The 4th Annual IEEE International Conference on
Cyber Technology in Automation, Control and Intelligent Systems
June 4-7, 2014, Hong Kong, China

Optimization Algorithm of Manipulator Robot Performances

Abderrahmane BELAIDI
Faculty of chemistry and hydrocarbons,
University M'hamed Bougara of Boumerdes,
35000, Algeria
e-mail: abd_belaidi@yahoo.fr

Hadjira BELAIDI
Signals and systems laboratory, IGEE, UMBB University,
Boumerdes 35000, Algeria.
Advanced Technologies Development Centre (CDTA)
BP 17, Baba Hassen, Algiers 16303, Algeria.
e-mail : hadjira983@yahoo.fr

***Abstract*— One aim of controlling a manipulator robot is to maximize its performances such as accuracy, speed, time etc…. However, limiting the power of actuators causes a limitation of their generalized accelerations and velocities; this is due to the high inertial forces, which create dangerous voltages at the machines elements.**

This paper propose an algorithm to solve the problem of maximizing the manipulator performances considering that the end-effector position and orientation is characterized by a 6x1 vector i.e., six degrees of freedom; moreover, the need to move the end-effector from the start position to the target with a minimum time, without violating its boundaries.

The presented solution is well suited to this context. It is optimal with respect to time constraints, and it allows a direct calculation.

Keywords*— *manipulator robot; optimization algorithm; speed, acceleration ; minimal time.

I. INTRODUCTION

One aim of controlling a manipulator robot is to maximize its performances (accuracy and speed). However, limiting the power of actuators causes a limitation of their generalized accelerations $\ddot{q}_i$ (1) and their generalized velocities $\dot{q}_i$ (2); this is due to the high inertial forces, which create dangerous voltages at the machines elements, and very high friction forces (bearings wear) that actuators cannot support.

$$|\ddot{q}_i| \le \ddot{q}_{i\,\max} \tag{1}$$

$$|\dot{q}_i| \le \dot{q}_{i\,\max} \tag{2}$$

In 2D coordinate system $(\dot{q}_i,\ \ddot{q}_i)$, we have a permissible range of complex form, where we can satisfy the above inequalities as shown in Fig. 1.

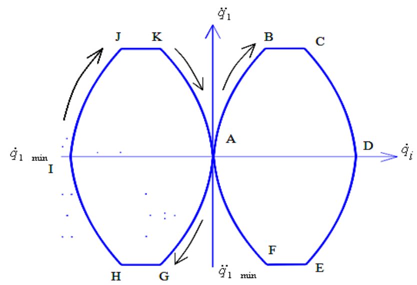

Fig. 1. Description of the speed, acceleration relationship.

The papers [1, 2, 3] present some essential works to determine the relative positions of the robot manipulator links. These positions are used to develop the algorithm and the optimization program of the motion.

Several works have addressed the optimization problem such as [4, 5, 6]. However, this problem is far from being exhausted due to the different realizations of mechanical system models. Among the problems in this area we have: the calculation of geometric models of facilities and technological equipment with numerical control (CNC), the possibilities of hold-put of pieces, the use of kinematic and dynamic models of manipulator robots used in these facilities and equipment, the mobile robotics equipment etc..

Posing the problem of maximizing the manipulator performances considering that the end-effector position and orientation is characterized by a 6x1 vector i.e., six degrees of freedom (three for positioning and three for the orientation); moreover, the need to move the end-effector from the start point to the target with a minimum time, without violating the limits. At boundary conditions (start and target position), the speed of the manipulator robot links is zero.

The formulation of the problem in this way is very difficult to solve, because virtually the manipulator admits limitations on the generalized coordinates q_i (we cannot get all the positions of the end-effector). To facilitate the task we assume that except for the start and target positions any position can be achieved.

I. OPTIMAL MOTION OF A MANIPULATOR ROBOT IN STATIC ENVIRONMENT

The accurate data for determining the optimal motions of a manipulator are:

The accuracy of the permissible positioning $[\Delta P]$;

- The boundary positions of the joints;
- The initial and target positions of the manipulator movements;

- The transition matrices, defining the layout of equipment according to the relative coordinate systems of the manipulators;
- The matrices giving the positions of pieces according to the equipment;
- The possible cases of the motion vector direction approaches in the vicinity of the workspace and of loading and unloading;
- The boundary points belonging to the contours of geometrical shapes, of trajectories, and the boundaries on the generalized coordinates of the robot manipulator;
- The kinematic or dynamic model of the manipulator;

The problem is to find the positions of the crossing points constituting the trajectory, relative to the n^{th} workspace of the manipulator. Solving this problem is done by the computer-aided design and modeling of the manipulator movement based on the kinematic or dynamic model of the mechanical structure.

The crossing points of the trajectory corresponding to the minimum time to describe the trajectory of the robot manipulator are chosen based on the calculations of the overall crossing time corresponding to different grid points (the grid built in the manipulator workspace from the start to the end point).

The main theoretical and methodological steps, used during the construction of the algorithm are:
We assign to each set up equipment a coordinate system $o_{ei}x_{ei}y_{ei}z_{ei}$ and for the robot system $o_rx_ry_rz_r$ (see Fig. 2).
$o_{ei}x_{ei}y_{ei}z_{ei}$: Coordinate system corresponding to the i^{th} equipment.

For the transition between systems we used the following 4X4 matrices.
M_{rp}: The transition matrix from piece system to robot system,
M_{rm}: The transition matrix from the machine system relative to that of the robot,
M_{mp}: The transition matrix from the piece system relative to that of the machine,

$$M_{rp}=M_{rm}M_{mp}, \tag{3}$$

The position of the piece in the coordinate system of the robot end-effector is:

$$M_{rp} = M_{01}M_{12}...M_{n-1,n}M_{Op}. \tag{4}$$

The coordinates of any point of a manipulator link is given by:

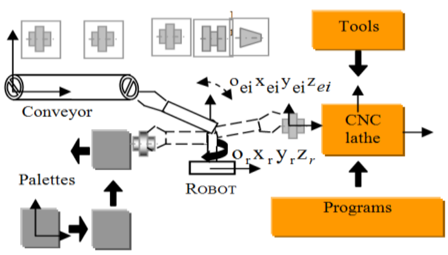

Fig. 2. The configuration of the cell and of the coordinate systems.

$$[x_0]=M_{01}M_{12}...M_{k-1,n}[x_k]; \tag{5}$$

where: $M_{k-1,p}$ the transition matrix of the p^{th} coordinate system to the $(k-1)^{th}$ system..

The environment of the robotized site is given as a set of points in space, limited by planes or surfaces of the second degree. [7]

The approach vector which describes the movement direction of the piece, during the loading and unloading of the machine, is given in the form of a set of points, arranged in line segments.

The figure below (Fig.3) describes the algorithm for optimizing the performances of different manipulators robot types used for positioning and orientation in an environment cluttered by static objects.

In block 4 of Fig. 3, the triage of the different combinations of the starting and target point coordinates is performed. These combinations give the possible trajectories variants from the beginning to the end configuration. Block 5 gives the grid in configuration space. Some points of the grid point set are used for crossing points possible trajectories.
In block 7 the movement of the manipulator is modeled i.e. the generalized coordinates variations are calculated basing on the kinematic or dynamic model.

In the case of using the kinematic model of the current coordinates are calculated using the formula (6):

$$q_{i+1}^{j} = q_i^{j} + \dot{q}_i^{j}\Delta t \tag{6}$$

i : is the step number along the time axis,
$\dot{q}_j$: generalized velocity of the j^{th} coordinate system.

In case the dynamic model is used in block 7, the already developed motion equation of the manipulator is integrated. In this same block, the Cartesian coordinates of feature points are calculated.

In block 8, the N feature points of the curve are organized. Whereas, the coordinates of feature points are compared with those of boundary points in block 9. Two types of boundaries are used: i) the boundaries applied to the geometry of the machine and, ii) the boundaries applied on the end-effector position.

In the following section we present an algorithm with considering the dynamic position of the obstacles.

II. OPTIMAL MOTION OF A ROBOT MANIPULATOR IN DYNAMIC ENVIRONMENT.

In this section, using the algorithm presented in section 2 allowed us to expand its capabilities to be used in dynamic environment. So in this algorithm, by subdividing the modeling of the movement in some portions and a verification

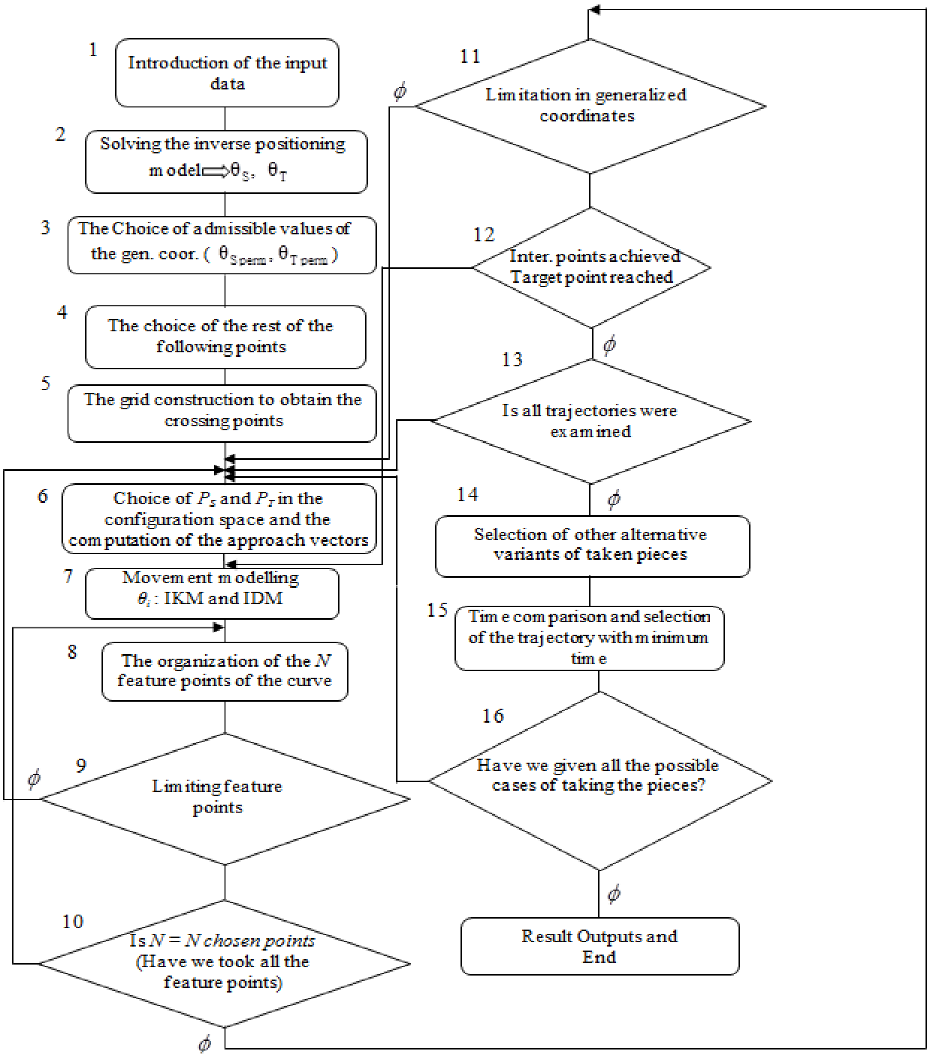

Fig. 3. Optimal algorithm with static obstacles.

of the presence of dynamic obstacles is carried out, which allows to naturally increase the performance of the algorithm.

The algorithm is divided into three parts:

- Identification of the trajectory over time regardless of the dynamic obstacles;
- Travel time optimization of the manipulator end-effector from the initial point to the target point by using the dynamic programming method;
- Motion modelling by taking into account the obstacles dynamics.

The diagram of the modified algorithm of the manipulator robot motion is presented in Figure 4. The first part i.e. the identification of the trajectory over time is presented in the following. Based on the kinematic or dynamic model of the manipulator, for given crossing points, the travel time from the initial point to the target point is calculated regardless of obstacles dynamics.

Kinematic model for the travel time this calculation with the formulas:

$$t = \max_i \left\{ \left(q_i^k - q_i^H \right) / \dot{q}_i \right\} \qquad (7)$$

In block 8 of Fig. 4, the second sub-problem i.e. the travel time optimization problem of the manipulator robot from the initial point to target point is solved, using the dynamic programming method.

The configuration space of the manipulator is divided into k area. In each k^{th} area, the point of n dimension configuration space is chosen, from which arises the crossing point examination.

In the case of motion with only one crossing point, block 8 pick through the points cloud, i.e. distributing the elements of this set in increasing order.

In blocks 9, 10 and 11, the modelling is performed with considering the dynamic obstacles and the searching for the optimal trajectory.

In block 9, after each step as function of time the transition to block 10 is carried out, in which the end-effector point and the feature points are moved by checking the case which does not correspond to the boundary points, the applied shapes geometric of the equipment and the characteristics of the manipulator robot. In case it corresponds to the obstacle go to block 8, where we chose the following trajectory with a higher travel time.

Calculations following the given algorithm (shown in Fig. 4) showed a significant reduction in loss of computer time in comparison with the previous algorithm (shown in Fig. 3).

III. CALCULATING THE TIME CORRESPONDING TO EACH TRAJECTORY POINT

Considering, L the number of segments describing the trajectory positions. Thus, we have the vectors $\vec{x}_0, \vec{x}_1, ..., \vec{x}_L$ through which the path passes ($\vec{x}_0$: start position vector, $\vec{x}_L$: end position vector) as shown in Fig. 5.

In the following, the method allowing calculation of the sequence of vectors $\vec{q}_L$ corresponding to the generalized coordinates q_{1L}, q_{2L}, ..., q_{6L} of the L^{th} point is presented.

With this result we can describe the trajectory of the manipulator robot in the generalized coordinate space. The problem lies in determining the time corresponding to the time of transition of the end-effector by the set of points of the trajectory.

Consider the parameter s monotonically varying along the trajectory. This parameter may be one of the generalized coordinates, any Cartesian coordinate or curvilinear coordinate

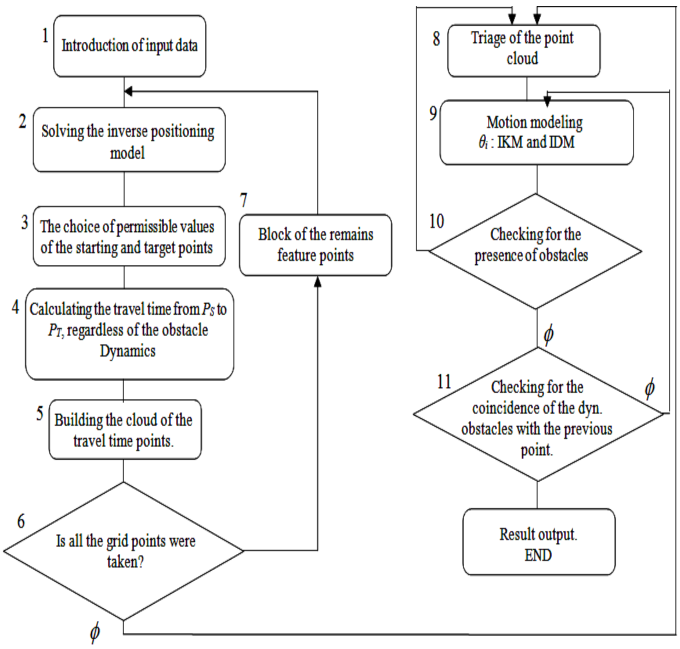

Figure 4. Optimal algorithm considering dynamic obstacles.

system with respect to the absolute coordinates system. The following formulas:

$$\dot{q}_{jL} = \left[q_{j(L+1)}(t_L - t_{L-1})^2 - q_{j(L-1)}(t_{L+1} - t_L)^2 + q_{jL}(t_{L+1} - t_L)^2 - (t_L - t_{L-1})^2\right]$$
$$/\left[(t_{L+1} - t_L)\ (t_L - t_{L-1})\ (t_{L+1} - t_{L-1})\right]$$
$$\ddot{q}_{jL} = 2\left[q_{j(L+1)}(t_L - t_{L-1})^2 + q_{j(L-1)}(t_{L+1} - t_L)\ - q_{jL}(t_{L+1} - t_{L-1})\right]$$
$$/\left[(t_{L+1} - t_L)\ (t_L - t_{L-1})\ (t_{L+1} - t_{L-1})\right] \tag{8}$$

And while replacing time by its corresponding parameter s, the first and second derivatives of the generalized coordinates with respect to the parameter s can be calculated as follows:

$$q'_{jL} = \left(\frac{dq_j}{ds}\right)_L ;\ q''_{jL} = \left(\frac{d^2 q_j}{ds^2}\right)_L \quad (L = 0,\ 1,\ ...,M) \tag{9}$$

L: represents the point number on the trajectory described by the end-effector.

Consider also the average of the first and second derivatives:

$$q'_{j\ L,L+1} = (q'_{j\ L} + q'_{j\ L+1})/2$$
$$q''_{j\ L,L+1} = (q''_{j\ L} + q''_{j\ L+1})/2\ ; \tag{10}$$

Through these values we can express the generalized velocities and generalized accelerations as follows:

$$\dot{q}_j = \frac{dq_j}{ds}\frac{ds}{dt} = q'_j \dot{s}; \tag{11}$$

$$\ddot{q}_j = \frac{d}{dt}(q'_j \dot{s}) = q'_j \ddot{s} + q''_j (\dot{s}^2) \tag{12}$$

For the L^{th} point belonging to the range of the trajectory, the corresponding generalized speeds and accelerations will be:

$$\dot{q}_{j\ L} = q'_{j\ L} \dot{s}_L ; \tag{13}$$

$$\ddot{q}_{j\ L,L+1} = q'_{j\ L,L+1} \ddot{s}_{L,L+1} + q''_{j\ L,L+1} \frac{(\dot{s}_L^2 + \dot{s}_{L+1}^2)}{2} ; \tag{14}$$

We can express the acceleration $\ddot{s}_{L,L+1}$ as function of speed $\dot{s}_L$ by using (15):

$$\ddot{s} = \frac{d\dot{s}}{dt} = \frac{d\dot{s}}{s}\dot{s} = \frac{d}{ds}\left(\frac{\dot{s}^2}{2}\right) \tag{15}$$

Or by its finite difference:

$$\ddot{s}_{L,L+1} = \frac{\dot{s}_{L+1}^2 - \dot{s}_L^2}{2(s_{L+1} - s_L)} \tag{16}$$

For this, instead of (13) and (14) yields:

$$\dot{q}_{jL}^2 = (q'_{jL})^2 \dot{s}_L^2 ; \tag{17}$$

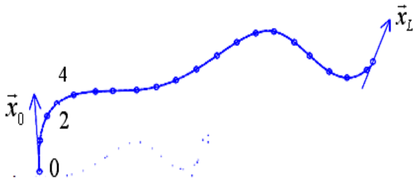

Figure 5. The vector positions of the trajectory.

$$\ddot{q}_{j\;L,L+1} = q'_{j\;L,L+1}\frac{(\dot{s}^2_{L+1}-\dot{s}^2_{L})}{2(s_{L+1}-s_L)} + q''_{j\;L,L+1}\frac{(\dot{s}^2_{L}-\dot{s}^2_{L+1})}{2};\quad (18)$$

And considering the boundaries applied to the generalized velocities and accelerations (1) and (2) and by examining the formula (18) we observe that it is a function of speed $\dot{s}_L$:

$$\dot{s}^2_L \le v^2_{j\;\max}/(q'_{jL})^2;\quad (19)$$

$$-2a_{j\;\max} \le \dot{s}^2_L\left(q''_{jL,L+1} - \frac{q'_{j\;L,L+1}}{s_{L+1}}\right) + \dot{s}^2_{L+1}\left(q''_{j\;L,L+1} + \frac{q'_{j\;L,L+1}}{s_{L+1}-s_L}\right) \le 2a_{j\;\max};\quad (20)$$

In the extreme points of the manipulator robot trajectory we have:

$$\dot{s}^2_0 = 0 \quad (21) \quad \text{and} \quad \dot{s}^2_M = 0 \quad (22)$$

To obtain the minimal displacement, $\dot{s}_L{}^2$ is taken so that the conditions (19) and (22) are satisfied. Travel time can be expressed by the speed $\dot{s}_l$ as follow:

$$t_{L+1} - t_L = \int_{s_L}^{s_{L+1}} \frac{ds}{\sqrt{\dot{s}^2}} = \frac{s_{L+1}-s_L}{\sqrt{(\dot{s}^2_{L+1}+\dot{s}^2_L)/2}}\quad (23)$$

$$t_M - t_0 = \sum_{L=0}^{M-1}(t_{L+1}-t_L)\quad (24)$$

The required objective to achieve the best choice of the value of $\dot{s}_L{}^2$ can be solved using dynamic programming [8].

Considering τ_L the minimum time for which the end-effector of the manipulator robot can be moved from the point s_L (the speed $\dot{s}_L$ is given) to the target position, while satisfying the conditions (19), (20) and (22). And considering ε the accuracy parameter, i.e. it is the $\dot{s}_L{}^2$ size variation step. The intervals (s_L, s_{L+1}) are successively examined, by starting with the last interval for which $L=M-1$. In each interval, two functions $\dot{s}^2_{L+1\;opt}$ $\dot{s}_L{}^2$ and $\tau_L(\dot{s}_L{}^2)$ are arranged, and the array changes with the step ε.

Supposes that the function $\tau_{L+1}(\dot{s}^2{}_{L+1})$ is known. For an optimal movement of the manipulator, the size corresponding to $\dot{s}^2_{L+1\;opt}$ for reaching its target with time τ_L is $\dot{s}^2_{L+1}$.

For the last interval, where $\dot{s}^2_{L+1}$ cannot be reached, in accordance with (1), any size except the zero value satisfies $\tau_M(0) = 0$.

The array for $L<M$, is constructed as follow: The argument $\dot{s}_L{}^2$ successively increases with the step ε, starting from zero. For each value of $\dot{s}_L{}^2$ in the array τ_L, at the beginning a high value of τ_L is inserted. Further, by examining the array for the functions $\tau_{L+1}(\dot{s}^2{}_{L+1})$ and for each item in this array we verify the fulfilment of the inequality (19) and (20). If $\dot{s}_L{}^2 = 0$ and $\dot{s}^2_{L+1} = 0$, all these inequalities are met without conditions. For a high value of $\dot{s}_L$, the inequality does not fulfilled for any size of $\dot{s}^2_{L+1}$.

Suppose, the inequality is fulfilled for any couple $\dot{s}_L{}^2$, $\dot{s}^2_{L+1}$ thus it is calculated as follows:

$$\tau_L^{preal}(\dot{s}^2_L) = \tau_{L+1}(\dot{s}^2_{L+1}) + \frac{s_{L+1}-s_L}{\sqrt{(\dot{s}^2_{L+1}-\dot{s}^2_L)/2}}\quad (25)$$

The obtained values are compared with those written on the array τ_L with the value $\tau_L(\dot{s}_L{}^2)$ obtained before. If the new value of $\tau_L{}^{preal}$ is smaller than the previous one, we insert it to the array, otherwise we retain the oldest value. If the array τ_L is renewed, we reset the value of $\dot{s}^2_{L+1}$, for which the value of $\tau_L{}^{preal}$ is smaller than that found before, in the array $\dot{s}^2_{L+1\;ort}$. Ones, all the arrays $\tau_{L+1}(\dot{s}_L{}^2)$ are examined, we find the minimum value of τ_l and the best value of $\dot{s}^2_{L+1}$, for which this minimum is reached, for the point $\dot{s}^2_L$. As already said, for high value of $\dot{s}^2_L$ the inequality will not be met; as established, $\tau_L(\dot{s}_L{}^2)$ and $\dot{s}^2_{L+1\;ort}\left(\dot{s}^2_L\right)$ requires a finite number of operations. Successively, the function $\tau_{L-1}(\dot{s}^2_{L-1})$, $\tau_{L-2}(\dot{s}^2_{L-2})$ and so on are arranged, finally we get the array $\tau_L(\dot{s}_L{}^2)$; with its help we can construct the array $\tau_0(\dot{s}_0{}^2)$. However, agree with (21), from this array only a single point $\dot{s}^2_0 = 0$ is needed, and it is enough to calculate $\tau_0(0)$. It will be the minimal time for which the manipulator robot stabilizes at the point s_0, and we can go to the point s_M with zero speed.

To be able to see how the speed $\dot{s}_0$ varies along the trajectory, the interval (s_L, s_{L+1}) for which *L=0* must be successively examined. At the beginning of each interval $\dot{s}_L^2$, the known value (on the first interval) is zero. For this known value $\dot{s}_L^2$, the best value of $\dot{s}_{L+1}^2$ on the array $\dot{s}_{L+1\ ort}^2\left(\dot{s}_L^2\right)$ must be taken, and go to the next interval. This is how we must deal to find the best value of $\dot{s}_L^2$ meeting the performance. At the same time from the array $\tau_L(\dot{s}_L{}^2)^*$ the values of τ_L will be normally known, $t_L = t_0 + \tau_0 - \tau_L$, since the known speed $\dot{s}_L$ had to be reached at the known time t_L. Generalized velocities and accelerations are determined by formulas (13).

The calculations with the described algorithm demonstrate the following:

If the limits of speed and acceleration of a generalized coordinate q_L and lower limits according to the generalized coordinates, among other things are chosen, these optimal motion laws are formalized through three stages:

- Dispersion with acceleration limits $\ddot{q}_{L\ \max}$,
- Motion with speed limit $\dot{q}_{L\ \max}$,
- Limit the rapid suffocation speed with minimization of $\ddot{q}_{L\ \max}$.

In this case, the manipulator robot works as a system with one DOF (degree of freedom).

If the limits following the speed and acceleration in generalized coordinates are stiffened, the optimal motion character will be complicated. For example, we obtained the following law: dispersion with limit for q_3 with the acceleration $\ddot{q}_{3\ \max}$, motion with limitation of q_L and speed $\dot{q}_{L\ \max}$, stifling the speed with the minimum limit of $\ddot{q}_{3\ \max}$.

IV. Conclusion and Perspectives

An algorithm considering the dynamic obstacles and a program to solve the problem of optimizing the performance of cyclo-gram manipulator robots, used in technological processes, were developed.

The use of this algorithm has developed a robot learning map with position control system. These maps present the partition of the robot workspace with statement on its grid; the overall time of the manipulator robot trajectory, from the beginning to the target position, is shown in the nodes.

Learning maps enable the choice of the intermediate transition points of the robot trajectory corresponding to the minimum travel time, from the start position to the target, was developed. This algorithm is easily adapted to be used on the control robots working with NC machine tools (machine tools with Numerical Control) and allows achieving optimal performance in the movement of the studied process.

Through what is seen, the motion optimal laws react so that one of the boundary intervals (along $\dot{q}_L$ or $\ddot{q}_L$) arranges for one of the generalized coordinates becomes active (the inequality becomes equality).

The presented solution is well suited to this context. It is optimal with respect to time constraints, it allows a direct calculation.

As perspective we plan to achieve:

- A GUI of the experimental platform;
- A simulation in computer aided design and manufacturing software (SolidWorks);
- A simulation of the robotized cell.
- Create an application comparing the theoretical and practical results.

References

[1] Francisco Javier Montecillo Puente, Human Motion Transfer on Humanoid Robot, Ph.D. thesis, Toulouse University, France, chapter 2, 2010 ;

[2] In-Chul Ha, Kinematic parameter calibration method for industrial robot manipulator using the relative position, Journal of Mechanical Science and Technology 22 (2008) 1084~1090, 2008.

[3] Frank Shaopeng Cheng, Calibration of Robot Reference Frames for Enhanced Robot Positioning Accuracy, Robot Manipulators, Marco Ceccarelli (Ed.), ISBN: 978-953-7619-06-0, InTech, 2008.

[4] Peiwen Guoa , Amir Anvar a, Kuan Meng Tan, “Intelligent submersible manipulator-robot, design, modeling, simulation and motion optimization for maritime robotic research”, 20th International Congress on Modelling and Simulation, Adelaide, Australia, 1–6 December 2013.

[5] Briot S., Pashkevich A. and Chablat D., “Optimal Technology-Oriented Design of Parallel Robots for High-Speed Machining Applications”, IEEE 2010 International Conference on Robotics and Automation, In Press.

[6] Zhen Gao, Dan Zhang, Yunjian Ge, “Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches” ,Robotics and Computer-Integrated Manufacturing, Volume 26, Issue 2, April 2010, Pages 180–189.

[7] Christophe Jacquemin, Les mouvements d'un robot commandés par le cerveau d'un singe, Duke University (Caroline du Nord), Automates intelligents, 2000;

[8] Marc J Richard, François Dufour, Stanislaw Tarasiewicz, Dynamic programming for the control of robots, Mechanism and machine theory, vol. 28, issue 3, may 1993, pages 301-316 ;