ABSTRACT
In this paper an algorithm for the optimum collision-free path planning of a spatial robot, using multiple optimization criteria is developed, in the frame of a total inquire of optimum serial robots manipulation. The interface platform is an offline control system that exploits graphical data of the robot parts and workspace obstacles in order to import them in the optimization algorithm. The main optimization criteria are the total travel time from an initial known to a final known configuration through a user-defined amount of intermediate calculated poses, the avoidance of singular configurations and the path smoothness, taking into account the collision avoidance of all robot parts and obstacles, 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 using an offline control system of a 6-DOF manipulator.

KEYWORDS: Robot manipulator, Path planning, Genetic algorithm, Optimization method

1. INTRODUCTION
The present paper constitutes a part of a doctoral thesis regarding problems optimization in fields of robotics, based on a hybrid method. The proposed algorithm combines a genetic algorithm (GA) with a hill climbing method (quasi-Newton algorithm - QNA) and furthermore a constraints handling method (CHM) is involved. The developed algorithm uses the above-mentioned methods in order to avoid the disadvantages and exploit the advantages of each individual method. GA applied alone has the advantage of searching the whole space of solutions as well as not being entrapped in a local minimum. Furthermore GA is efficient only for limited number of variables and weakness signs are detected when the number of variables or the total search space rises. QNA, being a gradient-based search method, has the advantage of detecting local minimums for higher number of variables, but is strongly depending on the initial variables vector. The synthesis of these methods combines their advantages and detects local minimums in the whole search area. Finally CHM is applied in order to reduce the variables limits, reduce the space to be searched and accelerate the whole procedure. The proposed algorithm is very efficient in finding the optimal solution in a reduced computation time.

In order to test the efficiency of the developed method, it is applied on several fields of robotics. The determination of optimum robot base position and joint angles, considering discrete end-effector positions of spatial 6-DOF /1/ and 5-DOF /2/ manipulators with revolute joints, is approached using this method. Furthermore it is applied to robot design in order to determine the optimum robot geometry, robot base position and joint angles of a 2-DOF spatial RR manipulator /3/. The fitness function that quantifies the problem in each case consists of the sum of the deviations squares between the prescribed poses and the real poses of the end-effector, the
manipulability measure of each configuration, as well as the total cycle time of the process. Furthermore the workspace restricts and the variables limits are taken into account. Numerical examples according all the aforementioned problems, demonstrate the efficiency of the developed method in comparison with other optimization methods.

Furthermore, an off line system for manipulation of robots is developed that includes optimization procedures through an application that combines a user form with a 3D graphic model in Solidworks environment /4/. Some of the available tools for the robot programming are the setting and reading the end-effector pose through the direct and inverse kinematics problem, the graphical collision avoidance, the path simulation and the NC code automatic generation. This off line system includes the aforementioned developed optimization algorithms like robot base placement, path planning and point clouds generation. The proposed system can be used further for optimization of the cycle duration of the process.

The problem of optimal path planning for a given manipulator in the presence of obstacles has been the subject of considerable research. The end-effector path has to be determined between two specified poses, avoiding collision with workspace obstacles and optimizing a performance index. 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 the travel time on a free of obstacles path, in two or three dimensions, considering various criteria /5, 6, 7/. In other cases, the path planning methods focus on the exploitation of the redundant degrees of freedom, using heuristic approaches, such as genetic algorithms or neural networks /8, 9, 10, 11, 12/.

In the present paper a hybrid optimization method is developed to determine an amount of intermediate configurations as node poses of an end-effector path of given initial and final configurations. The optimized performance index includes the travel time on the proposed path, the avoidance of workspace obstacles, 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. All parts and obstacles point clouds representations are retrieved by means of given 3D models using automated procedures. Following previous experience of an optimization algorithm /1, 3/, a formulated approach focused on path optimizing is proposed, where each intermediate configuration interacts with the others through the travel time magnitude and the path smoothness. The developed method is applied, through an off line system /4/, to six degrees of freedom manipulator, in some numerical examples, where four up to twenty intermediate poses are involved.

2. MATHEMATICAL FORMULATION

In the present paper the manipulator is considered as an open space chain with six revolute joints (Figure 1). A reference frame $P_i$ attached at each link $i$ (i=0,1...6) and at tool $P_7$ are considered. The relative position between two successive frames is described using the 4x4 homogeneous transformation matrices and the Denavit-Hartenberg parameters /13/. In the table of figure 1 are inserted the corresponding D-H parameters, the joint angles limits and the maximum allowed angular velocities.

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 /14/. The measure of the manipulability is defined as:

$$w = \sqrt{|\det(J^TJ)|}$$

where J is the Jacobian matrix. For a non-redundant robot manipulator, the measure $w$ is simplified to $w = |\det J|$. The total travel time $t$, required to visit the “n” intermediate poses, starting from the initial known configuration and ending at the final known configuration, is given by:
Figure 1: Denavit-Hartenberg coordinate systems, parameters and the graphical model for a 6-DOF robot.

\[ t_1 = \sum_{k=0}^{n} t_k \]  

where \( t_k \) is the travel time between pose \( k \) and pose \( k+1 \) and \( "n" \) is the number of the intermediate poses, \( t_0 \) is the travel time between initial known pose and pose 1 and \( t_n \) is the travel time between pose "n" and the final known pose. The time \( t_k \) can be written as:

\[ t_k = \max\left( \frac{|q_{i,k+1} - q_{i,k}|}{q_{i,\text{max}}} \right), (i = 1,2,...,6) \]  

where \( q_{i,k} \) is the \( i^{th} \) joint value for \( k^{th} \) end-effector pose and \( q_{i,\text{max}} \) is the maximum allowed angular velocity of joint \( i \), imposed by the constructor. The slowest joint determines the motion time between poses \( k \) and \( k+1 \). The same equation is used to determine the travel time from and to the known configurations of starting and ending pose. Equations (2) and (3) denote that the manipulator configuration affects significantly the travel time of the end-effector.

Therefore, the optimal path planning could be formulated as an optimization problem, where the objective function \( F \) takes into account the deviations between the initial linear interpolated path and the calculated end-effector poses \( F_1 \), the distance between the robot parts and the obstacles \( F_2 \), 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" intermediate poses \( F_5 \):

\[ F = F_1 + \alpha \cdot F_2 + \beta \cdot F_3 + \gamma \cdot F_4 + \delta \cdot F_5 \]  

with \( F_1 = \sum_{k=1}^{n-1} \sqrt{\Delta x_k^2 + \Delta y_k^2 + \Delta z_k^2} \), \( F_2 = \sum_{k=1}^{n} \sum_{l=1}^{n} \left( \frac{1}{P_{l,l+1}} \right) \), \( F_3 = \sum_{k=0}^{n} t_k \), \( F_4 = \sum_{k=0}^{n} (t_k - t_{\text{aver}})^2 \), \( F_5 = \frac{1}{\sum_{k=1}^{n} \left( \frac{1}{w_k^2} \right)} \)
where $D_x, D_y, D_z$ are the coordinate deviations of pose “k” between initial interpolated path and the calculated one, $(P_i - P_j)$ 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 $r_{i,e}$, $t_k$ 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 $\alpha, \beta, \gamma$ and $\delta$ 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, namely the joint variables of the configurations for the intermediate poses. During the optimization procedure the imposed constraints regarding the unknown variables are described by:

$$x_{\text{min}} < x_{i} < x_{\text{max}}, \quad i = 1, 2, \ldots, m$$

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

### 3. COLLISION DETECTION PROCEDURE

The proposed algorithm includes some initiative steps, which are required in order to optimize the path. The pre-available graphical representation of the robot and the obstacles are exploited to create point clouds only on the surface of each object. These point clouds as data are easily used in a repetitive procedure, unlike the 3D graphical representations. The amount of points in each point cloud is determined parametrically and is chosen according to the collision probability risk and the geometrical complexity of each part. Additional data of each part is the circumscribed sphere centered on the gravity center of the part and a radius defined by the most remote point. The initial graphical robot representation and the assembled point clouds of all robot parts, which are obtained by means of a point clouds generation procedure, are presented in figure 2.

The point clouds model of robot links and obstacles are used in a collision detection procedure, applied for each calculated intermediate pose in two ways (Figure 3). During the path quality evaluation of a proposed intermediate configuration, which is described afterwards in the optimization procedure, a collision detection procedure is required. In the first step no collision is detected when the minimum distance $D_{\text{min},i,e}$ between the sphere centre of the robot part (i)
and the sphere centre of the obstacle \((e)\) is greater than the sum of the radius of these spheres \((R_i + R_e)\). In this case the collision check of the specific couple is stopped and there is no Penalty Function Value contribution for this couple \((i,e)\). In other case, the second step is activated and a point by point collision check is required, using the point clouds analysis. The procedure activates a double loop to detect the distances for all points of the robot link \((i)\) and obstacle \((e)\). When the minimum of these distances is smaller than the collision limit then a collision is established and a huge PFV is contributed. When all couples of robot parts and obstacles are checked and no collision is detected \((PFV=0)\), then the current robot configuration is accepted and further tested for its quality through the objective function. In any other case \((PFV\neq 0)\) the configuration is unacceptable and the set of a huge PFV leads to a definitely rejected solution.

The collision limit \((r_{i,e})\) is a user-defined value of the acceptable distance between a robot link \((i)\) and an obstacle \((e)\). However these values are constrained due to the fact that the point clouds have finite amount of points and the risk of a not-detected collision appears. In order to determine the minimum safe value of the collision limit, a procedure is applied, based only on the point clouds of robot links and obstacles. This algorithm calculates the maximum points distance in the points cloud for each part \((r_i)\) and each obstacle \((r_e)\). The worst placement of the lengths \(r_i\) and \(r_e\) is perpendicular and the hypotenuse determines a safe distance, which is used as the collision limit \((r_{i,e})\) for the couple of robot part \((i)\) and obstacle \((e)\).
4. PROPOSED ALGORITHM

The optimization problem is the determination of "n" intermediate collision-free robot configurations, through "s" steps of evolution, optimized with respect to the objective function, for a robot end-effector path among obstacles for known initial and final configurations. The optimization problem is solved with a hybrid method, which is base on the aforementioned hybrid method of the doctoral thesis. The present method 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 figure 4. The input data for the algorithm are the number of intermediate poses and the required steps of evolution, which is defined by the user in order to balance the quality of the result and the computational time. Furthermore input data are the links dimensions, 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 (4) the fitness function is defined, which is used in all steps of the algorithm to evaluate the path quality.

The first step of the proposed algorithm 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.

The Constraints Handling Method reduces the variables bounds about the values of each configuration of the current path, using a user-defined percentage, in order to accelerate the optimum search of GA and QNA that follow. In any case the global limits of each variable (Equation 5) 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 /15/ uses selection, elitism, crossover and mutation procedures to create new generations. The new genera-

![Figure 4: Flowchart diagram of the developed algorithm.](image-url)
tions 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 /16/ 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 loops of CHM, GA and QNA are applied until all the points of the current path are replaced. For the total path the quality is evaluated using the objective function at the end of each evolution step. Each next evolution step uses as path the optimum of previous evolution step. The end of evolution is achieved when the quality of the objective function reaches a predefined limit or a predefined amount of evolution steps. The final intermediate configurations are the proposed path of the end-effector.

5. NUMERICAL APPLICATION

The introduced methodology is applied on a manipulator with six degrees of freedom and six revolute joints used for welding of office furniture frames. The input data for the algorithm are the links dimensions (Figure 1), the joint angles variables and their initial bounds, the maximum allowed joint rotational speed (Table of figure 1) 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 figure 5.

![Figure 5: Initial (1) and final (2) robot configuration of the numerical application, including the end-effector collision for the simple interpolated path.](image-url)
The representations in this figure are solid models in order to be comprehensible and clearly shown. The proposed algorithm uses internally as data only the point clouds of parts and obstacles. Only the results of the algorithm are presented using 3D models. Several 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 the present paper the presented results are based on a case of two prescribed poses for a chair frame welding (see figure 5), using standard weighting factors as result of many try and error approaches and the objective function form as presented in equation (4). 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 the mutation probability = 8%. The reduced variables range during the CHM of optimization procedure is 0.25 rad (~15°). 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 results with respect to the amount of intermediate calculated configurations are presented in table 1. The efficiency of the proposed method is validated by the stable and high values of the manipulability measure /1, 2/ 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 observed due to the fact that the intermediate movements are linear and a path variation appears leading to useless movements. More results are presented for the case of four intermediate poses.

Table 1: Results with respect to the amount of intermediate poses.

<table>
<thead>
<tr>
<th>Example</th>
<th>Intermediate poses</th>
<th>Travel time (sec)</th>
<th>Time normal distribution deviations %</th>
<th>Computational time (h:min:sec)</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Max</td>
<td>Average</td>
<td>Max</td>
<td>Average</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0.605</td>
<td>2.02</td>
<td>3.27</td>
<td>0:05:51</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.705</td>
<td>1.98</td>
<td>5.95</td>
<td>0:12:24</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>0.791</td>
<td>2.43</td>
<td>7.4</td>
<td>0:23:35</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>0.904</td>
<td>1.88</td>
<td>4.8</td>
<td>0:35:58</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1.134</td>
<td>2.25</td>
<td>5.65</td>
<td>2:31:43</td>
</tr>
</tbody>
</table>

Weighting factors: α = 0.01, β = 20, γ = 1000, δ = 0.01

The example of four intermediate poses is described in detail, in order to make obvious the efficiency of the proposed algorithm. The joint angles values for the four intermediate poses, as well as for the initial and final prescribed configurations, are graphically illustrated in figure 6, and the optimum variables values are presented in the table of the same figure. It is obvious that for all the joint angles values the transition from the initial configuration to the final one is smooth.

The corresponding configurations of these results are graphically illustrated in figure 7 for the four intermediate poses. Furthermore, the total path is illustrated in the same figure, presenting the manipulator in the reference position.

6. CONCLUSIONS

In the present paper a hybrid optimization method is developed to determine an amount of intermediate configurations as node poses of an end-effector path of given initial and final configurations. The optimized performance index includes the travel time on the proposed path, the obstacles 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 robot links and obstacles point clouds representations are retrieved by means of
Figure 6: Joint angles values for the intermediate four poses, as well as the initial and final configurations.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Initial config.</th>
<th>Intermediate configuration 1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>Final config.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>27</td>
<td>32</td>
<td>20</td>
<td>3</td>
<td>-13</td>
<td>-23</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>30</td>
<td>41</td>
<td>58</td>
<td>76</td>
<td>89</td>
<td>75</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>26</td>
<td>30</td>
<td>30</td>
<td>22</td>
<td>5</td>
<td>-8</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>-69</td>
<td>-35</td>
<td>-2</td>
<td>15</td>
<td>35</td>
<td>67</td>
</tr>
<tr>
<td>$\theta_5$</td>
<td>-113</td>
<td>-96</td>
<td>-117</td>
<td>-104</td>
<td>-100</td>
<td>-115</td>
</tr>
<tr>
<td>$\theta_6$</td>
<td>-60</td>
<td>-22</td>
<td>5</td>
<td>22</td>
<td>42</td>
<td>78</td>
</tr>
</tbody>
</table>

Figure 7: Four intermediate configurations and the total path of the end-effector.

given 3D models using automated procedures. Numerical examples for six degrees of freedom manipulator, where four up to twenty intermediate poses are applied, demonstrate the efficiency of the developed method. The usage of nodal poses is the initiative step for an evolution of the
proposed method. These nodal joint angles values will be the key points of a polynomial curve that describes the angle value for the total movement. The polynomial curve determined through the objective function evaluation obtains a smooth path, free of obstacles and optimized according some performance index.

The developed algorithms are written in Fortran and the solid models are developed in SolidWorks environment. Both algorithms and graphics can be modified to agree with any manipulator or problem conditions.

7. REFERENCES