objective function (F) takes into account the deviations between the prescribed and the calculated end-effector poses (F_1), the distance between the robot parts and the obstacles (F_2) for collision avoidance, the total travel time among all poses successively (F_3), the normal distribution of travel time for all intermediate motions (F_4) and the manipulability measure w_k for the "n" poses (F_5):

$$F = F_1 + \alpha \cdot F_2 + \beta \cdot F_3 + \gamma \cdot F_4 + \delta \cdot F_5 \tag{1}$$

with

$$F_1 = \sum_{k=1}^{n} \left(\sqrt{Dx_k^{\,2} + Dy_k^{\,2} + Dz_k^{\,2}} \right)$$

$$F_2 = \sum_{k=1}^{n} \sum_{p=1}^{p^*} \sum_{e=1}^{e^*} \frac{1}{\left(P_i - P_e\right)_{\min}} + PFV$$

$$F_3 = \sum_{k=0}^{n} t_k^{2}$$

$$F_4 = \sum_{k=0}^{n} \left(t_k - t_{aver}\right)^2$$

$$F_5 = \sum_{k=1}^{n} \left(1 / w_k^2 \right)$$

where Dx_k, Dy_k, Dz_k are the coordinate deviations of pose "k" between initial interpolated path and the calculated one, $(P_i - P_e)_{\min}$ is the minimum distance between robot part "i" and obstacle "e", PFV is a Penalty Function Value activated when the minimum distance is lower than the collision limit, t_k is the travel time between poses k and k+1 and t_{aver} is the average travel time for the total path. The weighting factors α, β, γ and δ are used in order to scale the contribution of the corresponding terms in the objective function value. The minimization of the objective function determines the optimum values of the unknown parameters. During the optimization procedure the imposed constraints regarding the unknown variables are described by:

$$x_{\ell\,\min} < x_\ell < x_{\ell\,\max}, \quad \ell = 1,2,\ldots m \tag{2}$$

where m is the number of the variables and $x_{\ell\,\min}$ and $x_{\ell\,\max}$ are the lower and upper limits of the variable ℓ. These constraints take into account the limits of the joint variables imposed through the robot design and the geometry of the robot workcell.

Proposed Algorithm

The optimization problem is solved with a hybrid method that combines a Constraints Handling Method (CHM), a Genetic Algorithm (GA) and a Quasi-Newton Algorithm (QNA). The flow chart of the proposed algorithm is illustrated in Fig. 1. The input data are the joints type and number, the variables bounds and the algorithm parameters. In these parameters are included the initial parameters of the GA such as the population size, the crossover rate, the mutation rate, etc. and the number of the GA and QNA loops. Using the equation (1) the fitness function is defined, which is used in all steps of the algorithm.

The Constraints Handling Method reduces the variables bounds about the optimum value of each variable regarding the previous step solution, using a user-defined percentage, in order to

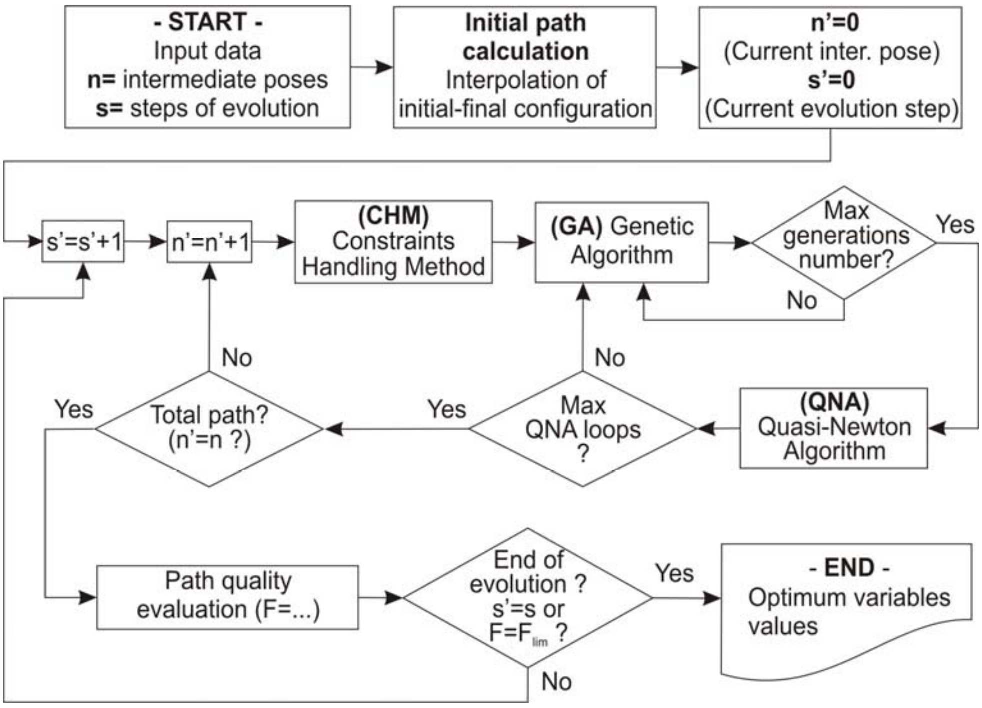


Fig. 1. Flowchart diagram of the developed algorithm

accelerate the optimum search of GA and QNA that follow. In any case the global limits of each variable (Equation 2) are preserved in order to work in acceptable range of joint angles.

During the genetic algorithm, starting populations are randomly generated to set variables values, which are used to calculate the fitness function value. Genetic algorithm [9] uses selection, elitism, crossover and mutation procedures to create new generations. The new generations converges towards a minimum that is not necessarily the global one. After some repetitions when the maximum generations' number is achieved, the variables values corresponding to the minimum fitness function value are selected as the optimum variables values of the genetic algorithm.

The optimum GA variables values are inserted in the QNA [10] as an initial variables vector guess. The quasi-Newton algorithm modifies the values of this vector using a finite-difference gradient method until a maximum iterations number or a local minimum is reached. Through this 'hill climbing' method a new fitness function value is obtained. The loop of QNA is applied several predefined times, including the repetition of GA loop, in order to locate several local minimums using the GA and approach the global one using the QNA loop. When the maximum loops number of QNA is achieved, the variables values corresponding to the minimum fitness function value are selected as the optimum QNA variables values. The end of the loops is achieved when the quality of the objective function reaches a predefined limit or a predefined amount of steps.

Especially in the case where the proposed methodology is applied to the path planning optimization, an initial step is introduced (see Fig.1). This initial step is the interpolation of "n" intermediate configurations between the initial configuration and the final one. This not optimized and not necessarily free of collisions path is used as the current path for the next evolutionary steps of the algorithm. Each step of the evolution procedure (s') uses a combination of CHM, GA and QNA for each intermediate configuration (n') of the manipulator. When all the intermediate poses of the path are calculated (n'=n), the obtained path is defined as the current one and the next evolution step is activated. The evolution of this step uses as current path the new obtained.

Applications

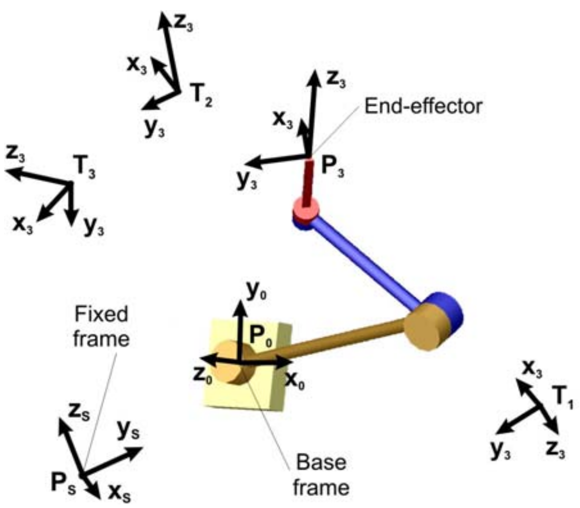
The introduced methodology is applied on several manipulators, in three cases. The first case is the determination of robot design parameters, as well as the robot base placement of a spatial RR manipulator [1, 11, 12]. Other application is the base placement and the configurations determination of a 5-DOF robot, taking into account only the positioning accuracy [1, 3]. Furthermore, the optimization method is applied on a 6-DOF manipulator for the optimum collision free path planning involving additionally the manipulability measure and the travel time of the end-effector [1,13].

Geometric design of a RR spatial robot

In the present application the geometric design parameters, the base placement and the joint angles of a 2-DOF spatial robot with 2 revolute joints (RR) are determined, when some end-effector poses are prescribed. The magnitudes to be determined are the base placement with respect to a reference system, the geometry of the two parts, as well as the geometry and the fastening angle of the tool to the second part. The variables that describe the problem are the Denavit-Hartenberg parameters θ_i, α_i, a_i and d_i (i=0,1,2), regarding the base placement and the 1st and 2nd link, as well as the parameters θ_3 and d_3 for the tool placement. For each prescribed tool pose, different joint angles values of θ_1 and θ_2 are used. Thus for one prescribed tool pose there are 14 unknown parameters, for two prescribed tool poses the unknown parameters become 16 and for three prescribed tool poses 18.

The input data used for the algorithm are the variables bounds, the end-effector poses (Fig. 2) and the algorithm parameters (Table 1). The initial applied variables limits are: $0<\theta_i<360^o$ (i=0,1,2,3), $0<\alpha_i<360^o$ (i=0,1,2), $0<a_0<1000$, $0<a_i<100$ (i=1,2), $0<d_0<1000$ and $0<d_i<100$ (i=1,2,3). The optimization index in this basic approach includes only the positioning accuracy (F_1), which means that the other parts of equation (1) have no contribution ($\alpha=\beta=\gamma=\delta=0$).

Three numerical applications corresponding to one, two and three target points are presented.



$$A^3_{S\,pr\,1} = \begin{bmatrix} 0.6562 & -0.1129 & 0.7460 & 162.0367 \\ -0.2188 & -0.9747 & 0.0449 & 82.1840 \\ 0.7221 & -0.1927 & -0.6643 & 48.7429 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A^3_{S\,pr\,2} = \begin{bmatrix} 0.1439 & 0.9289 & 0.3410 & 72.9645 \\ -0.9894 & 0.1398 & 0.0365 & 42.7749 \\ -0.0137 & -0.3427 & 0.9393 & 113.3997 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A^3_{S\,pr\,3} = \begin{bmatrix} 0.7645 & 0.6408 & -0.0689 & 43.2729 \\ -0.4770 & 0.4907 & -0.7291 & 16.1409 \\ -0.4334 & 0.5903 & 0.6809 & 88.6821 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. Prescribed end-effector poses

Table 1. Four methods parameters and the corresponding results

Exa-mple	Pre-scribed tool poses	Algorithm	Parameters				Results	
			Number of loops			Reduction of variables range	Computational time	Fitness value
			GAs (L_1)	QNAs (L_2)	CHMs (L_3)			
1st	One	Only GA	2.50E+05	-	-	-	0:06:52	6.36E+00
		GA and CHM	10,000	-	24	25%	0:07:48	3.36E-01
		GA and QNA	500	500	-	-	0:10:08	1.80E-06
		Proposed (1)	100	500	4	85%	0:13:04	**1.20E-09**
		Proposed (2)	10	10	1	85%	**0:00:03**	4.73E-06
2nd	Two	Only GA	5.00E+05	-	-	-	0:18:06	1.42E+02
		GA and CHM	20,000	-	24	25%	0:18:15	3.70E+00
		GA and QNA	500	1,000	-	-	0:34:31	6.00E-02
		Proposed (1)	100	500	9	65%	0:36:13	**4.33E-06**
		Proposed (2)	100	100	4	65%	**0:02:48**	2.66E-02
3rd	Three	Only GA	1.00E+06	-	-	-	0:41:53	3.77E+02
		GA and CHM	40,000	-	24	25%	0:42:18	1.95E+01
		GA and QNA	500	2,000	-	-	1:48:35	5.60E-01
		Proposed (1)	100	500	19	30%	2:44:29	**4.22E-03**
		Proposed (2)	100	200	9	55%	**0:22:06**	1.20E-01

The three prescribed poses of the tool frame (T_1, T_2, T_3) with respect to a fixed Cartesian coordinate system P_S are graphically presented in Fig. 2. Using these poses the matrices $A^3_{S\ pr\ k}$ (k=1,2,3) of the prescribed end-effector poses are evaluated (right part of Fig. 2).

In order to make obvious the accuracy advantage of the proposed method, four different algorithms were tested in each numerical example. The first one uses only GA, the second combines the GA with the CHM, the third one uses a combination of GA with the QNA and the fourth is the proposed one. The proposed method is tested using two approaches of finalizing the algorithm, based on the objective function value (1) or the computational time (2). The parameters involved in all tests, mainly in GA procedure, are the same and selected as optimums through many applied tests: population of individuals=50, cross probability=70% and mutation probability=8%. All the other algorithm parameters involved in the problem are different in each case and are presented in Table 1. The loops number of GA, QNA and CHM are selected in a way that the total generations number in four tests are equal, in order to be comparable.

Fig. 3 illustrates the value of the fitness function in logarithmic scale versus the generations' number of the first three tested methods (Table 1) and of the proposed (1) one based on the fitness function criterion, when three poses are prescribed.

As shown in this figure, the performance of the proposed algorithm is substantially better than that of the three other methods during the whole procedure both in accuracy and computational time. The obtained value of the fitness function illustrates clearly the advantage of the proposed algorithm. The optimum variables values obtained with the proposed algorithm (1), taking into account the fitness function criterion, are presented in Table 2. In each column are inserted the obtained variables values for the corresponding numerical example, according the base positioning, the geometry of each link as well as the configuration of each prescribed pose.

The comparison between the elements of the matrices $A^3_{S\ r\ i}$ and the corresponding elements of the prescribed ones $A^3_{S\ pr\ i}$ shows that the maximum positional deviation is lower than 0.0008 mm in the three numerical examples. The maximum deviation of orientations is lower than

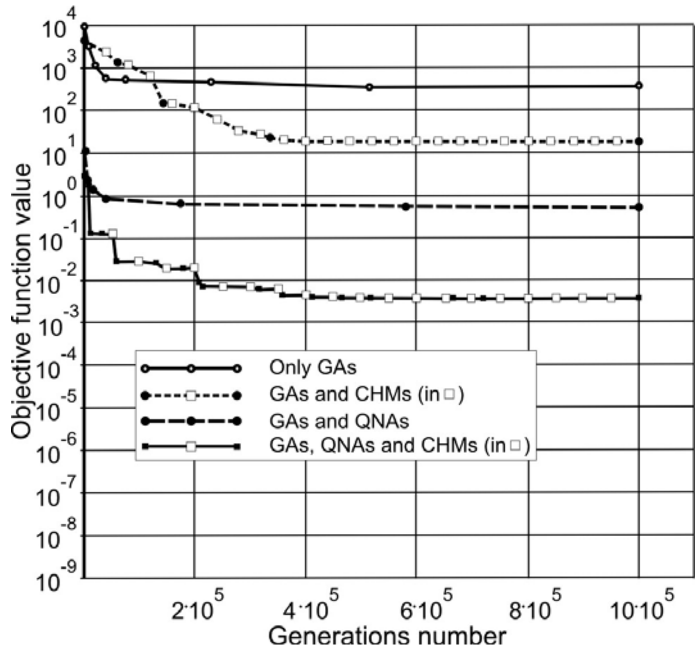


Fig. 3. The minimum objective function value for three prescribed end-effector poses

0.0009 rad (0.052 degrees) in first and second example, which is insignificant value, and lower than 0.0318 rad (1.822 degrees), which is acceptable value.

The second test of the proposed algorithm is used to obtain an acceptable fitness function value, with criterion the lower computational time. It is observed that the proposed algorithm leads much faster to a lower value of the fitness function in comparison with the other three methods.

Table 2. Optimum values of the variables of RR robot

Solutions	Variables			Results of the examples		
				1st	2nd	3rd
Base position	1	θ_0	(°)	0.0000	11.4268	36.8875
	2	α_0	(°)	237.6337	295.7161	50.0390
	3	a_0	(mm)	95.7807	63.8321	37.9261
	4	d_0	(mm)	69.1681	59.8139	6.5254
1st part geometry	5	α_1	(°)	80.8300	135.2362	215.5565
	6	a_1	(mm)	43.8419	40.8134	94.2007
	7	d_1	(mm)	65.7063	48.9870	44.3800
2nd part geometry	8	α_2	(°)	94.5122	83.0846	246.4630
	9	a_2	(mm)	4.0268	34.8337	40.0234
	10	d_2	(mm)	55.8065	10.4731	5.0121
3rd part geometry	11	θ_3	(°)	140.4193	43.4457	45.4301
	12	d_3	(mm)	5.4104	65.4491	27.7802
1st robot configuration	13	θ_1	(°)	26.7873	23.4496	5.2311
	14	θ_2	(°)	114.3732	64.9991	323.4674
2nd robot configuration	15	θ_1	(°)	-	74.9070	36.3000
	16	θ_2	(°)	-	228.3855	222.6390
3rd robot configuration	17	θ_1	(°)	-	-	67.1911
	18	θ_2	(°)	-	-	176.7161

Robot base placement of a 5-DOF manipulator

The introduced methodology is applied on a manipulator with five revolute joints (Fig. 4) used in manufacturing processes for the determination of the optimum base placement, when an amount of end-effector poses are prescribed. In the Cartesian space, the manipulator has five degrees of freedom. Three of them are defined from the coordinates x, y, z of the end-effector with respect to the coordinate system of the manipulator base (P_0). The other two are the orientation angles α and β of the end-effector ($\alpha=\pi/2+\theta_2+\theta_3+\theta_4$ and $\beta=\theta_5$). The optimization problem is focused on the determination of the base placement and the joint angles values for the "n" prescribed end-effector poses. These magnitudes are described by the D-H parameters θ_0, α_0, a_0 and d_0 for the base placement and by the joint angles variables θ_i (i=1,…,5) for each prescribed tool pose. Joint angles θ_i (i=1,…,5) have a different value for each prescribed tool pose, while all other 4 parameters are constant. Thus for one prescribed tool pose there are 8 unknown parameters and for two prescribed tool poses the number of the parameters becomes 12.

The input data used for the algorithm are the links dimensions, the variables bounds [3] and the end-effector poses (see Fig. 4). The optimization criterion is only the positioning accuracy of the end-effector (weighting factors α=β=γ=δ=0).

Four numerical applications corresponding to one, two, three and five target points are tested. The five prescribed poses (T_1,…,T_5) of the tool frame are graphically represented in Fig. 4. In table of this figure are given the coordinates (x, y, z) of the origin and the orientation angles (α and β) of the tool frame with respect to the fixed Cartesian coordinate system P_S, for these four applied cases. For each numerical application the acceptable level of fitness value is considered lower than 10^{-3} in the optimization problem solving.

The algorithm parameters involved in all tests are the same and selected as optimums for this problem, through many applied tests. All the other algorithm parameters such as the number of GAs, QNAs and CHMs with the respective reduction percentage of variables bounds are different in each numerical example (see Table 3).

Prescribed poses	Origin coordinates			Orientation angles	
	x (mm)	y (mm)	z (mm)	α (°)	β (°)
1st - T_1	0.0	598.4	185.4	-50.0	0.0
2nd - T_2	329.0	569.9	660.1	20.0	150.0
3rd - T_3	-131.8	228.3	19.9	-90.0	100.0
4th - T_4	122.6	377.2	8.8	-90.0	90.0
5th - T_5	-80.3	377.6	12.7	-90.0	90.0

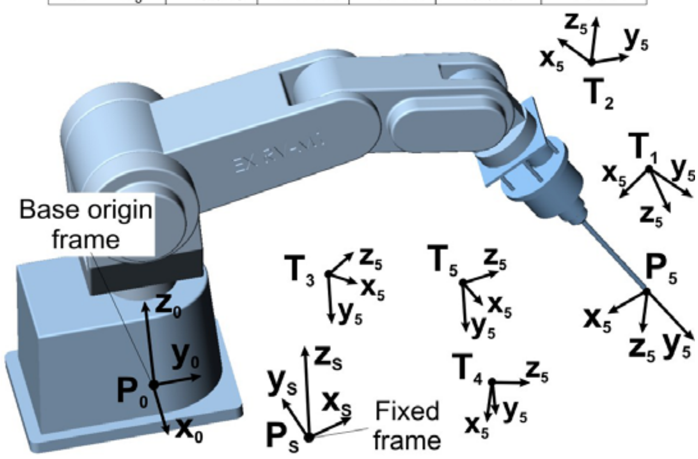


Fig. 4. Prescribed values of the end-effector poses

Table 3. Algorithm parameters and the corresponding results

Exa-mple	Prescribed tool poses	Parameters				Results	
		Number of loops			Reduction of variables range	Computational time	Fitness value
		GAs (S1)	QNAs (S2)	CHMs (S3)			
1	1 – T_1	10	30	4	85%	0:00:04	1.44E-05
2	2 – T_1,T_2	10	30	4	85%	0:00:20	6.10E-05
3	3 – T_1,T_2,T_3	30	100	4	85%	0:01:12	5.80E-04
4	5 – T_1,…,T_5	30	100	14	30%	0:04:00	7.00E-04

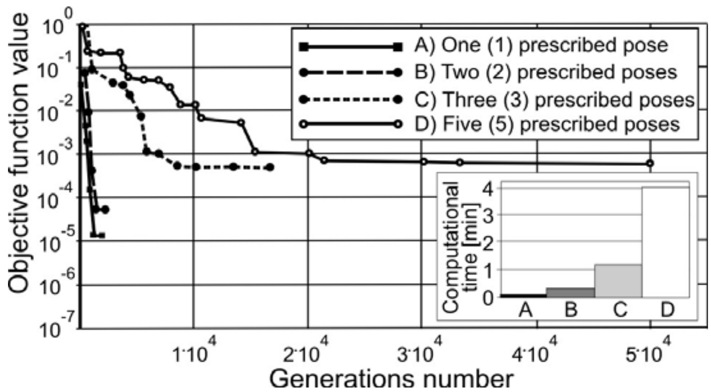


Fig. 5. The minimum objective function value versus generations number

The algorithm parameters, the obtained computational time and the corresponding fitness function value for the four numerical examples are inserted in Table 3. Fig. 5 illustrates the value of the fitness function in logarithmic scale versus the generations number of the four examples. For the numerical examples of one, two and three prescribed poses the algorithm converges to a lower value of fitness function in a quite short time. The solution of the fourth example corresponding to five prescribed poses reaches the prescribed value with a higher computational time. The optimum variables values deal to 0.0014 mm maximum positional deviation and 0.0223 rad (1.28 degrees) maximum deviation of orientations.

Path planning of a 6-DOF manipulator

The third application of the proposed algorithm is the optimization of collision free path, between an initial and a final robot configuration, using an amount of intermediate calculated configurations [13]. The manipulator in this case is known and already placed. The variables that describe the optimization problem are the joint angles θ_i (i=1,..,6) for the "n" intermediate poses. In a case of two intermediate poses the calculated variables are 12, for four intermediate poses the calculated variables are 24, and so on.

The input data for the algorithm are the links dimensions (Fig.6), the joint angles variables and their initial bounds (Table 4), the maximum allowed joint rotational speed and the prescribed robot configurations. Furthermore, the amount of intermediate calculated configurations, as well as the maximum allowed steps of evolution is parameter of the algorithm too. The graphical representations of initial and final robot configurations, as well as the initial interpolated path with the collision of robot parts and obstacles are presented in Fig. 6.

The representations in this figure are solid models in order to be comprehensible and clearly shown. The data used internal the proposed algorithm are only the point clouds of parts and obstacles. The point clouds are generated by means of a developed algorithm that uses the 3D graphical models to apply points on the surfaces of each item. These point clouds are the data for a collision detection procedure, applied during the path evaluation by means of the objective

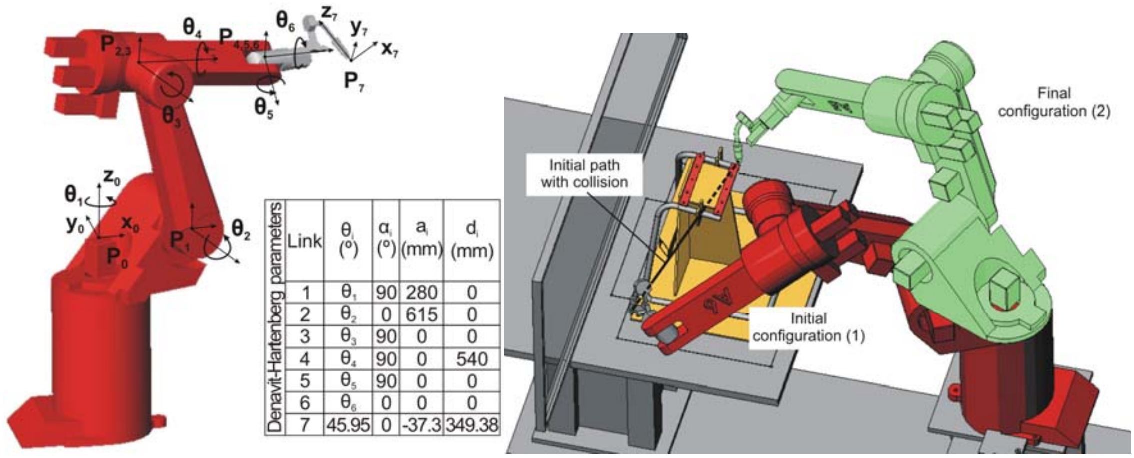


Fig. 6 6-DOF robot, Denavit-Hartenberg parameters and the initial (1) and final (2) robot configurations

Table 4. Initial variables bounds and maximum allowed joint speeds

Variables	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 (°)
Min	-180	-30	-60	-270	-300	-180
Max	150	135	210	90	-54	180
Max allowed speed	140 °/s	140 °/s	140 °/s	270 °/s	270 °/s	400 °/s

function. Only the results of the algorithm are presented using 3D models. Many numerical applications were conducted using the proposed method, trying several different data according the prescribed poses, the intermediate poses amount, the objective function forms, the weighting factors of objective function and so on.

In the frame of this application the presented results are based on a case of two prescribed poses for a chair frame welding (see Fig. 6), using standard weighting factors as results of many try and error approaches and the objective function form as presented in equation (1). The results with respect to the amount of intermediate calculated configurations are presented in Table 5. The efficiency of the proposed method is composed by the stable and high values of the manipulability measure for all tests, by the acceptable computational time for an off-line optimization method and by the minimum normal distribution of the travel time deviations. The increase of travel time for more intermediate calculated poses is due to the fact that the intermediate movements are linear and much more difficult to be optimized using the same parameters.

The parameters involved in all tests, mainly in GA procedure, are the same and selected as

Table 5. Results with respect to the amount of intermediate poses

Exa-mple	Intermediate poses	Travel time (sec)	Time normal distribution deviations %		Computational time (h:min:sec)	W	
			Average	Max		Average	Max
1	4	0.605	2.02	3.27	0:05:51	0.3	0.35
2	6	0.705	1.98	5.95	0:12:24	0.31	0.36
3	8	0.791	2.43	7.4	0:23:35	0.32	0.36
4	10	0.904	1.88	4.8	0:35:58	0.33	0.37
5	20	1.134	2.25	5.65	2:31:43	0.33	0.38

Weighting factors: $\alpha = 0.01$, $\beta = 20$, $\gamma = 1000$, $\delta = 0.01$

Table 6. Prescribed and intermediate configurations for four intermediate poses

Configuration / Joint angles	*Initial configuration*	Intermediate configurations 1st	2nd	3rd	4th	*Final configuration*
θ_1	*27*	32	20	3	-13	*-23*
θ_2	*30*	41	58	76	89	*75*
θ_3	*26*	30	30	22	5	*-8*
θ_4	*-69*	-35	-2	15	35	*67*
θ_5	*-113*	-96	-117	-104	-100	*-115*
θ_6	*-60*	-22	5	22	42	*78*

optimums through many applied tests. The reduced variables range during the CHM of optimization procedure is 0.25 rad (~15^o). The loops number of GA is 10 and the loops number of QNA is 3 and are selected in a way that the solutions are accurate and quick enough simultaneously.

The example of four intermediate poses is described in detail, in order to make obvious the efficiency of the proposed algorithm. The optimum variables values for the four intermediate

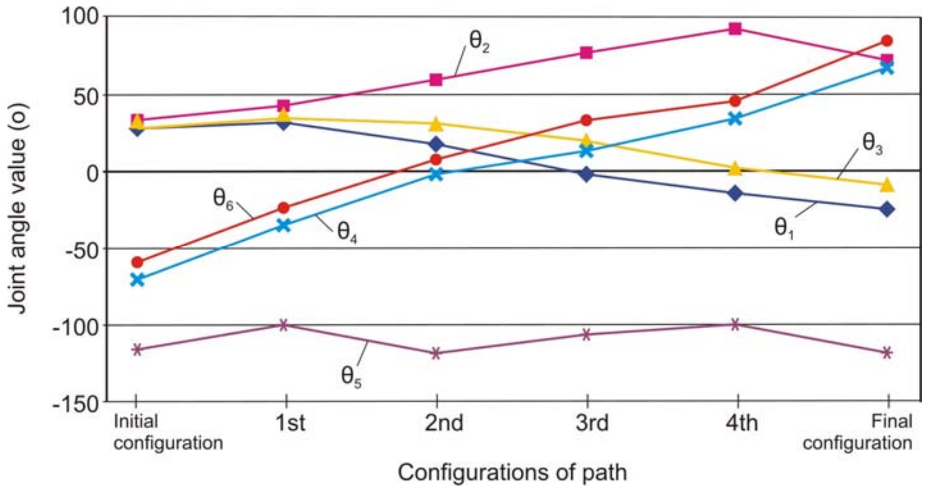


Fig. 7 Joint angles values for the intermediate four poses, as well as the initial and final configurations

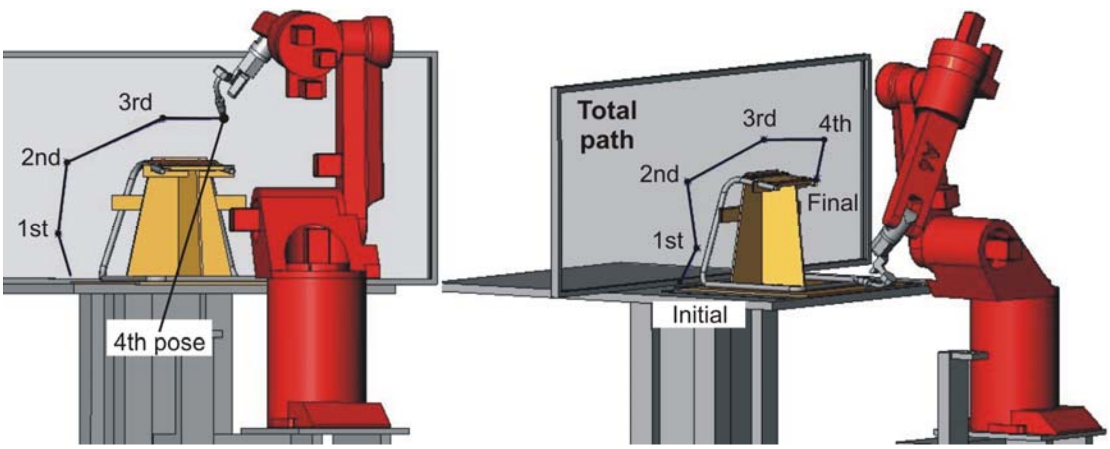


Fig. 8 Fourth intermediate configuration and the total path of the end-effector

poses, obtained with the proposed method are presented in Table 6. Furthermore the joint angles values for all the intermediate, as well as for the initial and final prescribed configurations, are graphically illustrated in Fig. 7. It is obvious that for all the joint angles values the transition from the initial configuration to the final one is smooth. The fourth configuration of the corresponding path of these results is graphically illustrated in Fig. 8, as well as the total path.

Conclusions

In the present paper a hybrid optimization method is developed to determine optimum solutions in problems of robotics. The proposed methodology is applied in research areas, such as the geometric design of spatial manipulators, the base placement of a robot in combination with the inverse kinematics problem and the path planning of a already placed robot. The optimized performance index includes the travel time on the proposed path, the obstacles avoidance, as well as the avoidance of singular configurations, combined or individually. Furthermore the smoothness of the path and the normal distribution of the intermediate poses are taken into account, in the case of path planning. The robot links and obstacles point clouds representations are retrieved by means of given 3D models using automated procedures. Numerical examples for two, five and six degrees of freedom manipulators regarding the mentioned problems, demonstrate the efficiency of the developed method.

The developed algorithms are written in Fortran and the solid models are developed in SolidWorks environment. The computational time refers to a Pentium 4 PC using 1.6 GHz CPU. Both algorithms and graphics can be modified to agree with any manipulator or problem conditions.

References

1. Sagris D., *Placement and movements' optimization of industrial manipulators, by means of a hybrid method*, Doctoral thesis, Zitis Publications, Thessaloniki, 2008.
2. Mitsi S., Bouzakis K.-D., Sagris D., Mansour G., *Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm*, Robotics and Computer-Integrated Manufacturing, Elsevier, Vol.24, (2008), pp.50-59.
3. Sagris D., Mitsi S., Bouzakis K.-D., Mansour G., *5-DOF robot base location optimization using a hybrid algorithm*, Mecatronica, vol.1, pp.76-81, 2004.
4. Meystel A., Guez A., Hillel G., *Minimum time path planning for robot motion in obstacle strewn environment*, ACM Annual Computer Science Conference, ACM Press, pp.367-376, 1986.
5. Zhang W., Sobh T. M., *Obstacle avoidance for manipulators*, Systems Analysis Modelling Simulation, vol.43, no.6, pp.749-757, 2003.
6. Mitsi S., Bouzakis K.-D., Sagris D., Mansour G., *Optimum collision free robot path planning using a hybrid genetic algorithm*, Bulletin of the Polytechnic Institute of Jassy, Vol. LIII, Fasc. 5, pp.369-376, 2007.
7. Denavit J., Hartenberg R. S., *A kinematic notation for lower pair mechanisms based on matrices*, Transactions of the ASME, Journal of Applied Mechanics, vol.E22, pp.215-222, 1955.
8. Yoshikawa T., *Manipulability of robotic mechanisms*, International Journal of Robotics Research, vol.4, no.2, pp.3-9, 1985.
9. Coley D., An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific Press, 1999.
10. IMSL, Fortran subroutines for mathematical applications, Visual Numerics, (1997).
11. Sagris D., Mitsi S., Bouzakis K.-D., Mansour G., *Geometric design optimization of spatial RR robot manipulator using a hybrid algorithm*, Acta Technica Napocensis, Series: Applied Mathematics and Mechanics, Romania, Vol.47, No.2, pp.717-722, 2004.

12. Mavroidis C., Lee E., Alam M., *A New Polynomial Solution to the Geometric Design Problem of the Spatial R-R Robot Manipulators Using the Denavit-Hartenberg Parameters*, Transactions of the ASME, Journal of Mechanical Design, Vol. 123, pp.58-67, 2001.
13. Mitsi S., Bouzakis K.-D., Sagris D., Mansour G., *Robot path planning optimization, free of collisions, using a hybrid algorithm*, 3rd International Conference on Manufacturing Engineering, Kalithea-Chalkidiki, Greece, pp.475-484, 2008.

Optimizarea robotilor industriali cu ajutorul unui algoritm hibrid genetic

Rezumat

In cadrul acestui articol se prezinta unele probleme de optimizare ale robotilor industriali folosind un algoritm hibrid genetic. Criteriile principale de optimizare sunt durata ciclului de lucru, ocolirea pozitiilor singulare, evitarea obstacolelor, cat si domeniul de lucru ale articulatiilor robotului. Problema de optimizare este rezolvata cu ajutorul unei metode hibride care combina un algoritm genetic, un algoritm quasi-Newton si o metoda de manuire a constrangerilor, folosind o functie multi-obiectiva si diverse constrangeri. Validitatea algoritmului propus este verificata in trei aplicatii ale robotilor privind proiectarea sau utilizarea lor intr-o celula de fabricatie.