

^{1,2}Rabab BENOTSMANE, ¹Seyf Eddin KACEMI, ¹Mohammed REDHABENACHENHOU

CALCULATION METHODOLOGY FOR TRAJECTORY PLANNING OF A 6 AXIS MANIPULATOR ARM

¹Department of Automatic, Faculty of Electrical Engineering, University of Mohamed Boudiaf, Oran, ALGERIA²Institute of Information Science, Faculty Mechanical Engineering & Informatics, University of Miskolc, Miskolc, HUNGARY

Abstract: In this paper, we summarize the basic methodology for planning and generating trajectories, to validate our own calculation from a real model of a 6-axis manipulator arm, according to an algorithm and a polynomial calculation using MATLAB as a simulation tool. In the way we'll see a brief introduction about trajectory planning by mentioning the methods existing in this field. Then we will clarify the explanation of the fundamental principles of trajectory generation for manipulator arms by describe both approaches in joint and operational space. Next, we will see exactly how to present this application on a manipulator arm with 6 degrees of freedom by simulating the behavior of the position and velocity and acceleration for each joint.

Keywords: Trajectory Planning, 6-axis manipulator Arm, Polynomial computation, MATLAB

1. INTRODUCTION

In the field of industrial robotics, manipulator robots have taken a very wide space of use. These robots are typically used in demanding and repetitive tasks required for many applications in various industries such as automotive industry, nuclear industry and for handling heavy object and precision assembly etc. [1]. They present an essential point in the evolution of the industry of future known as industry 4.0 [2].

The first robots used for pick-and-place applications were SCARA robots (Selective Compliant Assembly Robot Arms). Those robots are used for their architectural simplicity which is of series type, characterized by the fact of working at low velocities and accelerations, which depend directly on the cycle time. These characteristics represent an important point in the study of generating the trajectory to plan well defined paths which a path is the place of the points in the joint space, or in the operational space that the manipulator must follow in the execution of the assigned movement. It is a pure geometric description of movement to specify the cycle time of velocities and accelerations [3].

The purpose of trajectory generation is to design the reference inputs of the motion control system which ensures that the manipulator executes the intended trajectory while ensuring significant results in terms of quality of control.

Due to the general problem of trajectory planning, to ensure significant results in terms of quality control and easy execution of the movement required for industrial robots, in this article, we will explain the fundamentals of trajectory generation for manipulator arms by describe both approaches in joint and operational space. Then, we will present this application on a manipulator arm with 6 degrees of freedom by simulating the behavior of the position and velocity and acceleration for each joint.

2. BACKGROUND KNOWLEDGE IN ROBOTICS FIELD

A manipulator arm is composed of serial rigid links which are fixed to each other by revolute joints that allow relative motion from the base frame through the end-effector [4, 5]; those joints are usually instrumented with position sensors which allow the relative position of neighboring links to be measured.

We generally describe the position of the manipulator arm by giving a description of the tool frame [6], which is attached to the end effector. In this context we can find two descriptions [7] on figure 1:

- The joint space: It's the description of the motion to be made by the robot by its joint values.
- The operational space: known as Cartesian space, in which the position of a point is given with three numbers, and the orientation of a body is given with three numbers also.

The following figure 2 shows the transition between the joint space and the operational space by the direct and inverse kinematics.

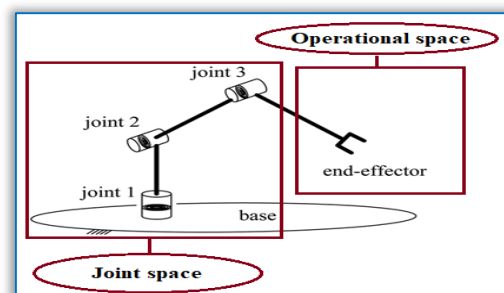


Figure 1. Description of joint and operational space of manipulator arm

General method for trajectory planning

The problem of the motion generation is to compute the reference setpoints of position, velocity, acceleration [8], which are presented as a function of time by ensuring the transition of robot by a desired trajectory, defined by a sequence of location of the end effector or joint configurations (called points) formed the path of robot as shown on figure 3, these points can be [9, 10]:

- program by learning,
 - given online by an exteroceptive sensor,
 - comes from a database of a CAD system [11].
- Regarding of pure displacement, we can distinguish the following classes of movement [12, 13]:

» In joint space:

- The movement between two points with free trajectory among the points.
- The movement between two points via intermediate points. Specified to avoid obstacles, with free trajectory along the intermediate points.

» In operational space:

- The movement between two points with constrained trajectory amongst points. "Rectilinear trajectory for example".
- The movement between two points via intermediate points with trajectory constraint betwixt the intermediate points.

Motion planning and control of the robot system

Both approaches to trajectory planning in joint space and the operational space are schematized on figures 4 and 5 [14-16].

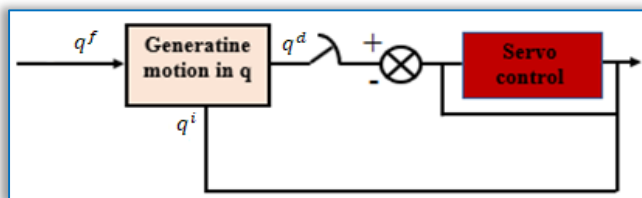


Figure 4. Trajectory planning in the joint space [9]

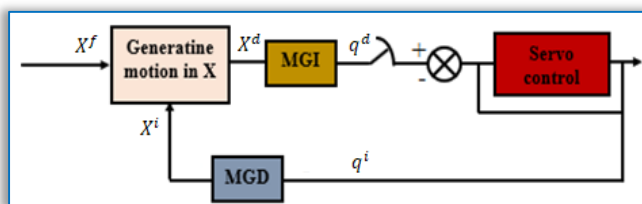


Figure 5. Trajectory planning in the operational space [9] where: MGI: inverse geometric model, MGD: direct geometric model.

Planning of movement between two points with free trajectory among points. We consider our robot of 6 degrees of freedom, we assume that q^i and q^f are joint coordinates vectors corresponding to the initial and final configurations. We denote respectively by K_{vj} and K_{aj} the maximum joint vectors for velocities and accelerations. K_{vj} Parameters are usually calculated from actuator characteristics and reduction ratios transmission organs while K_{aj} parameters are approximated by the ratio of maximum torque to maximum inertia seen by the joints (upper bound of the A_{jj} terms defined in the dynamic model). Once computed motion with these kinematic constraints, then we can proceed to a change of scale to satisfy the maximum torque constraints obtained with the dynamic model [9, 10].

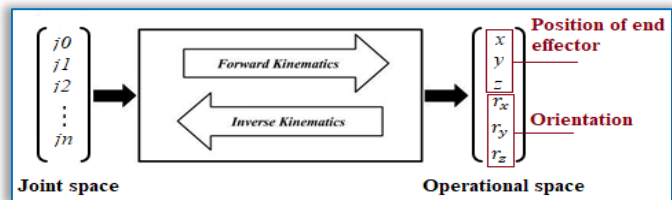


Figure 2. The relation between the joint space and operational space, where: j: number of joint in robot.

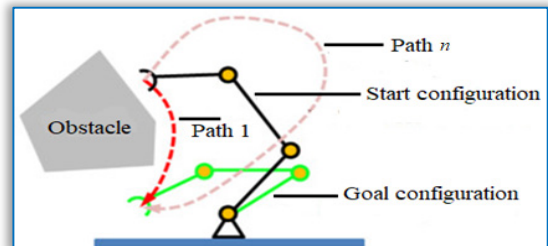


Figure 3. Path planning of manipulator arm [13]

Choosing a motion planning method depends on the application considered. Each approach has its own limits; inherent in the fact that constraints are expressed either in the joint space (velocity, torque, workspace limits) or be in the operational space (precision, obstacle) as shown on figure 6, considering these remarks we discuss in our study the generation of motion between two points with free trajectory amongst the points.

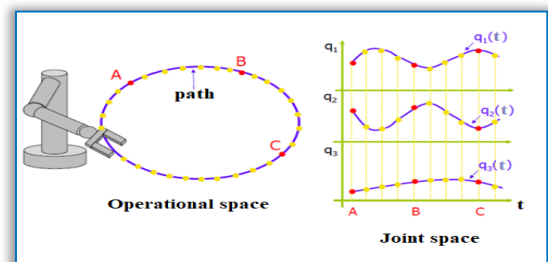


Figure 6. Generating motion in operational space and joint space

The movement $q(t)$ between q^i and q^f as a function of time t is described by the following equations:

$$q(t) = q^i(t) + r(t) \cdot D \quad [\text{rad}] \quad (1)$$

$$\dot{q}(t) = \dot{r}(t) \cdot D \quad [\text{rad/sec}] \quad (2)$$

where: q^i : Joint coordinates vectors to the initial configuration, $\dot{q}(t)$: Velocity function, $r(t)$: Interpolation function, D : Duration of displacement.

The limit values of the interpolation function $r(t)$ are given by: $r(0) = 0$ and $r(t_f) = 1$.

The expression (1) is also written:

$$q(t) = q^f(t) - [1 - r(t)] \cdot D \quad [\text{rad}] \quad (3)$$

Formulation that is appropriate in the case of target tracking, when the term q^f varies [15], in this case:

$$D = q^f(t_f) - q^i(0) \quad [\text{rad}] \quad (4)$$

Several functions make it possible to satisfy the change by q^i at $t = 0$ and by q^f at $t = t_f$ as the polynomial interpolation, the bang-bang law, the trapezoid law, etc.

In our case we are limited to the study of polynomial interpolation.

Polynomial interpolation for generating trajectory of robot arm

The most frequently encountered polynomial interpolation modes are linear interpolation and interpolation by polynomials of degrees three and five [12], the instructions generated for our manipulator arm are polynomial of degree five [12].

3. THE FIVE-DEGREE POLYNOMIAL METHOD

For high velocity robots or carrying important loads, it is necessary to ensure the continuity of accelerations to avoid mechanical problems. It is said that the function of movement is class C^2 [18]. Constraints must be satisfied, and the interpolation polynomial must be of degree five, with the additional constraints:

$$\begin{cases} \ddot{q}(0) = 0 \\ \ddot{q}(t_f) = 0 \end{cases} \quad (5)$$

where: \ddot{q} : Acceleration in the joint space, t_f : Time taken by the constrained joints.

It is shown that the position function can be in the form (1) or (3) with:

$$r(t) = 10 \cdot \left(\frac{t}{t_f}\right)^3 - 15 \cdot \left(\frac{t}{t_f}\right)^4 + 6 \cdot \left(\frac{t}{t_f}\right)^5 \quad (6)$$

The change of position, velocity and acceleration for the joint j are presented in figure7, the maximum velocities and accelerations have for expressions (9-10):

$$|\dot{q}_j \max| = 15|D_j| / (8 \cdot t_f) \quad [\text{rad/sec}] \quad (7)$$

$$|\ddot{q}_j \max| = 10|D_j| / (\sqrt{3} \cdot t_f^2) \quad [\text{rad/sec}^2] \quad (8)$$

where: \dot{q}_j : Velocity of the joint j in the joint space, \ddot{q}_j :

Acceleration of the joint j in the joint space, D_j :

Duration of a joint displacement.

where: T : represents t_f in figure7.

— Calculation of the minimum time of a joint

The minimum time for a joint j must ensure velocity and/or acceleration. From the expressions of the maximum velocities and acceleration obtained with the different interpolation methods presented, we deduce the minimum time t_{fj} that is necessary for a joint j to achieve a given displacement.

where: t_{fj} : The minimum time of joint j .

— Joint coordination for generating trajectory

The search for joint coordination is to synchronize the movements for that all the joints leave and arrive at the same time. The duration of a joint displacement is a function of the distance D_j to be traveled as well the gains of velocities and acceleration K_{vj} and K_{aj} for that it is not identical from one joint to another, The minimum total time t_f is the time taken by the constraining joint, joint for which the minimum time is the bigger.

$$t_f = \text{Max}(t_{f1}, \dots, t_{f6}) \quad [\text{sec}] \quad (8)$$

The minimum time for a joint j is calculated by:

$$t_{fj} = \text{Max} \left[15 \cdot |D_j| / (8 \cdot K_{vj}), \sqrt{10 \cdot |D_j| / (\sqrt{3} \cdot K_{aj}^2)} \right] \quad [\text{sec}] \quad (9)$$

where: K_{vj} : Maximum joint vectors for velocities, K_{aj} : Maximum joint vectors for accelerations.

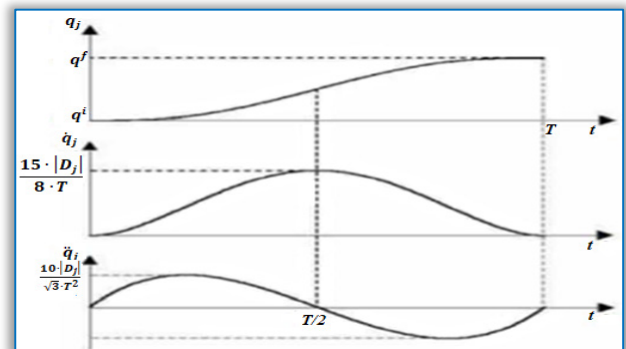


Figure 7. Changes in position, velocity and acceleration for joint j [9]

Elaborated calculation methodology for generating trajectory parameters of 6-axis manipulator arm

This part was taken from our master's memory which was to build a 6 axis manipulator arm, to realize this type of robot, it is necessary to go through several parts starting with the modeling of the system in order to calculate the geometry and the kinematics the models Dynamics then study the mechanical engineering using the software SOLIDWORK "computer-assisted design" to determine the mechanisms used in each joint after we focus on the electrical study by identifying all the necessary electronic component in this robot, finally we are based on the control part of the system; control this type of manipulator arm; it is first necessary to initialize the desired entries known by the trajectory of the robot this trajectory gathers the positions and the velocities and accelerations that the robot must reach on its way; The next part of this article explain the method of calculating the parameters needed to plot the trajectory of a 6-axis manipulator arm.

Our goal in this paper is to show how we can compute these parameters from our experience, by giving a real example and simulate it to get the results as graphs in Simulink to compare the theoretical and the experimental study.

Validation methods of results

The knowledge of the chosen method mentioned before as well as the previous equations helped us to generate the desired movement for each joint by the following application.

We initialize for each joint the following inputs:

The configurations of the initial and final vector element q^i and q^f are made by our choice, in our simulation we took the following inputs data shown in Table 1 and Table 2.

Table 1. Initialization of the initial and final positions for the six joints

Joint j	The initial position q^i_j for joint j [rad]	The final position q^f_j for joint j [rad]
1	0	$\frac{\pi}{2}$
2	π	0
3	$\frac{\pi}{6}$	π
4	0	$\frac{\pi}{2}$
5	π	2π
6	0	2π

where: q^i_j : Initial position in the joint space j, q^f_j : Final position in the joint space j.

Table 2. Initialization of Input data to calculate the gain vectors K_{vj} and K_{aj}

Joint j	The maximum velocity of motor ω [rad/s]	The ratio of transmission n	The motor torque τ_m [Nm]	The maximum inertia J of the joint $j = A_{jj}$
1	460.83	0.31	$\tau_{m1} = 6$	$J_1 = A_{11}$
2	460.83	0.19	$\tau_{m2} = 6$	$J_2 = A_{22}$
3	460.83	0.25	$\tau_{m3} = 6$	$J_3 = A_{33}$
4	460.83	0.20	$\tau_{m4} = 6$	$J_4 = A_{44}$
5	370.37	0.40	$\tau_{m5} = 2$	$J_5 = A_{55}$
6	370.37	0.26	$\tau_{m6} = 2$	$J_6 = A_{66}$

where: ω : Maximum velocity of motor in joint j, n: Ratio of transmission, τ_{mj} : Motor torques in joint j, J: Maximum inertia in joint j, A_{jj} : defined in the dynamic model of our 6-axis manipulator arm where \underline{A} is the matrix of inertia.

The maximum velocity and acceleration vector elements K_{vj} and K_{aj} will be obtained by the following calculations:

1.) For the velocity gain K_{vj} which represents the maximum velocity ω_{joint} of the joint j:

We take the maximum velocity of motor ω_{motor} of joint j and the ratio of transmission n, and we apply the following relation taken from the ratio of the gears:

$$\frac{\omega_{joint}}{\omega_{motor}} = n \quad \text{So, we have: } k_{v1} = \omega_{joint} = \omega_{motor} \cdot n \quad [\text{rad/sec}] \quad (10)$$

2.) For the acceleration gain K_{aj} which represents an acceleration δ .

We have the following relation between it and the motor torque τ_m :

$$\tau_m = J \cdot \delta \quad [\text{Nm}] \quad \text{So: } \delta = (\tau_m) / J \quad [\text{rad/sec}^2] \quad (11)$$

Note: Maximum engine velocities and torques are obtained by their datasheets.

Numerical application using inputs data

Case study of the maximum velocity gain K_{vj} for six joints

Table 3 presents the result of the velocity gains K_{vj} for six joints using the inputs data of maximum velocities of motors $\omega_{joint j}$ and the ratio of transmission n.

Case study of the maximum acceleration gain K_{aj} for 6 joints

Table 4 presents the results of the acceleration gains K_{aj} for six joints using the inputs data of motor torque τ_m for six engines and the maximum inertia J of the joint j which represent A_{jj} .

4. RESULTS OF SIMULATION TOOL USING MATLAB SOFTWARE

After the computation of velocity and acceleration gains, we pass to write the program on MATLAB [19] which used to generate the trajectory of our manipulator arm for each joint j . We run this program in the editor file, we put the result on function blocks which we have six motion generator blocks on Simulink diagram. The figure 8 presents a block of one motion generator for a joint j .

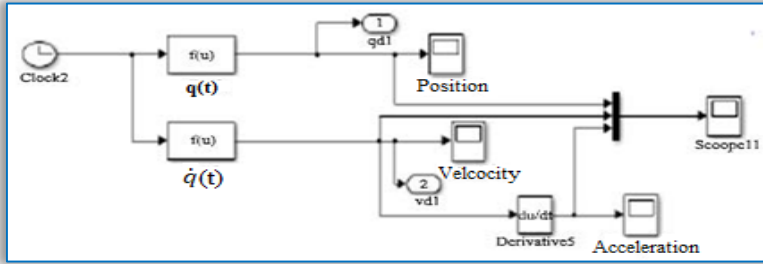


Figure 8. Diagram of a motion generator for a joint 1

The results of simulation in MATLAB are presented in the following figure 9. The first curve presents the desired positions in starting from the initial to the final in which one joint from robot should take, in other hand the second one present the curve of velocity and the third one the curve of acceleration to realize a good trajectory.

Comparative analysis between results of theoretical and experimental part of trajectory planning

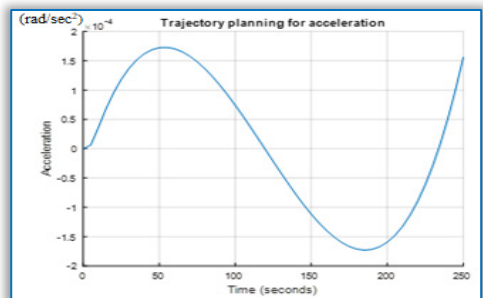
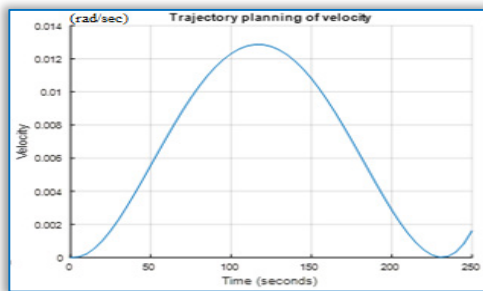
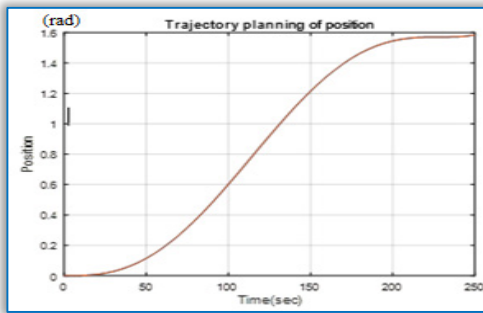


Figure 9. Curves of the desired trajectory for a joint 1

Table 3. Vector of gains K_{vj} for the six joints

Joint j	Calculation of gains K_{vj} [rad/s]
1	$K_{v1} = 142.85$
2	$K_{v2} = 87.55$
3	$K_{v3} = 115.2$
4	$K_{v4} = 115.2$
5	$K_{v5} = 148.14$
6	$K_{v6} = 85.18$

Table 4. Vector of gains K_{aj} for the six joints

Joint j	Calculation of gains K_{aj} [rad/s ²]
1	$K_{a1} = \delta_1 = 6/A_{11}$
2	$K_{a2} = \delta_2 = 6/A_{22}$
3	$K_{a3} = \delta_3 = 6/A_{33}$
4	$K_{a4} = \delta_4 = 6/A_{44}$
5	$K_{a5} = \delta_5 = 6/A_{55}$
6	$K_{a6} = \delta_6 = 6/A_{66}$

Note: The calculation of the inertia matrix is done in real time

In order to validate our calculations and results obtained by the simulation tool, we must make a comparative study which needs to determine the contaminated error between the graph of the theoretical part and the simulated one, this error must be negligible to improve our results, in our case we analyze a joint number 1 as we mentioned before, so the evaluation of this error requires to calculation of the boundary conditions presented in figure 10 of graph of the theoretical part and compare it with the measurements obtained in the graph of experimental part.

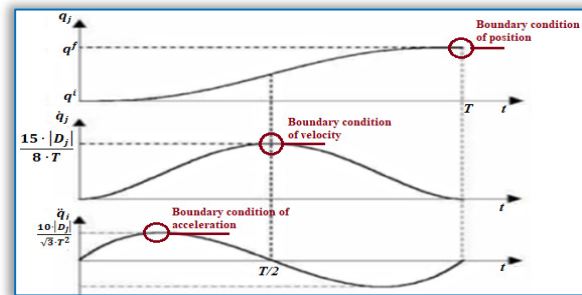


Figure 10. Boundary conditions of position, velocity and acceleration for joint j [9]

According to the graphs of figure 10, we have three boundary conditions for position BCP, velocity BCV; acceleration BCA presented by q^i, q^f, D_j, T , the following Table 5 presents the reel values obtained by calculation for this boundary conditions. The Table 6 presents the results in theoretical part and experimental part for the first joint.

By the simulation result, we can denote that we get the same graphs mentioned in the theoretical part based on polynomial of five-degree method figure 7 and after the comparison by calculation of each boundary condition we found the error

between the theoretical and the experimental values is negligible, which proves that our calculation methodology and simulation (the schematization of blocks on MATLAB Simulink) are completely validated.

Table 5. Calculation of boundary conditions for position, velocity, acceleration

Joint 1		Calculation of boundary conditions	
Position	As we mention earlier that q^i and q^f is taken by our choice, we have $q^i = 0$ and $q^f = \frac{\pi}{2}$ as theoretical values.		
Velocity	Boundary condition of velocity in theoretical part defined by the following expression: $\frac{15 \cdot D_j }{8 \cdot T}$ where $D_j = q^f(t^f) - q^i(0)$ and T is cycle time $T = 230$ [sec].		
Acceleration	Boundary condition of acceleration defined by the following expression: $\frac{10 \cdot D_j }{\sqrt{3} \cdot T^2}$.		

Table 6. Comparison between the theoretical and experimental results

Theoretical results			Experimental result		
Position	Velocity	Acceleration	Position	Velocity	Acceleration
$q^i = 0$ $q^f = \frac{\pi}{2}$ [rad]	$\frac{15 \cdot D_j }{8 \cdot T} = 0,012$ [rad/s]	$\frac{10 \cdot D_j }{\sqrt{3} \cdot T^2} = 1,7 \cdot 10^{-4}$ [rad/s ²]	BCP = 1,58 [rad]	BCV = 0,0129 [rad/s]	BCA = 1,8 · 10 ⁻⁴ [rad/s ²]

5. CONCLUSION

Generally a robot acts on its environment through its movements due to its ability to plan its trajectory which is an essential point of its autonomy, generating trajectory presents a field of research very vast in robotics, it is the fundamental tool for creating correct instructions defined as the inputs of system in order to can control it, it consists of planning a well-defined motion in the joint or operational space takes into consideration the possible positions to be traversed by the robot as well its velocities and accelerations to each path.

In this article, we have presented a brief introduction about trajectory planning by mentioning the methods existing in this field. Then we talked about its application in joint space by basing on planning of movement between two points with free trajectory among points which presents the one of the simplest applications that performs the calculation of certain parameters finally we illustrate a real example of the methodology necessary to generate the trajectory of a 6-axis manipulator arm by simulating the behavior of results obtained in MATLAB and estimating the calculated error and improving that our methodology for calculation is completely true.

References

- [1] Chemori, A.: Generation de trajectoire et commande avancee de robot paralleles rapides et precis: Application au robot VELOCE, LIRMM « Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, http://arrow.irccyn.ec-nantes.fr/Documents/LIRMM_M2_11-12_No1.pdf, Accessed: 5/01/2018
- [2] Kovács, Gy., Kot, S.: Facility layout redesign for efficiency improvement and cost reduction, Journal of Applied Mathematics and Computational Mechanics, 2017. 16 (1), 63-74
- [3] Vaclav, H.: Manipulating trajectory planning. https://cw.fel.cvut.cz/old/_media/courses/a3m33iro/048robotworldrepresforreasoningen.pdf, Accessed: 19/03/2018.
- [4] Kumar, V.: Introduction to robot geometry and kinematics. <https://www.seas.upenn.edu/~meam520/notes02/IntroRobotKinematics5.pdf>, Accessed: 08/12/2017.
- [5] Craig, J. J.: Introduction to robotics: mechanics and control, Pearson/Prentice Hall, 2005.
- [6] Serdar, K., Zafer, B.: Robot kinematics: forward and inverse kinematics, https://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/robot_kinematics__forward_and_inverse_kinematics, Accessed: 07/12/2017.
- [7] Saeed, B. N.: Introduction to robotics: analysis. control. applications, John Wiley & Sons, 2010.
- [8] Motion planning, generating a path via IK in the cartesian space, <http://www.coppeliarobotics.com/helpFiles/en/oldMotionPlanningModule.htm>, Accessed: 25/12/2017.
- [9] Wissama, K., Etienne, D.: Modélisation identification et commande des robots, Hermes /Lavoisier, 1999.
- [10] Philippe, C.: Modélisation et commande des robots, Hermes, 1986.
- [11] Kunwoo, L.: Principles of CAD/CAM/CAE systems, Boston, Addison Wesley Longman, 1999.
- [12] Paul, R.: Robot manipulators: mathematics, programming and control, MIT Press, 1981.
- [13] Siciliano, B., Sciavicco, L., Villani, L.: Robotics: Modelling, planning and control, Springer Publishing Company, 2008.
- [14] Zhao, R.: Trajectory planning and control for robot manipulations, Robotics [cs.RO], Université Paul Sabatier - Toulouse III, 2015, <https://tel.archives-ouvertes.fr/tel-01285383v2/document>, Accessed: 28/12/2017.
- [15] Siciliano, B., Khatib, O.: Springer handbook of robotics, Springer, 2009.
- [16] Alessandro, D. L.: Trajectory planning in Cartesian space, http://www.diag.uniroma1.it/~deluca/rob1_en/14_TrajectoryPlanningCartesian.pdf, Accessed: 16/02/2018.
- [17] Taylor, R. H.: Planning and execution of straight line manipulator trajectories, IBM Journal of Research and Development, 1979. 23 (4), 424-436.
- [18] Cartan, H.: Cours de calcul différentiel, refondue et corr, Paris, Hermann, 1977.
- [19] Chapman, S. J.: MATLAB programming for engineers, Thomson, 2004.