

49th CIRP Conference on Manufacturing Systems (CIRP-CMS 2016)

Trajectory Planning for Reconfigurable Industrial Robots Designed to Operate in a High Precision Manufacturing Industry

Oliver Avram^{a*}, Anna Valente^a^a*SUPSI, ISTEPS Institute of Systems and Technologies for Sustainable Production, Galleria 2, Manno 6928, Switzerland*^b*Second affiliation, Address, City and Postcode, Country** Corresponding author. Tel.: +41-058-666-6610; fax: +41-058-666-6571. E-mail address: oliver.avram@supsi.ch

Abstract

This paper presents an algorithm for an automated planning of time-jerk optimal trajectories on a reconfigurable robot with an anthropomorphic structure. The trajectory planning algorithm is designed to provide reachable input set-points for the reconfigurable control system to accommodate not only changes in task objectives but also various reconfigurations of the robot kinematic structure and motion behavior adaptation in response to operational constraints. Firstly, appropriate reference frames are defined on the robotic modules and the position and orientation of the end effector in the Cartesian space is dynamically generated through a reconfigurable forward kinematics module. Secondly, through an iterative inverse kinematic method, the robot configuration space for the start and goal destinations and the geometric path for every motion tasks are generated. Finally, the joint space trajectories with coordinated motion profiles are generated by incorporating the motion laws and joint kinematic parameter values. The general outcome of the algorithm is a trajectory plan of the robot joints expressed as a time history of the robot motion. The trajectories are dynamically generated to satisfy widely varying tasks, constraints and objectives aiming towards a fully reconfigurable control without any hardware or software adjustment.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 49th CIRP Conference on Manufacturing Systems

Keywords: Robot reconfiguration; Motion planning; Jerk-bounded trajectory

1. Introduction and motivation

Reconfigurable industrial robots possess the key ability to leverage their physical configuration flexibility, such as degrees of freedom, reach and working-space, into strong competitive assets in open-ended manufacturing environments characterized by high product diversity and rapid evolution of production demand [1,2]. They are versatile and more adaptive than conventional systems as the ability to reconfigure allows to disassemble and reassemble to form new configurations which are better suited for new tasks. However, this ability to accommodate diverse manufacturing tasks with optimal configurations requires not only a thoroughly designed repository of mechanical, electrical and electronic modular components but also a reconfigurable planning and control of the robot motion.

The motion planning of commercially available robots is generally optimized for extremely specific set of working conditions which makes them unsuitable for use in environments characterized by widely varied tasks, objectives and constraints.

Over the years the researchers have proposed various methods to tackle the problem of trajectory planning for industrial robots. Some authors dedicated their efforts to time-optimal trajectory planning, mainly in order to address the need for an increased productivity [3-6]. Alternatively, some researchers have favored increasing trajectory smoothness in a trade off with time-optimality [7-10]. These approaches aimed the development and implementation of fast trajectories combined with optimal jerk values to reduce the excitation of the resonant frequencies of the mechanical system.

Although in the past time-optimal and smooth robot trajectories were proposed by several authors they were never applied to reconfigurable robots capable of quick adjustment of production capacity and functionality. Current industrial robots have physical limitations with respect to their configuration and capabilities since they are preconfigured to do specific tasks. Their links and joints are not modular and their structure is characterized by fixed kinematic parameters making them unsuitable for adapting their motion to frequent and unpredictable manufacturing tasks.

The method proposed in this paper introduces a strategy to implement reconfigurable concepts in the generation of trajectories that can cope with rapidly evolving requirements; a defining characteristic of today's manufacturing industry. It implements a direct offline approach to the motion equations where the path planning and time history of the motion are analyzed together, as opposed to indirect methods which first calculate the space of possible paths and then adjust a trajectory. The planner can also freely adapt the trajectories to the production requirements since the robot is not constrained to follow predefined paths and the kinematic characteristics are determined during the trajectory generation.

This research is intended to lay background for future studies in fully reconfigurable control systems. It proposes a generic method enabling the automated generation of pick and place trajectories to adapt to the kinematic reconfigurations of the robot as well as to operational constraints enforced by limited motion range policies imposed on joints experiencing degrading performance.

2. ReRob

The current paper discuss the trajectory planning as applied to ReRobI, that is the first reconfigurable robot whose performance targets the traditional industrial manipulators. ReRob I is conceived as a reconfigurable industrial robot characterized by a serial anthropomorphic structure. At the present stage of the research, the most appealing configuration is a 6 DoF and a spherical wrist as it is the most popular in the industrial world. The arm presents two DoF at the shoulder, one at the elbow and three at the wrist. It has been designed for high precision assembly of micro components in the optoelectronics field[11].



Figure 1. ReRob mechanical representation

3. Methodology overview

The proposed methodology has been implemented as a number of MATLAB® functions combined with additional tools

for robot modeling and simulation provided by the Corke's Robotics Toolbox [12]. It was designed for an automated offline trajectory planning for pick and place routines relevant to the reconfigurable robot arm. In our case the pick and place operations comprehend the handling of optoelectronic components for both assembly and transfer manipulations between successive destinations. A set of an initial and final destination coordinates associated to a task as well as an entire list of tasks pertaining to a specific production scenario are known a priori and stored in a MySQL database. The geometrical information pertaining to the modular elements of the robot (i.e. links, joints, end-effectors and tips) is also stored in the database. The method is organized in three main phases, namely the reconfigurable forward kinematics, the trajectory planning and the part program generation.

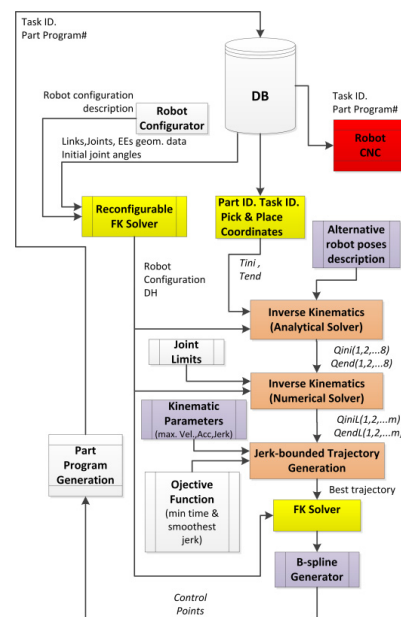


Figure 2. Methodology overview

3.1. Reconfigurable kinematics

The first problem to appear in reconfigurable robots is that the modeling of their kinematic and dynamic functionality cannot rely upon a fixed-robot geometry since they are closely related to every individual configuration. A reconfigurable kinematics algorithm is proposed for an automated generation of the DH parameters specific to a configuration without having them explicitly programmed. The modeling of this phase is facilitated by a framework consisting of a component database, a procedure to assign reference frames on individual modules and to represent modular robot architecture as well as a configuration-independent technique for an automated extraction of kinematic parameters based on line geometry and vector algebra[13].

The information about a specific configuration of the robot satisfying a given set of production requirements is computed by a robot configurator algorithm and provided as input to the reconfigurable forward kinematics solver. The objective of the

reconfigurable kinematics solver is to automatically generate the specific set of DH parameters as soon as a new configuration of the robot arm, assembled from kinematically independent modular elements, becomes available. The term configuration is used along this article to denote a particular structure of the robot characterized by the number, type and disposition sequence of reconfigurable robotic modules.

A generic reconfigurable kinematic model represents a forward step towards a reconfigurable architecture having the capability to accept either new modules (i.e. joints, links) or new applications (i.e. end-effector tips to accommodate various gripping/grasping strategies) without restoring from scratch.

3.2. Trajectory planning

The trajectory planning algorithm is divided in two main phases: the first one is dedicated to finding alternative trajectories to guide the robot to the final destination with the least error possible while the second phase tries to find amongst the feasible trajectories previously generated the best suited one with respect to an objective function.

The trajectory planning starts from the knowledge of the initial and goal destination coordinates assigned to a task. Generally, we can assume that there exist an infinite number of possibilities to cover the distance between the Cartesian coordinates of the pick and place points. This number is narrowed down by computing a set of feasible poses in the joint space of the robot and considering all their combinations for the path planning activity between the two Cartesian destinations mentioned above. An example of two alternative trajectories linking the same initial robot pose with two alternative poses at the goal destination is pictured in Figure 3.

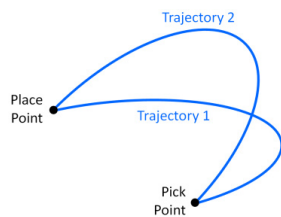


Figure 3. Example of alternative trajectories in Cartesian space

The sets of joint angle values associated to these poses are displayed in radians in Table 1.

Table 1. Joint angle values for robot poses at pick and place destinations

| T. | Pose | θ_1 | θ_2 | θ_3 | θ_4 | θ_5 | θ_6 |
|----|--------|------------|------------|------------|------------|------------|------------|
| 1 | @Pick | -2.89 | -1.57 | 0.24 | -3.14 | -1.34 | -0.24 |
| | @Place | 1.05 | -0.56 | -1.4 | 2.64 | -1.27 | 2.06 |
| 2 | @Pick | -2.89 | -1.57 | 0.24 | -3.14 | -1.34 | -0.24 |
| | @Place | 1.16 | -0.49 | -1.57 | -3.14 | -1.34 | 1.97 |

Two inverse kinematics techniques are employed sequentially to determine the corresponding sets of joint angle values. The main steps to determine the feasible poses at both initial and final destinations associated with a task are given as follows:

Algorithm.

INPUT: Specific DH matrix for robot configuration, Robot pose codes vector, pick and place coordinates in homogeneous matrix form and positioning accuracy vector
 OUTPUT: Feasible alternative robot poses at pick and place destinations.

```

1) For every pose code
  Call analytical inverse kinematics function to:
  Compute joint angles at pick destination;
  Compute joint angles at place destination;
  end for ;

2) For every set of joint angles
  Call numerical inverse kinematics function to:
  Re-compute angle values with software limits;
  end for ;

3) For every set of recalculated joint angles
  Call forward kinematics function to:
  Compute robot's TCP coordinates at pick and
  place destinations;
  Estimate the error between nominal coordinates
  and calculated TCP coordinates

  if error > accuracy vector
    discard angle values;
  else
    add values to joint angle vector;
  end if ;

  end for;
end.
```

First of all, the joint variables are solved analytically by a dedicated function which needs as input the robot kinematics, the coordinates of the start and end destination points specified in homogenous matrix form and a specific code describing a particular pose of the robot.

Each code can be described by pairing specific orientations (i.e. left/right, up/down and flipped/not flipped) to the robot configuration components, namely shoulder, elbow and wrist, exclusively in this order. For instance the pose having the shoulder on the left, the elbow up and the wrist not flipped can be identified with the code LUN. Following this procedure a total number of 8 robot poses can be identified for every point located in the Cartesian space.

The analytical computation of the joint angles, implemented following the method described in [14], can be considered as a first guess approach since it does not consider the joints' limit range. The first step to solving analytically the joint angles is to compute, through forward kinematics, the homogenous transformation T_6 , which describes the position and orientation of the end effector of the robot:

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6$$

For a serial robot arm, the A_i transformation describes the coordinate frame A with respect to a fixed coordinate frame $i-1$. The T_6 matrix product is evaluated from right to left by using the intermediate product matrices U_i , defined as follows: $U_6=A_6$, $U_5=A_5U_6$, ..., $U_1=A_1U_2$. For instance, A_6 can be expressed as a function of cosine and sine of θ_6 as follows:

$$A_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ s\theta_6 & c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, by pre-multiplying T_6 with each of the A_i transformations we will obtain a sequence of V_i matrices. In order to obtain the solutions for θ_i , where $i=1,2,\dots,6$, the V_i matrices are equated with the matrices U_{i+1} . An example is shown below:

$$T_6 = V_0 = U_1$$

$$A_1^{-1}T_6 = V_1 = U_2$$

In evaluating the V_i matrices we shall refer to its elements by three subscripts V_{ijk} , where i refers to the i th V matrix and j and k refer to the j th row and k th column respectively. The first analytical solution is obtained for θ_1 , the second one for θ_2 and so on until the final solution is found for θ_6 . For the sake of brevity, only the calculation details of the first three joint angles are depicted below.

Two solutions are possible for θ_1 and they correspond to the shoulder of the robot:

$$\theta_1 = \tan^{-1} \frac{p_y}{p_x} + \sin^{-1} \frac{d_3}{r}$$

$$\theta_1 = \tan^{-1} \frac{p_y}{p_x} + \pi + \sin^{-1} \frac{d_3}{r}$$

where d_3 is the characteristic distance of joint 3, $p_x=T_6(1,4)$ and $p_y=T_6(2,4)$ and :

$$r = \sqrt{p_x^2 + p_y^2}$$

They are assigned by evaluating the geometric constraint given by the first indicator of the pose code. For instance, if the indicator is R, denoting right-hand side mounting of the shoulder, then θ_1 is calculated with the first formula. If the shoulder is mounted on the left-hand side, specified by L indicator, the second formula is applied.

If the first solution is used for right-handed configuration of joint 1, the elbow of the arm is above the line from the shoulder to the wrist, whereas in the latter solution it is below this line. The above and below positions are reversed if the

configuration of joint 1 is left-handed. The solutions for θ_2 are derived from the following equations by taking into account the second indicator of the pose code (i. e. U for elbow up and D for elbow down):

$$\theta_2 = \tan^{-1} \frac{p_z}{V_{114}} + \psi$$

$$\theta_2 = \tan^{-1} \frac{p_z}{V_{114}} - \psi$$

$$\psi = \cos^{-1} \frac{a_2^2 - d_4^2 - a_3^2 + V_{114}^2 + p_z^2}{2a_2r}$$

$$V_{114} = p_x \cos \theta_1 + p_y \sin \theta_1$$

$$r = \sqrt{V_{114}^2 + p_z^2}$$

where $p_z=T_6(3,4)$, a_2 and a_3 are the lengths of link 2 and link 3 respectively and d_4 is the characteristic distance of the joint 4.

Once the angle value of joint 2 is known the angle θ_3 can be estimated with the following formula:

$$\theta_3 = \tan^{-1} \left(\frac{a_3}{d_4} \right) - \tan^{-1} \frac{\cos \theta_2 V_{114} + \sin \theta_2 p_z - a_2}{\cos \theta_2 p_z - \sin \theta_2 V_{114}}$$

The analytical computation of the joint angles is followed subsequently by a numerical inverse kinematics method, which is applied iteratively for every pose to re-evaluate the joint angle values by imposing limit constraints. These limits are provided by an external module receiving information related to the health status of the joints and providing decision support to constrain the motion on joints experiencing degrading performance. As a consequence all the unfeasible poses are filtered out at this stage and by considering all the combinations between the poses at start and goal destinations, a number of joint-space trajectories can be generated for a task. The trajectories in the joint space are represented as synchronized S-curves employing a set of trigonometric formulas adapted from [6], with nodes uniformly distributed along the time scale. The S-curve exhibit finite jerk spread over a period of time enabling the generation of smooth trajectories with faster settling times due to small residual vibrations [3,15]. The motion along these trajectories must be realized without violating bounds on the kinematic performance while considering upper velocity, acceleration and jerk limits for every joint. Amongst all feasible alternatives, the trajectory that covers the distance in the shortest time and with the smoothest jerk will be retained and converted in Cartesian coordinates with the support of a forward kinematics function. The mathematical formulation of

the optimal balance between the smoothness and time efficiency of a trajectory can be written as follows:

$$f = \min(k_T t_{hi} + k_J \sum_{j=1}^n \int_{t_0}^{t_f} j_j(t) dt), i = 1, 2, \dots, m$$

where k_T is the weighting factor for the execution time, k_J is the weighting factor for the jerk, t_{hi} is the synchronized motion time for trajectory h_i , t_0 and t_f are the initial and final motion times, $j_j(t)$ is the jerk function for the j th joint, n is the degree of freedom of the robot and m is the number of alternative trajectories.

3.3. Part program generation

The last step consists in the conversion of the Cartesian point coordinates defining the selected trajectory into a set of control points to be fed to the robot's CNC, which is realized by the B-spline generator module. The robot is controlled by a Beckhoff™ industrial PC which rebuilds the reference joint trajectories generated in MATLAB® as cubic spline functions. Each spline interpolates the set of control points, individually estimated for every reference trajectory and integrated into its associated part program. In order to cope with uncertainties related to robot dynamics and torque limits during the motion task execution, the nominal trajectories will be continuously adjusted by the online closed loop control of the robot.

4. Case studies

The proposed approach has been evaluated with respect to two use case scenarios. Their objective is to characterize the ability of the planner to automatically adjust the trajectory of the robot in response to various changes in the working environment such as the modification of the tasks' objectives, generation of new kinematic configurations or the change of the number and types of constraints applied to a specific configuration. Both scenarios refer to the same task whose pick and place Cartesian coordinates are $P_{ini} = [-400 \ -100 \ 300]$ and $P_{end} = [300 \ 700 \ 500]$. Figure 4 shows the robot picking a chip-on-carrier, a microelectronic component used in the assembly of laser diodes.

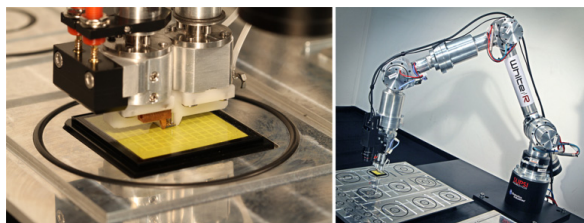


Figure 4. ReRob picking optoelectronic component (chip on carrier)

The first application of the method addresses the reconfigurability feature of the robot and aims to show how the planner will automatically adapt a trajectory in response to a change in the kinematic configuration. By simply changing the end effector tip or replacing an existing modular link of

the robot with a shorter version of itself we obtain a robot with altered kinematic capabilities which will be referred to as the new configuration. The first step consists in the re-evaluation of the DH parameters followed by the calculation of the transformation matrices for each joint and subsequently the determination of the location of the TCP through the forward kinematics. For the sake of clarity, we will consider that both robotic arms, before and after reconfiguration, are employed to perform the same task to cover the distance between the same pick and place Cartesian coordinates as mentioned above.

Table 2. Robot configurations

| Configuration | Link1 | Link2 | Link3 | Link4 | Link5 |
|---------------|-------|-------|-------|-------|-------|
| C1 | 86 | 420 | 253 | 150 | 195 |
| C2 | 86 | 290 | 253 | 150 | 195 |

The forward kinematics of the two configurations is computed with regard to the geometric description of the links, as depicted in Table 2. As can be seen in Figure 5, the length reduction of the link 2 altered not only the shape and spatial disposition of the trajectory but also the robot poses at the start, end and all along the generated trajectory.

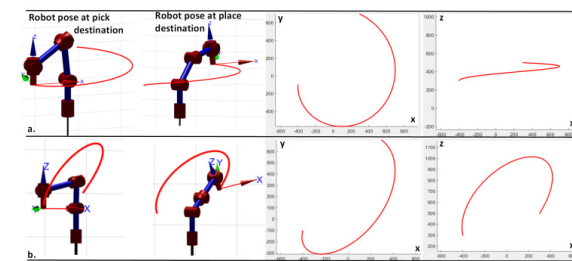


Figure 5. Robot pose at pick /place destinations and trajectory for configuration (a) C1 ; (b) C2

A second example of the automated trajectory planning is inspired by the need to adapt the motion behavior of the joints experiencing a faster degradation process. In this case the utility of the planner consists in the regeneration of existing trajectory alternatives without changing the robot kinematics. The regeneration of the trajectory is triggered by a reduction in the software limits of the joints resulted from a constant monitoring activity of the health status of the joints, which raises alerts when their predefined thresholds are exceeded.

Table 3. Joints' motion range

| Joints | J1 | J2 | J3 | J4 | J5 | J6 |
|--------|------------|-----------|-----------|------------|----------|------------|
| Case1 | [-180,180] | [-120,30] | [-210,30] | [-180,180] | [-90,45] | [-180,180] |
| Case2 | [-180,180] | [-90,30] | [-210,30] | [-180,180] | [-90,45] | [-180,180] |

For illustration purposes we consider the robot configuration C1 as described in Table 2 and two sets of joint angle values as described in Table 3. In the second case the motion range of the joint 2 features a reduction of 30 deg. with respect to its counterpart. A few alternative trajectories are computed successively for each case and represented altogether in Figure 6.

The reduction of the software limits associated with the joint 2 makes the previous set of trajectories (i.e. in red in

Figure 6a) unfeasible and the planner will automatically adapt the motion of the robot between the start and goal destination in order to cope with the new operational constraints.

The trajectory that is best suited to map the requirements of the objective function is selected and converted into a set of points that will be used in the generation of the part program associated to each task. These points are employed by the CNC cubic spline algorithm as control points for trajectory interpolation. In this regard, we have computed the control points for one of the trajectories generated with reduced software limits (i.e. blue trajectory no.1 in Figure 6a) and represented both in Figure 6b. Prior to the execution of a task the robot's controller loads from the database the associated part program integrating the interpolation points as a sequence of G00 and G01 blocks.

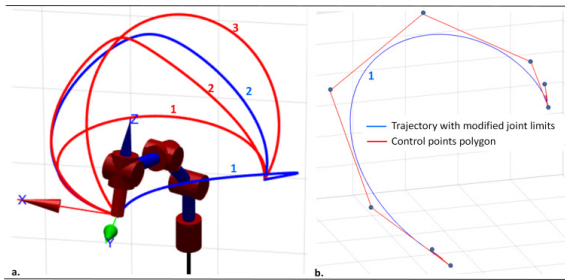


Figure 6. a) Alternative trajectories for the configuration 1 of the robot with nominal and modified joint limits, b) trajectory with control points polygon

5. Conclusions and future work

This paper presented an automated trajectory planning of a reconfigurable anthropomorphous arm based on given description of start and goal target coordinates of a task and a jerk-bounded trigonometric S-curve trajectory modeling approach to ensure smooth motion in the robot's workspace.

This planner is of particular value in the reconfigurable field of robotics, being capable to map specific trajectories to particular kinematic configurations, task objectives and operational constraints and to facilitate their interpolation by robot's CNC without any software or hardware change. The approach results in being enabler towards a fully reconfigurable control.

Although the planner shows good results it can produce overly direct trajectories which can load excessively specific joints. Future work will address the enhancement of the trajectory planner with a customizable weighting scheme enabling a higher degree of flexibility in the distribution of the motion across the joints.

Acknowledgements

The research has been partially funded by FP7 European Project entitled white'R- whiteroom based on Reconfigurable robotic island for optoelectronics. Project Number 609228.

References

- [1] Yim M, Shen W, Salemi B, Rus D, Moll M, Lipson H, Klavins E, Chirikjian G. Modular self-reconfigurable robot systems. *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43-52, 2007.
- [2] Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G, Van Brussel H. Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology*, vol. 48, no. 2, pp. 527-540, 1999.
- [3] Kim J, Kim SR, Kim SJ and Kim DH. A practical approach for minimum-time trajectory planning for industrial robots. *Industrial Robot: An International Journal*, vol. 37, pp. 51-61, 2010.
- [4] Constantinescu D and Croft EA. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of Robotic Systems*, vol. 17, pp. 233-249, May 2000.
- [5] Antonelli G, Chiaverini S, Gerio P, Palladino M and Renga G. SmartMove4: an industrial implementation of trajectory planning for robots. *Industrial Robot: An International Journal*, vol. 34, pp. 217-224, 2007.
- [6] Perumaal S and Jawahar N. Synchronized trigonometric S-curve trajectory for jerk-bounded time-optimal pick and place operation. *International Journal of Robotics & Automation* 2012, vol. 27, no. 4, p. 385-395, DOI:10.2316/Journal.206.2012.4.206-3780
- [7] Macfarlane S and Croft EA. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 42-52, Feb. 2003.
- [8] Liu S. An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In: *Proc. of the 7th International Workshop on Advanced Motion Control*, pp. 365-370, 2002.
- [9] Nguyen KD, Ng T and Chen I. On Algorithms for Planning S-curve Motion Profiles. *International Journal of Advanced Robotic Systems*, vol. 5, pp. 99-106, 2008.
- [10] Gasparetto A, Lanzutti A, Vidoni R and Zanotto V. Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. In: *Robotics and Computer-Integrated Manufacturing* 28 (2012) 164-181.
- [11] Valente A. Reconfigurable Industrial Robots - An Integrated Approach to Design the Joint and Link Modules and Configure the Robot Manipulator. *Advances in Reconfigurable Mechanisms and Robots II*, vol. 36, pp. 779-794, doi:10.1007/978-3-319-23327-7_67
- [12] Corke PI. *Robotics, Vision & Control*. Springer 2011, ISBN 978-3-642-20143-1
- [13] Avram O and Valente A. A regenerative approach to dynamically design the forward kinematics for an industrial reconfigurable anthropomorphic robotic arm, submitted to *Smart Innovation, Systems and Technologies*, ISSN: 2190-3018
- [14] Paul RP and Zhang H. Computationally Efficient Kinematics for Manipulators with Spherical Wrists based on the Homogeneous Representation, *The International Journal of Robotics Research* 1986, 5:32, doi: 10.1177/027836498600500204
- [15] Kroger T and Wahl FM. Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events. In: *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94-111, Feb. 2010 doi: 10.1109/TRO.2009.2035744