Robotics and Computer-Integrated Manufacturing 26 (2010) 162–173

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Optimal location of a general position and orientation end-effector's path relative to manipulator's base, considering velocity performance

Athanasios Nektarios *, Nikos A. Aspragathos

Mechanical Engineering and Aeronautics Department, University of Patras, Patras, Greece

ARTICLE INFO

Article history:
Received 3 October 2007
Received in revised form
8 May 2009
Accepted 1 July 2009

Keywords:
Path following
Orientation
Trajectory planning
Optimization
Workcell design

ABSTRACT

The objective of the present paper is to introduce an offline algorithm searching for the optimal or suboptimal placement of a robot's base during workcell design, so that its end-effector can perform a position and orientation path following task of a given 3D curved path and orientation, maximizing the manipulator's velocity performance. The global index employed for this velocity performance optimization is the approximation of the minimum manipulator velocity ratio (AMMVR).

The AMMVR is computed from a finite set of MVR values computed at specific locations along the path selected by a trajectory approximation algorithm. This algorithm approximates the path using an interpolation method based on the cubic spline interpolation to guarantee velocity continuity.

The optimization is performed by a simple genetic algorithm, using as fitness function the AMMVR and the manipulator's constraints. The algorithm's effectiveness is demonstrated by simulation tests using a six-degree-of-freedom non-redundant Puma type manipulator and the obtained results are presented and discussed.

1. Introduction

One of the most common problems in designing a robotic workcell is the optimal placement of the manipulator relative to the performed task. Usually the placement of the task in the robot workspace is determined by trial-and-error methods in a robot simulator environment, based on the designer's intuition and expertise. However, this practice is time consuming. Moreover, it is relatively difficult to take into account design criteria such as minimized cycle time, reduced wear of the robotic parts, reduced energy consumption and joint velocities limits. Therefore, a technique that could provide suboptimal or optimal solutions for the manipulator's placement based on these criteria is highly desired. The manipulator capabilities play an important role to achieve the optimal solution based on these criteria. The force and velocity capabilities depend on the manipulator's configuration and therefore on the placement of the manipulator's base relative to the performed task. When the manipulator is extended, its end-effector can move fast, but it cannot apply high forces, whereas when it is retracted, it can apply high forces, but it cannot move fast.

Various indices and measures were proposed to describe the kinematic ability of a robotic manipulator and most of them were derived from the definition of the manipulability measure (MM) proposed by Yoshikawa [1]. Dubey and Luh [2] introduced the manipulator velocity ratio (MVR), expressing the transmission ratio of the end-effector velocity to the velocity of the joints. This measure describes not only the kinematic ability of the manipulator at the specific configuration, but it also takes into account the direction of the end-effector's motion.

Salisbury and Craig [3] used the condition number to give an indication of the shape of the manipulability ellipsoid, providing a criterion for the uniformity of performance to any direction. Ermolov and Podurajev [4] proposed hybrid indices, considering the kinematic, dynamic and stiffness properties of a manipulator. The hybrid manipulator measures are suitable for cases where a combination of velocity and force performance is specified.

The MMs were applied in a variety of tasks from assembly to path following in order to select the best location of the task in the manipulator's workspace. Nelson et al. [5] used the MM as a criterion for locating assembly tasks in the manipulator's workspace. Martin et al. [6] proposed an integral criterion to be employed in path planning. This criterion depends on the joint configuration and velocities and provides a family of locally optimal solutions that are not homotopically equivalent. The advantages of using an integral cost criterion compared to instantaneous ones were explained and the risks of not reaching an optimal solution were highlighted. Similarly, Uchiyama et al. [7] integrates the determinant of the Jacobian along the path to be followed. Since the kinematic performance can be described using the MM, integrating along the path provides the kinematic

* Corresponding author.
E-mail address: t.nektarios@kinematik.gr (A. Nektarios).

0736-5845/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2009.07.003

behavior of the manipulator for completing the path following task.

Furthermore, optimization algorithms were developed using the aforementioned measures as objective functions. Nelson and Donath [8] introduced an offline algorithm for locating a robotic task using the MM. The optimization is performed by a modified steepest descent method, which maximizes the robot's manipulability performance, while avoiding singularities and remaining inside its work envelope. Recently, Sobh and Toundykov [9] developed an optimizing algorithm to automatically design manipulators taking into account their kinematic abilities. The task that the manipulator is called to perform is the basis for synthesizing the manipulator's geometrical parameters. The measure used for this optimization is again the MM.

Aspragathos and Foussias [10] proposed a method for determining the optimal location of a linear path following task in the robotic workspace using a 3DOF manipulator, considering the velocity performance and efficiency. After comparing various measures, the minimum MVR along the path, as global measure is formulated to represent the velocity performance along the path. The MVR was computed at specific points along the path, selected through Taylor's algorithm [11]. The minimum of these MVR values is selected, which characterizes an approximation of the worst performance along a path. The main idea is that maximizing the formulated measure, the manipulator performs better, even in the worst case. The approximate minimum MVR (AMMVR) measure is maximized through an optimization performed by a simple genetic algorithm.

Smooth and continuous position and orientation path applications, such as welding, painting, laser or water jet cutting, require a global measure that takes into account the smoothness of the path, the orientation of the tool and other constraints for the determination of the best placement of the task into the robots workspace. The present paper introduces an offline algorithm that determines the suboptimal or optimal placement of the manipulator relative to a curved end-effector path so that a position and orientation path following task can be completed with the lowest possible joint velocity performance, by extending Aspragathos and Foussias [10] work to any shape of the path and to six dimensions taking into account the end-effector orientation. The problem is now more complicated as the path is described by a general parametric curve instead of a straight line segment. Moreover, the end-effector must follow the position, as well as the orientation along the path. The specific interpolation technique offers smooth motion and velocity continuity, making this algorithm useful for various path following applications such as welding, painting and laser or water jet cutting, which require full control of the position and orientation of the end-effector, while taking also into consideration the velocity performance. In the optimal location, the cycle time can be reduced since the end-effector velocity could reach its maximum by driving the joint velocities to their maximum limits.

The main advantage of the proposed approach is that it combines multiple criteria and constraints compatible with the aforementioned applications, where smooth curved paths should be followed with the maximum velocity to reduce the cycle time. Moreover, it takes into account these constraints in conjunction with a global optimization criterion based on the ratio of the end-effector versus the joint velocity for the determination of the best placement of the robot relative to the task. Therefore, any robot control algorithm applied will take full advantage of the joint actuators velocity capabilities.

2. Problem statement

This paper is focused on finding the optimal placement of a manipulator relatively to the path following task considered for the early stages of workcell design. The introduced approach could be applied to any non-redundant manipulator and be used for applications requiring smooth end-effector motion along a given path while maximizing its velocity performance. The optimal placement of the manipulator relatively to the task is a prerequisite for any later control algorithm implementation to fully exploit the actuators velocity capabilities. In case that the placement is not the most suitable, a control algorithm cannot achieve the most of the actuators velocity capabilities, resulting to poor end-effector velocity performance.

The problem can be generally stated as follows. Determine the optimal relative position and orientation between the robot's base and the desired path following task, so that the robot's end-effector moves along the path with the highest possible velocity using the most efficient joint velocities. Placing the manipulator at the optimal location the task is performed using the minimum possible joint velocities for a given end-effector velocity. Therefore, the motors velocities could be fully utilized to reduce the task's cycle time and increase productivity. Furthermore, the wear of the robot's mechanical parts and its energy consumption are minimized.

Fig. 1 presents the desired path that must be followed (solid line) and the path that the robot's end-effector actually follows (dashed line). The position and orientation of the desired path to be followed are defined by the following parametric form:

$$\underline{X}_d(s) = \begin{bmatrix} x_d(s) & y_d(s) & z_d(s) & \phi_d(s) & \theta_d(s) & \psi_d(s) \end{bmatrix}^T \tag{1}$$

where $x_d(s)$, $y_d(s)$, $z_d(s)$ are the path's position coordinates relative to $\{U\}$, the reference frame, and $\phi_d(s)$, $\theta_d(s)$, $\psi_d(s)$ are the respective orientation coordinates (Euler Angles), with s being the length of the path.

However, the actual robot's end-effector path is determined by an interpolation algorithm applicable for paths of any continuous shape, which is defined by a similar form

$$\underline{X}_r(s) = \begin{bmatrix} x_r(s) & y_r(s) & z_r(s) & \phi_r(s) & \theta_r(s) & \psi_r(s) \end{bmatrix}^T \tag{2}$$

The frames $\{F_s\}$, $\{F_i\}$ and $\{F_f\}$ represent the starting, the intermediate and the final nodes, referring to the robot tool frame, with $i = 1, 2, \ldots, k-2$ with k the number of all the frames along the path. The nodes play an important role for the optimization, since the measure that is compatible with the optimization specifications is computed at the nodes.

Concerning the relative position and orientation of the manipulator and the task, there are two options: either to keep the manipulator's base locked in the workspace and find the position and orientation of the task or the other way around. For this paper the second option was chosen (Fig. 2).

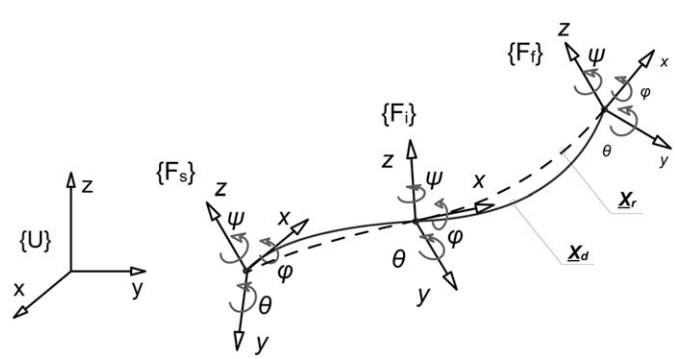

Fig. 1. Desired path approximated by the robot's end-effector.

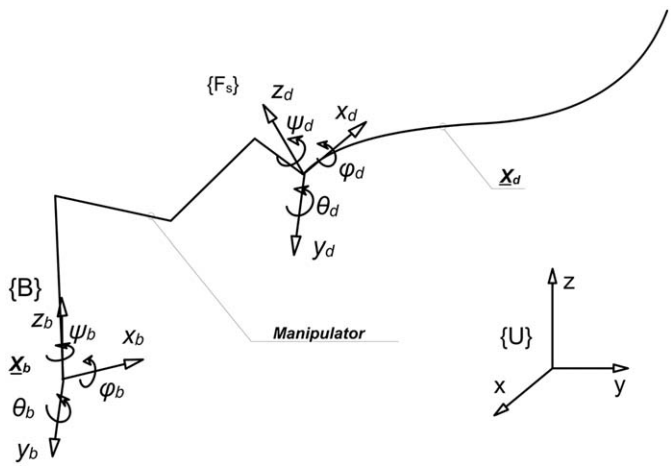

Fig. 2. Placing the manipulator's base so that the end-effector starting coordinates coincide with the starting node coordinates of the desired path.

The path to be followed is defined by its starting node frame $\{F_s\}$, whose position and orientation is given with respect to the reference frame $\{U\}$. Assuming that the end-effector coincides with the frame $\{F_s\}$, the position and orientation of the manipulator's base $\underline{X}_b$ can be computed by the kinematic equations for the given initial joint coordinates $\underline{q}(s_s)$. Considering $\underline{q}(s_s)$ as the optimization variable, when the optimization algorithm computes the optimal $\underline{q}(s_s)$, the relative position and orientation of the manipulator's base $\underline{X}_b$ referred to the frame $\{U\}$, is determined.

The problem is defined as placing the manipulator at the optimal location $\underline{X}_b$, so that the end-effector's path $\underline{X}_r(s)$ can approximate with the desired accuracy the path to be followed defined by $\underline{X}_d(s)$, while using the most efficient joint velocities $\dot{\underline{q}}(s)$ to achieve the highest end-effector velocity $\dot{\underline{X}}_r(s)$.

2.1. The manipulability velocity ratio

Aspragathos and Foussias [10] compared four indices along a path: the minimum MVR, the average MVR, the minimum MM and the average MM. In this comparison, the relation of each index along the path with the maximum joint velocity was examined. The minimum MVR was proved experimentally to be the only one that is highly correlated with the maximum joint velocity. Thus the minimum MVR measure is selected as the most compatible measure with the specifications of the problem examined in this paper and it is used as the objective function for the optimal location search algorithm, introduced in Section 4.

According to Dubey and Luh [2], the MVR is defined as the ratio of the end-effector velocity norm to the joint velocity norm and is described by the following equation:

$$r_v = \sqrt{\frac{\dot{\underline{X}}_v^T \dot{\underline{X}}_v}{\dot{\underline{q}}_v^T \dot{\underline{q}}_v}} \tag{3}$$

where $\dot{\underline{X}}_v$ and $\dot{\underline{q}}_v$ are the weighted vectors of the end-effector velocity and of the joint velocities, respectively. The weighted matrices are used in order to restore the physical meaning, since the velocity vector elements might not all have the same physical units.

Assuming that $\dot{\underline{X}}$, $\dot{\underline{q}}$ and J are the end-effector velocity vector, the joint velocities and the robot's Jacobian matrix, respectively, the transformations $J_v = W_x^{1/2} \cdot J \cdot W_q^{-1/2}$, $\dot{\underline{X}}_v = W_x^{1/2} \cdot \dot{\underline{X}}$ and $\dot{\underline{q}}_v = W_q^{1/2} \cdot \dot{\underline{q}}$ are used, where $W_x \in R^{mxm}$, $W_q \in R^{nxn}$ are symmetric and positive definite weighting matrices, which in most instances are considered as diagonal ones. The joint velocity vector is given by

$$\dot{\underline{q}}_v = J_v^{-1} \dot{\underline{X}}_v \tag{4}$$

and setting $\underline{u}_v$ the unit vector in the direction of $\dot{\underline{X}}_v$

$$\underline{u}_v = \frac{\dot{\underline{X}}_v}{\sqrt{\dot{\underline{X}}_v^T \dot{\underline{X}}_v}} \tag{5}$$

So, the MVR is given by

$$r_v = \frac{1}{\sqrt{\underline{u}_v^T (J_v J_v^T)^{-1} \underline{u}_v}} \tag{6}$$

MVR expresses the relationship between the end-effector's and the joints' velocities. Contrary to the MM which depends only on the manipulator configuration, the MVR depends on both the configuration and the direction of the end-effector velocity vector.

Intuitively, a high MVR value can be interpreted as a relatively high end-effector output velocity compared to the joints' input velocities. On the other hand, a low value of MVR is a requirement of relatively high joint velocity to succeed a high end-effector velocity. Therefore, the logic behind examining the minimum MVR is that it describes the worst velocity ratio along the path. By maximizing the worst velocity ratio, the MVR is kept at a high level along the path. These are the main reason for choosing the minimum MVR as the global index for the optimal location search algorithm in comparison with similar methods for smooth motion applications that use the MM or its variations.

2.2. The approximation of the minimum MVR along the path

Normally, the MVR measures should be calculated on every point of the path to analyze the kinematic ability of a manipulator's end-effector along the desired path, which would require too much computational power. To overcome this obstacle, Aspragathos and Foussias [10] approximated the path by selecting points on the path using the nodes defined by the Taylor's interpolation algorithm [11]. This approximation can be implemented assuming that the near points to the selected ones for the MVR value computation have a very small deviation from that value.

This way the worst kinematic abilities throughout a path can be characterized by the approximate minimum value of MVR, which can be expressed as

$$\text{AMMVR} = \min[(r_v)_i, (r_v)_s, (r_v)_f] \tag{7}$$

with $i = 1, 2, \ldots, k-2$ the intermediate nodes and k the total number of nodes, where the MVR is calculated.

2.3. Interpolation method

The optimal solution considering the velocity performance and efficiency can be expressed by the maximum of AMMVR. To compute the AMMVR a trajectory approximation algorithm is used to determine the path nodes where the MVR is calculated.

The bounded position deviation algorithm (Aspragathos [12]) based on cubic spline interpolation is used for the trajectory approximation. The main advantage of the cubic spline interpolation compared with the linear one is the satisfaction of the velocity's continuity condition. In this paper the algorithm

introduced by Aspragathos [12] is extended to interpolate the end-effector orientation in addition to the position. The cubic spline interpolation is applied in the joint space.

$$\underline{q}(s) = \underline{q}_A + \underline{q}_B s + \underline{q}_C s^2 + \underline{q}_D s^3 \text{ with } s_s \le s \le s_f \tag{8}$$

with

$$\underline{q}_A = \underline{q}(s_s), \underline{q}_B = \dot{\underline{q}}(s_s)$$

$$\underline{q}_C = \left(\frac{3}{s_f^2}\right)(\underline{q}(s_f) - \underline{q}(s_s)) - \left(\frac{1}{s_f}\right)(\dot{\underline{q}}(s_f) + 2\dot{\underline{q}}(s_s))$$

$$\underline{q}_D = \left(-\frac{2}{s_f^3}\right)(\underline{q}(s_f) - \underline{q}(s_s)) + \left(\frac{1}{s_f^2}\right)(\dot{\underline{q}}(s_f) + \dot{\underline{q}}(s_s))$$

where the starting and final values of the joint angles and their velocities can be found by solving the inverse kinematic problem.

The joint velocities between two successive nodes are computed by the following relationship, obtained by differentiating Eq. (8).

$$\dot{\underline{q}}(s) = \underline{q}_B + 2\underline{q}_C s + 3\underline{q}_D s^2, \text{ with } s_s \le s \le s_f \tag{9}$$

By using a cubic spline interpolation in the joint space, the position, the speed and the acceleration are smooth and continuous functions. However, the acceleration derivative is constant for each spline. When continuity or control of the acceleration derivative is essential, a higher degree polynomial should be chosen, which will result to increased computational cost for the algorithm.

2.4. Position and orientation deviations

The robot's end-effector coordinates $\underline{X}_r(s)$ are computed for the interpolated intermediate joint coordinates $\underline{q}(s)$ (Eq. (8)) by using the direct kinematics. The deviations between the desired $\underline{X}_d(s)$ and the robot's end-effector path $\underline{X}_r(s)$, must be computed to verify that the manipulator follows the desired path with the desired accuracy.

The position deviation is computed by the Euclidean distance.

$$\Delta P = \sqrt{(x_d(s) - x_r(s))^2 + (y_d(s) - y_r(s))^2 + (z_d(s) - z_r(s))^2} \tag{10}$$

where the desired path position to be followed is described by the coordinates $x_d(s)$, $y_d(s)$, $z_d(s)$ and the path that is actually followed by the robot's end-effector is described by the coordinates $x_r(s)$, $y_r(s)$, $z_r(s)$.

However, the orientation deviation cannot be described in a similar way and therefore is computed by a quaternion multiplication. A quaternion consists of a scalar and a vector part (Fig. 3). It is usually described by the notation $Q = [w, \underline{v}]$, where $w \in \Re$ is the scalar part and $\underline{v} = [v_x, v_y, v_z] \in \Re^3$ with $v_x, v_y, v_z \in \Re$ is the vector part. According to Dam et al. [13] a unit quaternion can be transformed to the following form $Q = [\cos\alpha, \sin\alpha \underline{v}_s]$,

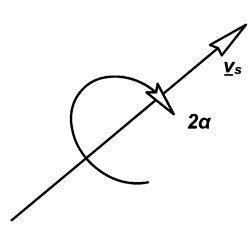

Fig. 3. Graphical representation of a quaternion.

which is represented graphically by a 3D vector $\underline{v}_s$ and a rotation angle $2a$.

The dot product of two unit quaternions $\underline{Q}_a$, $\underline{Q}_b$ results to

$$\underline{Q}_a \cdot \underline{Q}_b = ||\underline{Q}_a|| \cdot ||\underline{Q}_b|| \cos\gamma \Rightarrow \gamma = \arccos(\underline{Q}_a \cdot \underline{Q}_b) \tag{11}$$

since $\underline{Q}_a$, $\underline{Q}_b$ are unit quaternions $\|\underline{Q}_a\| = \|\underline{Q}_b\| = 1$.

As explained by Shoemake [14] two quaternions can coincide by rotating one of them towards the other by angle γ. From Eq. (11), the deviation between the desired and the actual orientation of the end-effector is given by

$$\Delta O = \arccos(\underline{Q}_d(s) \cdot \underline{Q}_r(s)) \tag{12}$$

with $\underline{Q}_d(s)$ and $\underline{Q}_r(s)$ the quaternions that describe the orientation of the desired path to be followed and the orientation of the robot's end-effector path, respectively.

3. The trajectory approximation algorithm

The trajectory approximation algorithm's main goal is to follow a parametric path given by $\underline{X}_d(s)$. The trajectory approximation algorithm approximates the desired path by comparing the position and orientation deviation between the desired path's coordinates and the robot's end-effector coordinates. If the deviations are not satisfactory, the desired path segment is divided into two smaller segments and the deviation comparison is repeated for the two new segments separately. On the other hand, if the deviations are smaller than the allowed ones, the robot can approximate the desired path with the desired accuracy and therefore, the algorithm is terminated.

Although the main purpose of the trajectory approximation algorithm was to follow the desired path, in this case it is used to compute the MVR at the nodes connecting the path segments. The general structure of the algorithm is presented in Fig. 4.

The manipulator's initial joint coordinates $\underline{q}(s_s) = [q_1(s_s)\ q_2(s_s) q_3(s_s) q_4(s_s) q_5(s_s) q_6(s_s)]^T$ are considered as given since they are provided by the optimization algorithm discussed at Section 4.

The trajectory approximation algorithm is analyzed in the following steps.

Step 1: Computing joint coordinates for the final node

Applying the inverse kinematic solution to the final node coordinates $\underline{X}_d(s_f)$ of the path to be followed, the node coordinates are transformed from manipulator end-effector coordinates to joint coordinates.

$$\underline{q}(s_f) = L^{-1}(\underline{X}_d(s_f)) \tag{13}$$

with L^{-1} the inverse kinematic solution.

Having the initial and final node coordinates in both: end-effector $\underline{X}_d(s_s)$, $\underline{X}_d(s_f)$ and joint coordinates $\underline{q}(s_s)$, $\underline{q}(s_f)$, ends the first step.

Step 2: Checks

However, some checks must be performed for the nodes located at s_n, where n denotes the referred nodes of the current segment, to avoid singularities and non-feasible solutions. Successfully passing these checks ensures that the manipulator can perform the task near the area of the current segment nodes.

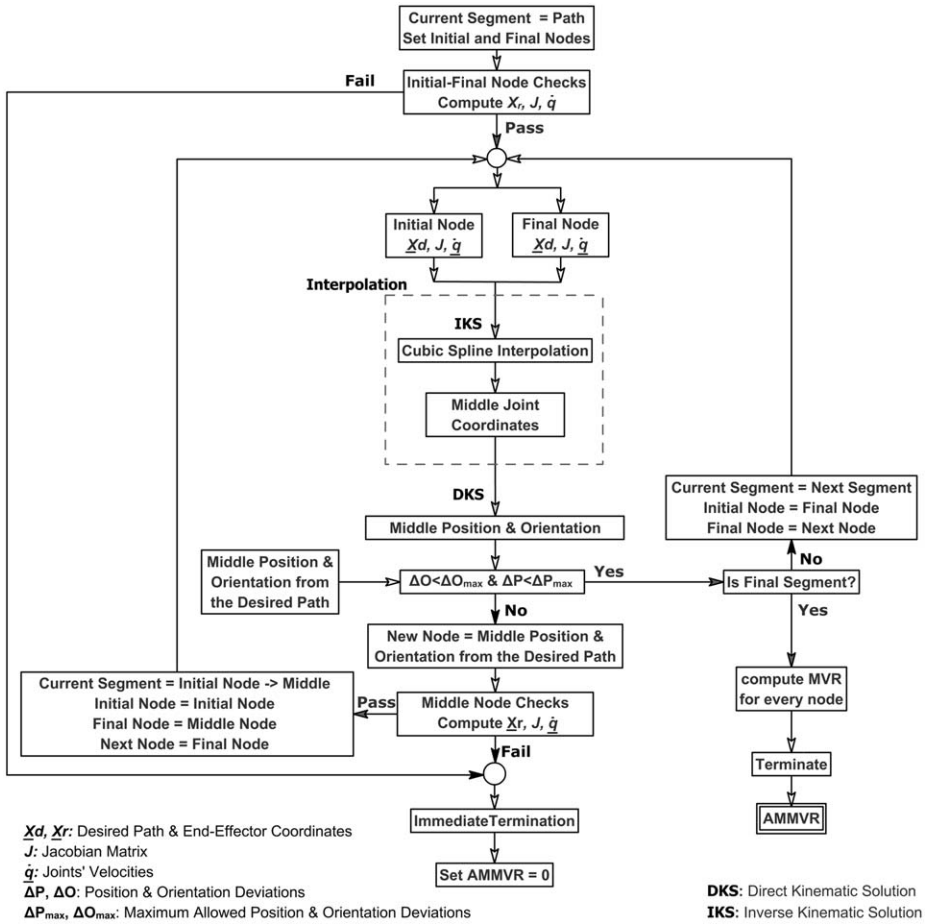

Fig. 4. Trajectory approximation algorithm diagram.

To avoid singularities the determinant of the jacobian matrix at the current segment node must not be equal to zero.

$$\det(J(\underline{q}(s_n))) \neq 0 \tag{14}$$

Moreover, the mechanical limits of the joint coordinates $\underline{q}(s)$ and their velocities $\dot{\underline{q}}(s)$ are taken into account and examined at the current segment nodes. Therefore, the joint coordinates must be inside the limits.

$$q_j^{\min} \leq q_j(s_n) \leq q_j^{\max} \text{ for } j = 1, \ldots, 6 \tag{15}$$

where $q_j(s_n)$ the jth element of vector $\underline{q}(s_n)$ and $q_j^{\min}$, $q_j^{\max}$ the respective minimum and the maximum limits of the joint coordinates. In addition, the velocity of the joint actuators is limited by the maximum velocity.

$$abs(\dot{q}_j(s_n)) \leq \dot{q}_j^{\max} \tag{16}$$

where $\dot{q}_j(s_n)$ the jth element of vector $\dot{\underline{q}}(s_n)$ and $\dot{q}_j^{\max}$ is the respective maximum velocity that the actuators of the manipulator can apply to both directions. The joint velocities $\dot{\underline{q}}(s_s)$ and $\dot{\underline{q}}(s_f)$ can be computed from the desired end-effector velocities $\dot{\underline{X}}_d(s_s)$ and $\dot{\underline{X}}_d(s_f)$, respectively, through the equations $\dot{\underline{q}}(s_s) = J^{-1}(\underline{q}(s_s))\dot{\underline{X}}_d(s_s)$ and $\dot{\underline{q}}(s_f) = J^{-1}(\underline{q}(s_f))\dot{\underline{X}}_d(s_f)$.

If one of the checks fails then the trajectory interpolation algorithm is terminated, since the solution is non-feasible.

Step 3: Middle joint coordinates

If the previous checks are successful, then using the cubic spline interpolation from Eq. (8) with $s_m = (ss+sf)/2$.

$$\underline{q}(s_m) = \underline{q}_A + \underline{q}_B s_m + \underline{q}_C s_m^2 + \underline{q}_D s_m^3 \text{ with } s_s \leq s_m \leq s_f \tag{17}$$

And the joints' velocities for the current segment are computed from Eq. (9).

$$\dot{\underline{q}}(s_m) = \underline{q}_B + 2\underline{q}_C s_m + 3\underline{q}_D s_m^2, \text{ with } s_s \leq s_m \leq s_f \tag{18}$$

Step 4: Compute the deviations

The coordinates of the robot's end-effector at the middle of the current segment are computed by applying the direct kinematic solution to the middle joint coordinates provided by step 3.

$$\underline{X}_r(s_m) = L(\underline{q}(s_m)) \tag{19}$$

The deviation between the desired and the actual position of the end-effector at the middle point is given by

$$\Delta P = \sqrt{(x_d(s_m) - x_r(s_m))^2 + (y_d(s_m) - y_r(s_m))^2 + (z_d(s_m) - z_r(s_m))^2} \tag{20}$$

After transforming the orientation to unit quaternions based on Dam et al. [13], the orientation deviation is

$$\Delta O = \arccos(\underline{Q}_d(s_m) \cdot \underline{Q}_r(s_m)) \tag{21}$$

Step 5: Compare the deviations

The deviations computed in the previous step are compared with the maximum allowed deviations $\Delta P_{\max}$, $\Delta O_{\max}$, which are defined by the desired path following accuracy.

If at least one deviation is greater than the respective maximum allowed, then a new node $\underline{X}_d(s_m)$ is placed in the middle of the current segment, dividing it into two parts: the current, from $\underline{X}_d(s_s)$ till $\underline{X}_d(s_m)$, and the next segment, from $\underline{X}_d(s_m)$ till $\underline{X}_d(s_f)$. The algorithm is then repeated for the two new segments from step 3.

Else if both are smaller, i.e. $\Delta P \leq \Delta P_{\max}$ and $\Delta O \leq O_{\max}$, then the algorithm moves to the next segment, if this is not the final one. In the case that this is the final segment, the algorithm moves to the 6th step for a successful termination.

Step 6: Calculate the MVR

In the final step, the MVR values are computed for all the nodes using Eq. (6), which is written for this specific case

$$r_v = \frac{1}{\sqrt{\underline{u}_v^T(s_n)(J_v(\underline{q}(s_n))J_v^T(\underline{q}(s_n)))^{-1}\underline{u}_v(s_n)}} \tag{22}$$

$$\text{with } \underline{u}_v(s_n) = \left(\frac{(W_x^{1/2}\dot{\underline{X}}_r(s_n))}{\sqrt{(W_x^{1/2}\dot{\underline{X}}_r(s_n))^T(W_x^{1/2}\dot{\underline{X}}_r(s_n))}} \right) \text{ and}$$

$$J_v(\underline{q}(s_n)) = (W_x^{1/2}J(\underline{q}(s_n))W_q^{-1/2})$$

Finally, the algorithm is successfully terminated returning the AMMVR value as the minimum of the MVR node values.

$$\text{AMMVR}(\underline{q}(s_s)) = \min[(r_v)_n] \text{ with } n = 1, 2, \ldots, k \tag{23}$$

4. Optimal location search algorithm

The optimization problem that has to be solved is defined by the following objective function.

$$\text{OF} = \text{AMMVR}(\underline{q}(s_s)) \tag{24}$$

The optimization is performed by a genetic algorithm (GA) for the following reasons. First of all a GA provides a robust search in complex and large spaces. In our case the trajectory approximation algorithm results to a very complex non-continuous and non-linear space for searching. The determination of AMMVR, which is the objective function, is procedural and since it is calculated at the node points, it is non-continuous and non-differentiable. Therefore, this optimization problem is well suited for GA searching. In general, a simple GA solves unconstrained problems but the proposed formulation of the fitness function allows constraints to be incorporated easily, so that the introduced method can be extended to work in an environment cluttered with obstacles or under other constraints.

The GA is initialized by producing a random population, with each chromosome composed of the manipulator's 6 joints' angles to reach the desired path's starting node $\underline{q}(s_s)$. The trajectory approximation algorithm is used to compute the chromosomes' fitness, which is the objective function under the constraints.

To reject the non-feasible solutions, which do not satisfy the constraints, the fitness function is set equal to zero, as it was proposed by Nearchou and Aspragathos [15]. This means that the chromosome violating the constraints is considered as unfit and as a result has zero probability to survive and be reproduced in the future generations. Therefore, the fitness function takes the following form.

$$\text{fitness} = \begin{cases} \text{AMMVR}(\underline{q}(s_s)) & \text{for} \begin{cases} \det(J(\underline{q}(s_n))) \neq 0, \text{ and} \\ q_j^{\min} \leq q_j(s_n) \leq q_j^{\max}, \text{ and} \\ abs(\dot{q}_j(s_n)) \leq \dot{q}_j^{\max} \end{cases} \\ 0 & \text{for} \begin{cases} \det(J(\underline{q}(s_n))) = 0, \text{ or} \\ q_j^{\min} > q_j(s_n), \text{ or} \\ q_j(s_n) > q_j^{\max}, \text{ or} \\ abs(\dot{q}_j(s_n)) > \dot{q}_j^{\max} \end{cases} \end{cases} \tag{25}$$

Based on the fitness function, the chromosomes – or technically the manipulator's initial joint angles $\underline{q}(s_s)$ – are subjected to reproduction, crossover and mutation and finally the new improved generation is produced.

The suboptimal or optimal (sub/optimal) solution is reached using two termination conditions. The stronger one terminates the algorithm when for enough consecutive generations the same solution is produced, which is assumed to be the optimal one. The second one terminates when a predefined maximum number of generations is reached. Both termination conditions should be used with caution and after a considerable number of test runs to ensure that the solution is optimal or near optimal (Goldberg [16]).

5. Experimental simulation

The introduced algorithm is quite generic; therefore it can be applied to any non-redundant manipulator with any mechanical structure. For the purposes of verifying the effectiveness of the proposed algorithm, a PUMA 560 robot was selected, since it is a common type of a 6 DOF industrial robot. The manipulator's rotational joints and geometrical data are presented in Fig. 5.

The desired path can be of any shape represented by an arbitrary curve. However, the algorithm due to the nature of the aforementioned interpolation can deal only with either convex or concave curves. This precaution measure is taken to avoid situations such as when the desired path is a symmetric S-type. Then at least considering the position deviation at the middle would be very low, causing the trajectory approximation algorithm to terminate without following the S path, but a straight line. In such cases, when a more complicated path including convex and concave parts is considered, the interpolation algorithm should be used for each convex or concave part separately.

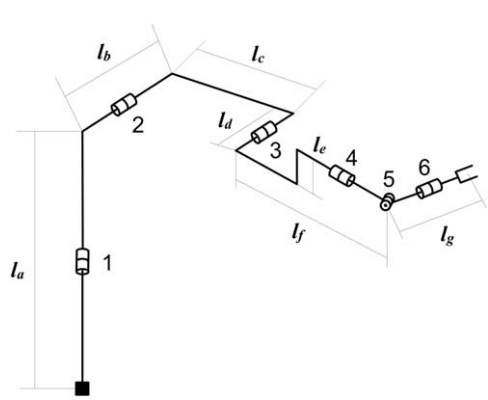

Fig. 5. Puma 560 joints' numbering and dimensions. $la = 1300$ mm, $lb = 1500$ mm, $lc = 1500$ mm, $ld = 500$ mm, $le = 200$ mm, $lf = 1200$ mm and $lg = 0$mm.

Table 1
Optimal location search algorithm results and the respective parameters.

Simulation number	Number of generations	Population size	Crossover probability (p_c)	Mutation probability (p_m)	Maximum allowed position deviation (mm)	Maximum allowed orientation deviation (deg.)	Initial solution (AMMVR)	Sub/optimal solution (AMMVR)
1	500	200	0.7	0.03	0.1	0.001	0.15986	0.24075
2	500	200	0.6	0.10	1	0.1	0.16480	0.26120
3	500	200	0.7	0.05	0.1	0.001	0.16136	0.26107
4	329	100	0.9	0.10	1	0.1	0.15933	0.26392
5	367	150	0.7	0.01	1	0.1	0.18465	0.23554
6	500	200	0.9	0.03	0.1	0.001	0.17774	0.26205
7	500	100	0.9	0.10	0.1	0.001	0.13213	0.23866
8	1000	200	0.7	0.05	1	0.1	0.16015	0.26051
9	170	200	0.8	0.10	1	0.1	0.15627	0.23188
10	945	150	0.8	0.05	1	0.1	0.16252	0.27441

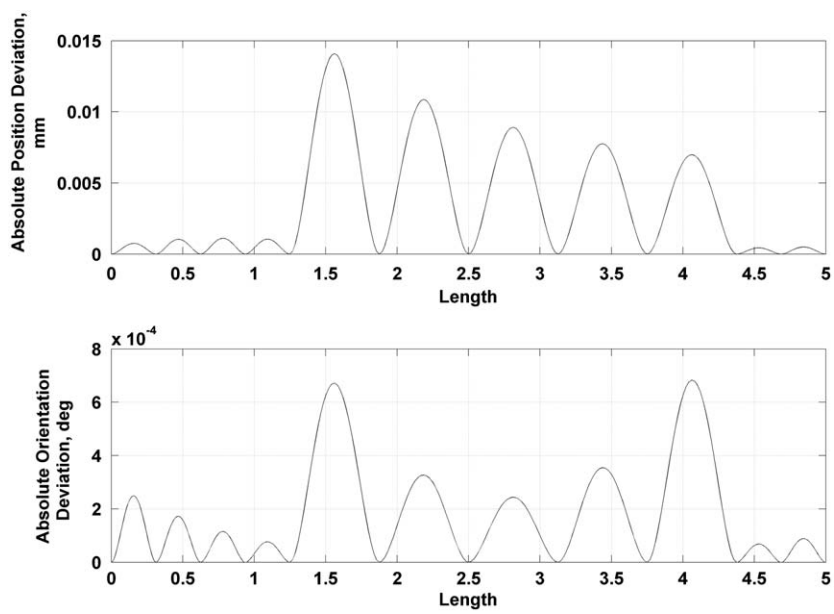

Fig. 6. Absolute values of position and orientation deviations through the path following task of simulation 1 for the sub/optimal solution (AMMVR = 0.24075).

For the simulations performed, the length function of time t is given by $s(t) = t$ for $t \in [0, 5]$ s and the path following task must be performed in a given time of 5 s.

6. Results and discussion

A considerable number of simulations ran under a variety of control parameters for the optimal location search algorithm and the most representative are presented in Table 1. The first four columns of the table, after the simulation number column, illustrate the GA parameters that were used for the simulations. The first one of these four columns shows the generation for which the optimal solution is found, the second one the population size and the following two the probabilities of crossover and mutation, respectively. At the next two columns the maximum allowed position and orientation deviations of the trajectory approximation algorithm are shown. In the last two columns the initial generation best solution and the suboptimal or optimal solution, i.e. the maximum value of the AMMVR, is presented.

Two cases were considered for testing the proposed approach. The first case has maximum allowed deviations for the position and orientation equal to 1 mm and 0.1°, respectively, while the second one is stricter with maximum allowed deviations 0.1 mm and 0.001°, respectively.

It should be noticed that the number of generations varies depending on which termination condition takes effect. For example, simulation 4 ends at the 329th generation, because of the best solution being constantly the same for more than 100 generations, whereas simulation 2 ends at the 500th generation. The early termination might cause the algorithm to fail in reaching a sub/optimal solution.

The absolute values of the deviations throughout the path are plotted for simulation 1 in Fig. 6. As expected the actual deviations are under the maximum allowed values, which are 0.1 mm and 0.001°, respectively, whereas the actual deviations do not exceed the values of 0.014 mm and 0.00067°, respectively with only 10 nodes. Therefore, the algorithm performs successfully the path following task with the desired accuracy.

A good approximation of the minimum MVR along the path is essential for the optimal location search algorithm. Therefore, the deviation of the actual MVR along the path to the approximated is investigated experimentally. Figs. 7 and 8 show the approximate and the actual MVR plots along the path for the sub/optimal solution of simulation 6 and the initial generation best solution, respectively. It can be observed that the deviation is quite reasonable and the error in approximating the minimum MVR is

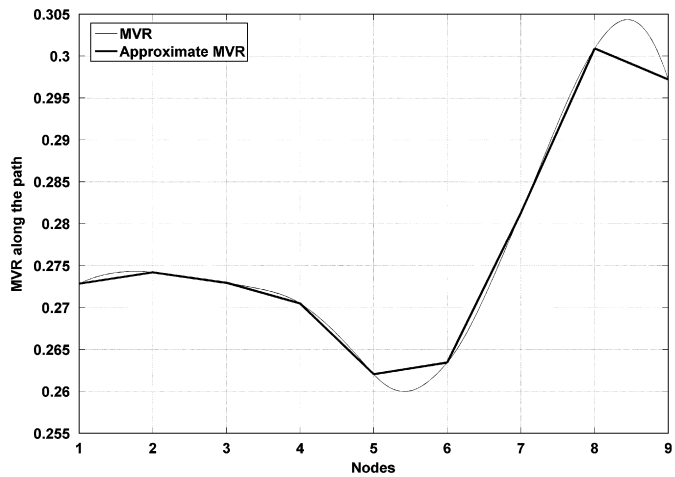

Fig. 7. MVR values and approximate MVR values along the path of simulation 6 for sub/optimal solution (AMMVR = 0.26205).

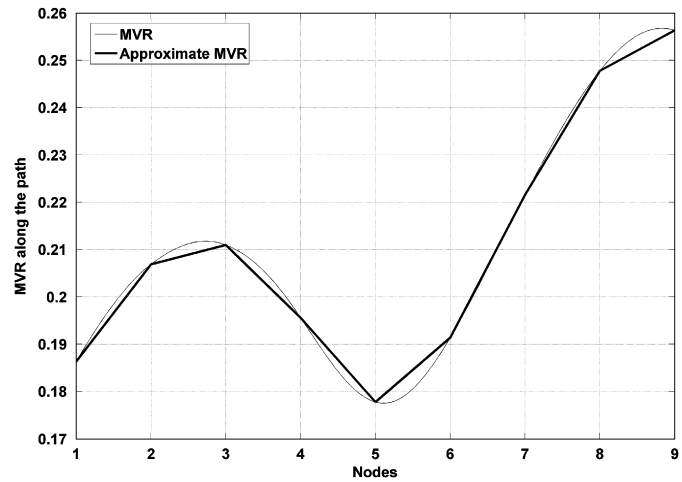

Fig. 8. MVR values and approximate MVR values along the path of simulation 6 for best solution of the initial generation (AMMVR = 0.17774).

equal to 0.8% and 0.1%, respectively, and occurs between nodes 5 and 6 for both cases. It should be noted that the MVR in the initial generation best solution case remains always under the value of 0.26000 (Fig. 8), while in the sub/optimal solution (Fig. 7) the MVR never falls under that value.

Fig. 9 illustrates the best AMMVR values throughout the optimization as the number of generations increases for simulation 8. A good approximation of the suboptimal AMMVR value (0.25833) is achieved at the 128th generation. Therefore, it can be concluded that the optimal location search algorithm converges quite fast to the sub/optimal solution. The final AMMVR value 0.26051 is reached at the 750th generation and the algorithm is terminated at the 1000th generation since no better solution was produced for 250 generations.

As the AMMVR value increases through the generations, the manipulator base is placed closer to the near optimal location. This is illustrated in Fig. 10, where the manipulator's two different configurations are compared. The configuration on the left is drawn for the best solution of the initial generation, whereas the right one is drawn for the sub/optimal solution, after the maximization of the AMMVR. The thick line segments represent the manipulator, with the end-effector being placed at the nodes, presented with small black spheres. The manipulator's orientation is shown by the thick linear segments set on the nodes (Fig. 10: enlarged detail). The thin curved segment represents the position of the path to be followed, whereas the thin lines starting from the nodes the desired orientation (Fig. 10: enlarged detail). For a better comparison the position and orientation of the manipulator's base are set at the origin $O[0, 0, 0, 0, 0, 0]$ for both subplots.

The best solution of the initial generation has a smaller AMMVR, equal to 0.15627, than the optimal one with an AMMVR equal to 0.23188. Comparing the two manipulator's configurations, the manipulator for the sub/optimal solution is more extended than the initial one. This is in fact what the manipulability theory predicts, since the end-effector can succeed higher velocities if the manipulator has an extended configuration, than if it was more retracted. However, as the location of the path reaches the robot workspace the MVR is reduced since the determinant of the Jacobian tends to zero due to boundary singularities.

In order to examine the joint's motion and velocity as the algorithm moves towards the optimal solution, two cases are compared: the sub/optimal solution (AMMVR = 0.23866) and the best solution from the initial generation (AMMVR = 0.13213) for

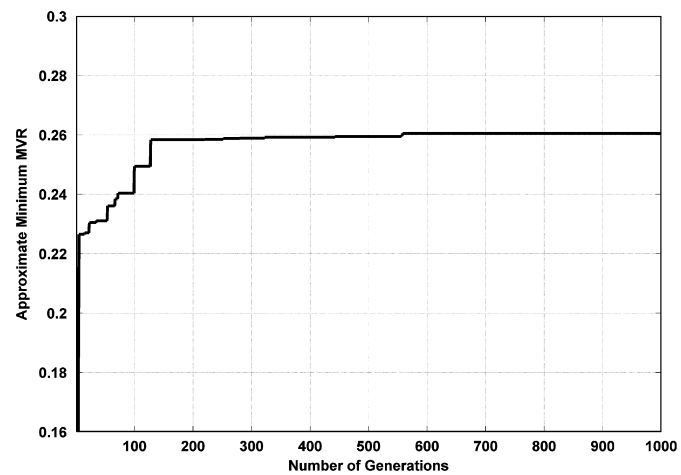

Fig. 9. AMMVR optimization through optimal location search algorithm generations for simulation 8 (AMMVR = 0.26051).

simulation 7. Fig. 11 shows each of the six joints motion for both cases. Both provide smooth motion as expected from the cubic spline interpolation.

Fig. 12 shows that the joints' velocity is also smooth as expected from the interpolation algorithm. It is observed that the greater AMMVR case has the absolute maximum joint velocity of 11°/s at joint 1, compared to the 18°/s at joint 4 of the case with the smaller AMMVR.

In Fig. 13 the differences between the maximum and the minimum values, i.e. the max–min variation, of the joints' angles and the joints' velocities for simulation 7 are plotted for each joint separately and for the accumulated sum of the six joints. It is noticeable that there is a general reduction of the joint motion max–min variations for the greater AMMVR, constraining the maximum values, as for example in the joint angle max–min variations subplot for joints 3, 4, 5 and 6. Focusing on the accumulated sum max–min variation of the six joints motion, it is clear that the variation of the sub/optimal location is 36% lower than the accumulated sum of the best solution of the first generation. The results are similar for the velocities, since the high max–min variation values of the initial solution are reduced in the sub/optimal case for joints 1, 3, 4, 5 and 6, whereas joint 2 is slightly increased. Comparing the accumulated sum of the joint velocities the reduction is 59%. Therefore, it can be

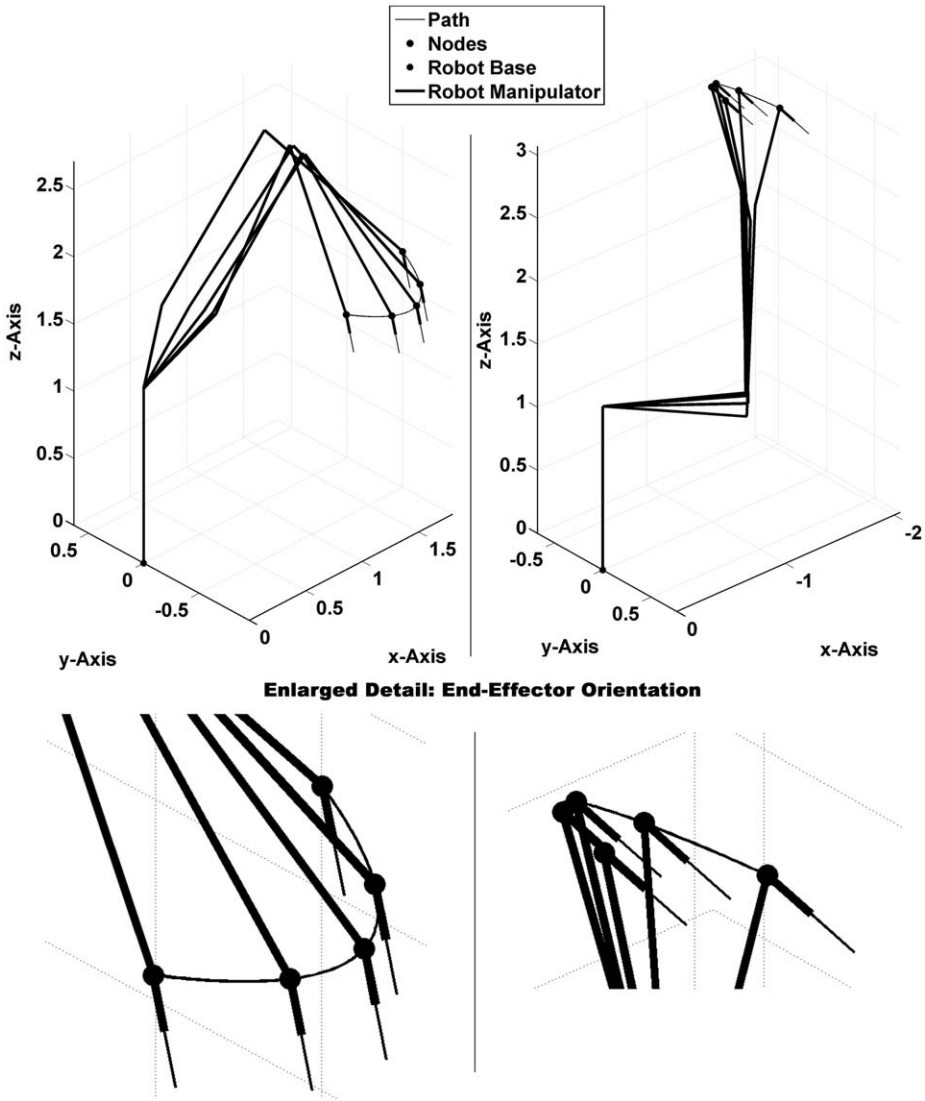

Fig. 10. Manipulator's configuration of simulation 9 for the best solution of the initial generation (AMMVR = 0.15627) and for the sub/optimal solution (AMMVR = 0.23188).

concluded that the joints' angle motion and velocities have reduced max–min variations compared to solutions with lower AMMVR values.

However, the reduced max–min variation is not the only indication for the benefits derived by using the proposed algorithm to determine the optimal location of the robot relatively to the task. Combining Fig. 13 with Figs. 11 and 12, it can be concluded that as the AMMVR increase through generations of the GA, the joint angles and velocities are minimized.

Focusing on the end-effector velocity $\dot{\underline{X}}_r(s)$, the absolute linear and angular velocity measures of the end-effector for the initial generation best solution and the sub/optimal solution of simulation 7 are plotted in Fig. 14. The end-effector velocity $\dot{\underline{X}}_r(s)=\begin{bmatrix}\underline{v}(s) & \underline{\omega}(s)\end{bmatrix}^T=\begin{bmatrix}v_x(s) & v_y(s) & v_z(s) & \omega_\phi(s) & \omega_\theta(s) & \omega_\psi(s)\end{bmatrix}^T$ is computed by $\dot{\underline{X}}_r(s)=J(\underline{q}(s))\dot{\underline{q}}(s)$, where $\underline{v}$, $\underline{\omega}$ are the linear velocity vector and the angular velocity vector, respectively.

The absolute linear velocity measure is identical for the two solutions and has a smooth parabolic form. On the other hand, the absolute angular velocity measure has some differences. For the first generation best solution, the angular velocity measure oscillates with a higher amplitude and frequency, whereas for the sub/optimal solution it is smoother. One possible reason for this measure difference is that the best solution of the initial generation can be near singularity affecting the calculation of the robot end-effector velocity.

Examining the individual coordinates of the end-effector velocity for the two solutions of simulation 7 (Fig. 15), it is evident that the case with the higher AMMVR has in general smoother end-effector velocity coordinates.

It can be concluded that the algorithm is ideal for path following applications, where smooth velocity is needed, such as welding and laser cutting. The fact that the joints' velocities are minimized, as the AMMVR increases and the end-effector velocities remain around the same values, leads to the conclusion that the optimal solution is the most velocity efficient solution.

7. Conclusion

In the introduced approach, an offline algorithm is developed that can be used for the optimal robotic workcell design. Its main

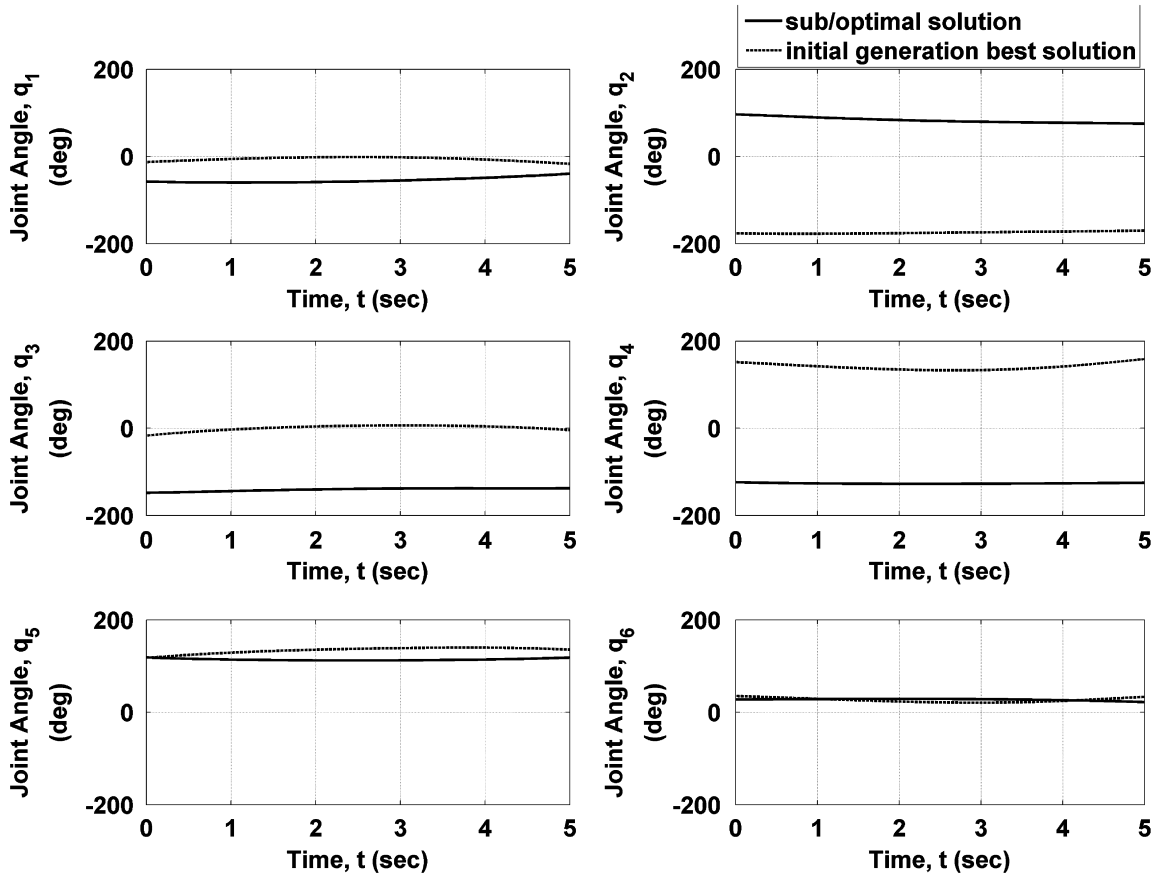

Fig. 11. Joint angles of simulation 7 for the best solution of the initial generation (AMMVR = 0.13213) and for the sub/optimal solution (AMMVR = 0.23866).

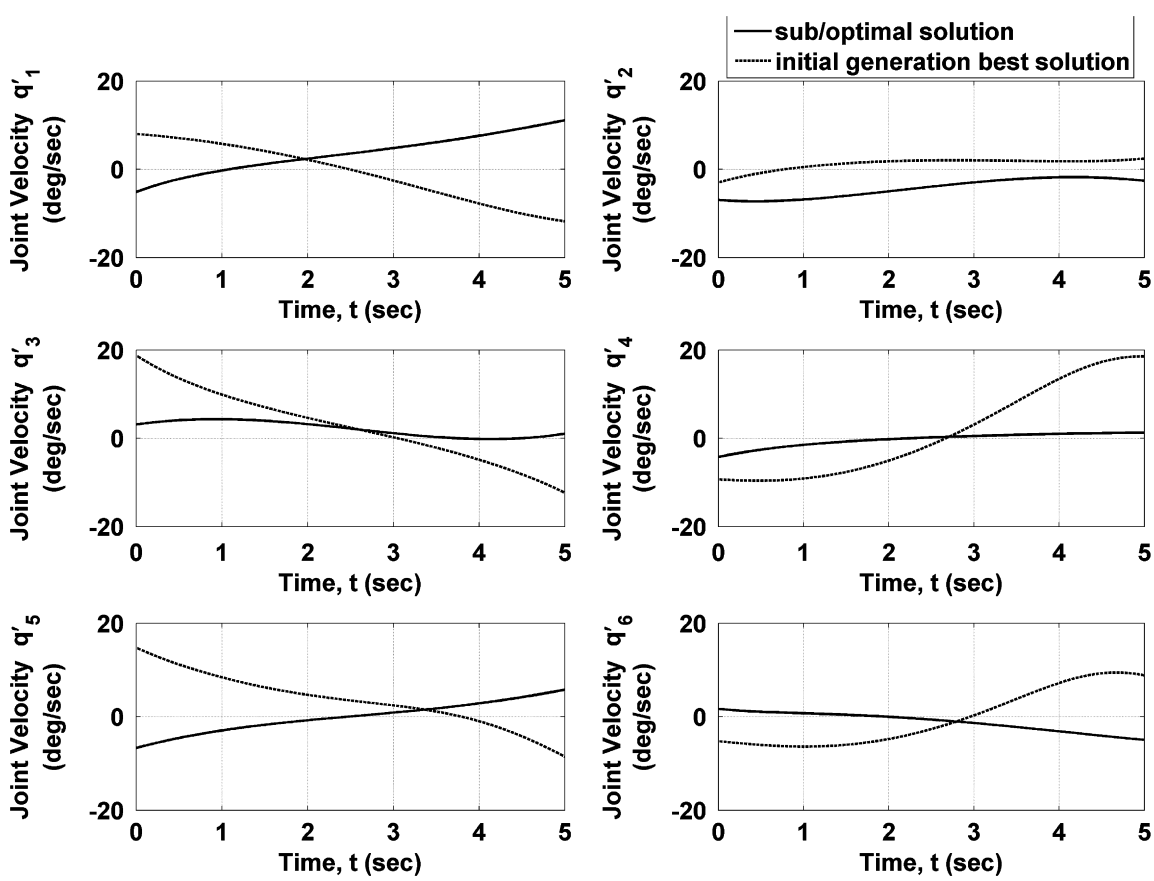

Fig. 12. Joint velocities of simulation 7 for the best solution of the initial generation (AMMVR = 0.13213) and for the sub/optimal solution (AMMVR = 0. 23866).

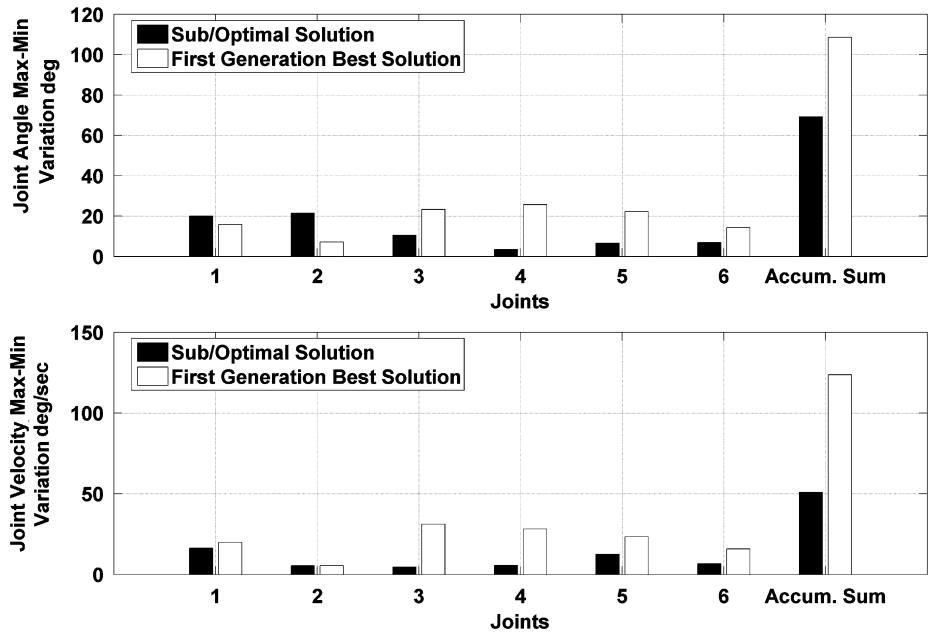

Fig. 13. Max–min variation comparison of joint angles and velocities of simulation 7 for the sub/optimal solution (AMMVR = 0.23866) and for the best solution of the first generation (AMMVR = 0.13213).

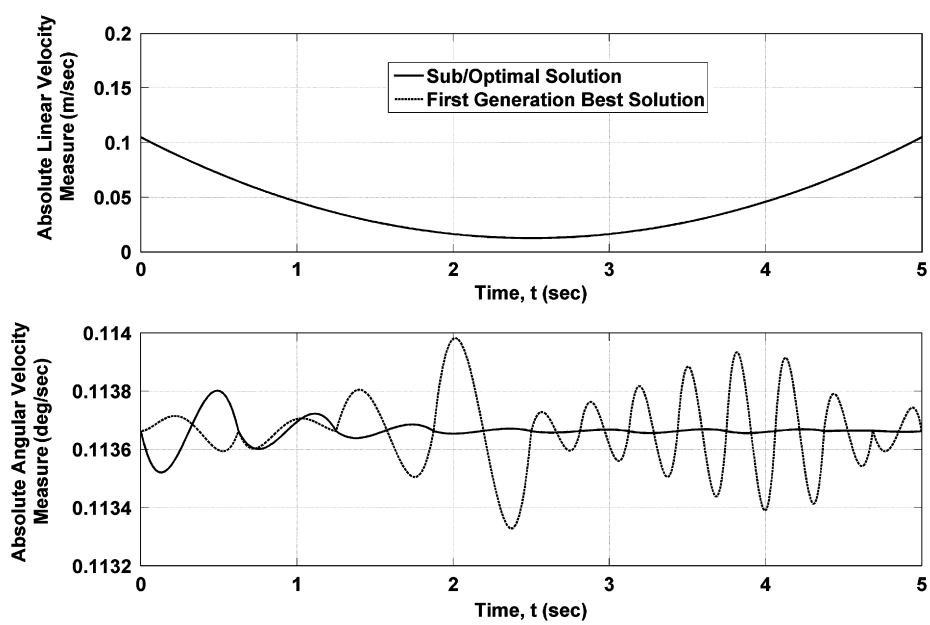

Fig. 14. End-effector velocity measures of simulation 7 for the sub/optimal solution (AMMVR = 0.23866) and for the best solution of the first generation (AMMVR = 0.13213).

aim is to determine the optimal or suboptimal relative position and orientation of a non-redundant robot's base to a position and orientation path following task, achieving the maximum possible end-effector velocity with the most efficient joint velocities of the end-effector. The near optimal solution promises smooth and maximum end-effector velocity performance, while at the same time it results to reduced robotic equipment wear and energy consumption. The path following task cycle time can also be reduced for the optimal solution, since increasing the minimized joints' velocity even higher end-effector velocities can be achieved.

The results derived from the simulation, demonstrate the algorithm's effectiveness, concluding that the AMMVR is a powerful global index, compatible to the optimization specifications. The optimization performed by the optimal location search algorithm is achieved relative quickly, since it converges to the greatest AMMVR in a few generations. It should also be noticed that the path following task for both position and orientation can be achieved with just a few nodes. Considering the optimization specifications, as demonstrated by the simulation, the joints' motion and velocity are minimized for the near optimal solution, whereas the end-effector velocity remains the same, resulting to a more efficient use of the manipulator. Furthermore, the motion and velocities of the joints, as well as the velocity of the end-effector are smooth and continuous, due to the cubic spline interpolation.

To conclude, the optimal location search algorithm provides a near optimal location of any non-redundant manipulator relative

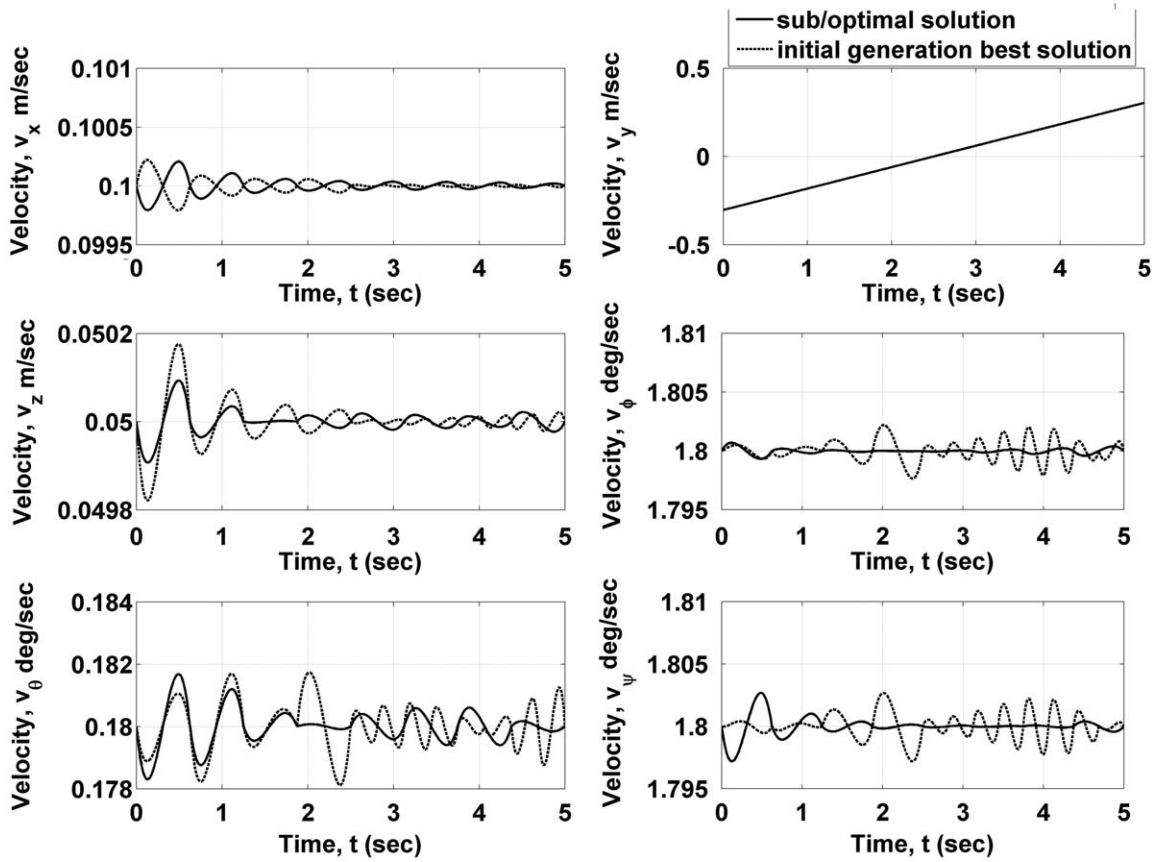

Fig. 15. End-effector velocity coordinates of simulation 7 for the sub/optimal solution (AMMVR = 0.23866) and for the best solution of the first generation (AMMVR = 0.13213).

to a path following task, considering the velocity performance of the end-effector. The relative placement of the robot and task derived from the proposed approach, can exploit the most out of the joint velocity capabilities through any control algorithm, making it suitable for applications such as welding, painting, laser or water jet cutting that demand smooth and fast end-effector motion along a parametric path.

References

[1] Yoshikawa T. Analysis and control of robot manipulators with redundancyIn: Robotics Research: The First International Symposium. MIT Press; 1984. p. 735–47.
[2] Dubey R, Luh J. Redundant robot control using task based performance measures. Journal of Robotic Systems 1988;5(5):409–32.
[3] Salisbury JK, Craig JJ. Articulated hands, force control and kinematic issues. International Journal of Robotics Research 1982;1(1):4–17.
[4] Ermolov I, Podurajev J. Intelligent robot motion planning based on hybrid manipulator measures calculus. In: Large Scale Systems: Theory and Applications (IFAC LSS'98), Patras: 1998. p. 1132–7.
[5] Nelson B, Pederson K, Donath, M. Locating assembly tasks in menipulator's workspace. In: IEEE international conference on robotics and automation. vol. 2. 1987. p. 1367–72.
[6] Martin DP, Baillieul J, Hollerbach JM. Resolution of kinematic redundancy using optimization techniques. IEEE Transactions on Robotics and Automation 1989;5(4):529–33.
[7] Uchiyama M, Shimizu K, Hakomori K. Performance evaluation of manipulators using the Jacobian and its application to trajectory olanning. In: Proceedings of second international symposium on robotics research, Kyoto: 1984. p. 155–62.
[8] Nelson B, Donath M. Optimizing the location of assembly tasks in a manipulators workspace. Journal of Robotic Systems 1990;7(6):791–811.
[9] Sobh TM, Toundykov DY. Optimizing the tasks at hand. IEEE Robotics and Automation Magazine 2004;p:78–85.
[10] Aspragathos NA, Foussias S. Optimal location of a robot path when considering velocity performance. Robotica 2002;5:139–47.
[11] Taylor RH. Planning and evolution of straight line manipulator trajectories. IBM Journal of Research and Development 1979;23:424–36.
[12] Aspragathos NA. Cartesian trajectory generation under bounded position deviation. Mechanisms and Machine Theory 1998;33(6):679–709.
[13] Dam EB, Koch M, Lillholm M. Quaternions, Interpolation and Animation. Denmark: University of Copenhagen; 1998. URL: 〈http://www.itu.dk/people/erikdam/DOWNLOAD/98-5.pdf〉; November 2006.
[14] Shoemake K. Animating rotation with quaternion curves. Computer Graphics 1985;19(3):245–54.
[15] Nearchou AC, Aspragathos N. Obstacle avoidance control of redundant manipulators using genetic algorithms. In: Third IEEE mediterranean symposium directions in control and automation. Limassol, Cyprus: 1995. p. 60–7.
[16] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1989.

Athanasios Nektarios was born in Athens, Greece in 1980. Mechanical Engineering and Aeronautics degree, University of Patras, Patras, Greece, 2004. MSc and DIC in Control Systems, Imperial College London, London, UK, 2005. MSc in Management, London School of Economics and Political Science, London, UK, 2006.

He is currently appointed the position of Design and Manufacturing Engineer, Kinematik S.A. Machinery Construction, Athens, Greece.

His research interest include path following, motion control and control systems. He is member of the Technical Chamber of Greece and of academic/research societies concerning robotics, automation and control systems.

Nikos A. Aspragathos leads the Robotics Group Mechanical and Aeronautics Engineering Department, University of Patras, Greece. Electrical Engineering degree (1975), and the PhD (1981), School of Engineering, University of Patras, Greece.

He has joined Greek industrial companies as a constructor and maintenance Engineer (1976–1980). In 1982 he appointed Lecturer in Mechanical and Aeronautical Engineering Department, University of Patras. He was a visiting member of staff in UWIST, Cardiff from 1987 to 1988.

His main research interests are robotics, intelligent control and design, industrial automation, CAD/CAM, and simulation. He has developed algorithms for robot path planning based on intelligent systems for handling of non-rigid materials and for microassembly automation. He is reviewer in several Journals and member of the editorial board of Mechatronics Journal. He has been and is currently involved in research projects funded by Greek and European Union sources.