Optimal time trajectories for industrial robots with torque, power, jerk and energy consumed constraints

Francisco Rubio, Francisco Valero, Josep Lluís Suñer, Juan Ignacio Cuadrado

CITV. Universidad Politécnica de Valencia
frubio@mcm.upv.es

Abstract

Purpose The purpose of this paper is to analyze the impact of the torque, power, jerk and energy consumed constraints on the generation of minimum time collision-free trajectories for industrial robots in a complex environment.
Design/methodology/approach An algorithm is presented in which the trajectory is generated under real working constraints (specifically torque, power, jerk and energy consumed). It also takes into account the presence of obstacles (to avoid collisions) and the dynamics of the robotic system. The method solves an optimization problem to find the minimum time trajectory to perform the tasks the robot should do.
Findings Important conclusions have been reached when solving the trajectory planning problem related to the value of the torque, power, jerk and energy consumed and the relationship between them, therefore enabling the user to choose the most efficient way of working depending on which parameter he is most interested in optimizing. From the examples solved we have found the relationship between the maximum and minimum values of the parameters studied.
Research limitations/implications This new approach tries to model the real behaviour of the actuators in order to be able to upgrade the trajectory quality. So a lot of work has to be done in this field.
Practical implications The algorithm solve the trajectory planning problem for any industrial robot and the real characteristics of the actuators are taken into account which is essential to improve the performance of it.
Originality/value This new tool enables us to improve the performance of the robot by combining adequately the values of the mentioned parameters (torque, power, jerk and consumed energy).

Keywords*:* Industrial Robots; Trajectory Planning; Dynamics Constraints; Optimal time

Paper Type*:* Research paper

Introduction

The main use of robots in industrial processes aims is to improve the quality of the products, increase productivity and reduce cycle times. To achieve these objectives, the way in which the motion of robot arms is generated is essential and this depends on the characteristics of the actuators that are responsible for generating those motions. Important working parameters and variables are the time needed to perform the trajectory, the energy consumed, the torque and power to be transmitted and the working velocities, which are conditioned by the accelerations and jerks produced.

Each of these working parameters has an impact on the work of the robot. Most of the algorithms that address the problem of collision-free trajectory planning for industrial robots try to optimize some of the working parameters or some of the objective functions. The optimization criteria most widely used can be classified as follows:
(1) Minimum execution time (related to productivity)
(2) Minimum jerk (related to the quality of work, accuracy and equipment maintenance)
(3) Minimum energy consumed (or minimum actuator effort) (related to savings)
(4) Hybrid criteria, e.g. minimum time and energy.

In the past, the early algorithms that solved the trajectory planning problem tried to minimize the time needed for performing the task (see Bobrow *et al.*, 1985, Shin *et al.*, 1985, Chen *et al.*, 1989). One disadvantage of those minimum-time algorithms was that the trajectories had discontinuous values of acceleration and torques which led to dynamic problems during the trajectory performance. Those problems were avoided by imposing smooth trajectories to be followed, such as spline functions which have been used in both path and trajectory planning.

Another way of tackling the trajectory planning problem was based on searching for jerk-optimal trajectories. It should be noticed that the effect of the jerk is vital for working with precision, without vibrations, and it also affects the control system and the wearing of joints and bars. These methods enable the errors, the stresses (in the actuators and also in the mechanical structure of the robot) and the resonance frequencies to shrink during trajectory tracking.

Jerk constraints were introduced by Kyriakopoulos (see Kyriakopoulos *et al.*,1988). Later, Constantinescou introduced (Kyriakopoulos and Saridis, 1988) a method for determining smooth and time-optimal path-constrained trajectories for robotic manipulators by imposing limits on the actuator jerks. The third derivative of the path parameter with respect to time, the pseudo-jerk, is the controlled input. The limits on the actuator torques turn into state-dependent limits on the pseudo-jerk. The time-optimal control objective gives way to an optimization problem by using cubic splines to parameterize the state space trajectory.

Piazzi also follows a jerk-optimal trajectory method. In Piazzi and Visioli, 1997 and Piazzi and Visioli, 2000, the interpolation of the robot joint trajectory is created by using cubic splines that ensure the overall continuity of velocities and accelerations in the robot motion. The resulting minimum jerk trajectory planning is shown to be a globally constrained mini-max optimization problem. This is solved by a newly devised algorithm based on interval analysis, and proof of convergence with certainty to an arbitrarily good global solution is provided, and the paper includes simulation examples. He describes the algorithm employed and the computational results are also presented. In Piazzi and Visioli, 1998, they addressed the global minimum-time trajectory planning of an *m*-joint mechanical manipulator. Using a joint space scheme with given intermediate points to be interpolated by piecewise cubic polynomials, they proposed the scheduling of the times between adjacent joints under velocity, acceleration and jerk constraints. This algorithm, which is proved to be globally convergent with certainty within an arbitrary precision, uses an interval procedure (a subroutine adapting the tools and ideas of interval analysis) in proving that a local minimum is actually a global one.

Huang also proposed an optimal jerk algorithm. In Huang *et al.*, 2006 he introduced a global minimum-jerk trajectory planning algorithm of a space manipulator. In later papers, Huang analyzes the influence of the torque on the path generated using genetic algorithms. He concludes that a genetic algorithm (minimum-torque path-planning method) has satisfactory performance and real values.

Another different approach to solving the trajectory planning problem is based on minimizing the torque and the energy consumed instead of the execution time or the jerk. This approach leads to smoother trajectories. An early example is seen in Garg *et al.*, 1992. In this paper Garg proposed a strategy for force balance and energy optimization for cooperating manipulators. The algorithm was applied to two SCARA robots forming a closed kinematic chain that were controlled using their individual controllers. A position control strategy was used for each robot and the corresponding end-effector forces were calculated. These forces were equalized and the corresponding power used was computed. They employed the linear programming technique to calculate external forces and the power used in the direction of motion was minimized.

Similarly, Hirakawa and Kawamura searched for the minimum energy consumed. In Hirakawa *et al.*, 1996 they proposed a method for solving the trajectory generation problem in redundant degree of freedom manipulators. They used a variational approach and the B-Spline curve was introduced to minimize the electrical energy consumed in a robot manipulator system.

In Field and Stepanenko, 1996, the authors proposed a technique of iterative dynamic programming to plan minimum energy consumption trajectories for robotic manipulators. The dynamic programming method was modified to perform a series of dynamic programming passes over a small reconfigurable grid covering only a portion of the solution space at any one pass. Although strictly no longer a global optimization process, this iterative approach retained the ability to avoid certain poor local minima while avoiding the dimensional issue associated with a pure dynamic programming approach. The modified dynamic programming approach was verified experimentally by planning and executing a minimum energy consumed path for a Reis V15 industrial manipulator.

Saramago also tried to optimize trajectory planning. In Saramago and Steffen, 1998, the authors introduced an approach to the solution of moving a robot manipulator at minimum cost along a specified geometric path in the presence of obstacles. The main idea was to express obstacle avoidance in terms of the distances between potentially colliding parts. The optimal travelling time and the minimum mechanical energy of the actuators were considered together to build a multi-objective function. A simple numerical example involving a Cartesian manipulator arm with two-degrees-of-freedom was described. In Saramago and Steffen, 2000, they considered not only the minimum time but also the minimum mechanical energy of the actuators. They built a multi-objective function and the results obtained depended on the associated weighting factor.

The subject of energy minimization continues to be of interest in the field of robotics and automated manufacturing processes. For example, in Cho *et al.*, 2006, they introduced the torque supplied by the actuators as constraints on the trajectory planning problem. In that study, the maximum range for velocity and acceleration was specified. The force exerted on the object to be manipulated was also considered so that the robot moved by both discrete and continuous paths.

Later, new approaches appear for solving the trajectory planning problem. The idea of using a weighted objective function to optimize the operating parameters of the robot arises. Among the authors who put it into practice is Chettibi, who addresses the problem by minimizing a cost function that is a combination of several important operating parameters. In Chettibi *et al.*, 2006, he introduced constraints on position, speed, jerks and torques.

The basic idea is to directly parameterize the evolution of joint vector *q(t)*. The optimal control problem becomes a parametric constrained optimization problem which is solved for the unknown transfer time *T* and the unknown parameters of the chosen model for *q(t)*. The cost function is a weighted balance of transfer time, mean average of actuators and power.

Gasparetto and Zanotto also use a weighted objective function (see Gasparetto and Zanotto, 2010). In order to get the optimal trajectory, an objective function composed of two terms is minimized: a first term proportional to the total execution time and another one proportional to the integral of the squared jerk (defined as the derivative of the acceleration) along the trajectory. This latter term ensures that the resulting trajectory is smooth enough. The proposed technique enables the kinematic constraints on the robot motion to be taken into account, expressed as upper bounds on the absolute values of velocity, acceleration and jerk. The method described is then applied to a 6-d.o.f. robot (a Cartesian gantry manipulator with a spherical wrist). This work continues the research of Gasparetto and Zanotto, 2007 and Gasparetto and Zanotto, 2008 where they work with acceleration constraints to determine the minimum time trajectory.

In this paper we will solve the collision-free trajectory planning problem for industrial robots under torque, power, jerk and energy consumed constraints. Note that this last constraint is very popular nowadays because of current sensitivity with regard to energy resources and respect for the environment.

We start from the previous work done by the authors (see Valero *et al.*, 2006, and Rubio *et al.*, 2009). Specifically, the two-stages trajectory planner is used: in the first one, a sequence of configurations that guarantees a collision-free path is generated, while in the second stage a trajectory is adjusted to the previous sequence.

The aim of this study is to focus on improving performance in this second stage.

Problem statement.

The proposed trajectory planner is arranged in two successive stages and has been applied to a PUMA 560 robot (the robot and its wire model is shown in Figure 1, for more details see Rubio et al., 2009).

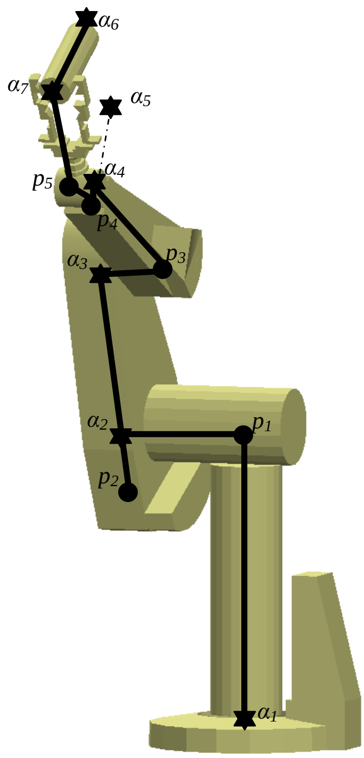


Figure 1 Robot Puma 560

In the first stage a discrete configuration space is generated from two feasible and distant robot configurations. The problem is set out in the Cartesian space, which leads to a much easier problem formulation. The configuration space generation is based on the concept of adjacent configuration (Valero *et al.*, 2006 and Rubio *et al.*, 2009).

The algorithm generates the configuration space, taking into account the geometrical characteristics of the problem. In the second stage, a weighted graph is related to the configuration space previously obtained, which allows a minimum weighted free-collision path to be obtained between the initial and final configurations. Once the path is available, the dynamic characteristics of the robot are included, setting an optimal trajectory planning problem between couples of adjacent configurations, which attempts to obtain the minimum time trajectory between them that is compatible with the robot features and the actuator capabilities. From the complete sequence of configurations, a trajectory in the joint space is generated that minimizes the total time required and is dynamically compatible with the robot features and subjected to torque, jerk and consumed energy constraints.

To adjust the trajectory, given a sequence of m robot configurations, $S = \{S_1(q_{i1}), S_2(q_{i2})\ldots S_m(q_{im})\}$, we look for minimum time trajectory (t_{min}), that contain them. The path is considered to be decomposed into m-1 intervals, so the time needed to reach the S_{j+1} configuration from the initial S_1 is t_j, and the time spent in the segment j (between S_j and S_{j+1} configurations) will be t_j-t_{j-1}. In each interval, cubic interpolation functions have been used for joint trajectories. They are defined by means of joint variables between successive configurations, so that for the segment j is:

$\forall t \in \left[t_{j-1}, t_j\right[\Rightarrow q_{ij} = a_{ij} + b_{ij}t + c_{ij}t^2 + d_{ij}t^3$ for i=1,…,dof (dof being the degrees of freedom of the robot) and j=1,…,m-1. (m is the number of the robot configuration)

To ensure motion continuity between configurations, the following conditions associated to the given configurations are considered.

- • Position: At each interval j the initial and final position should be coincident between S_j and S_{j+1}, giving a total of ($2dof\,(m$-$1)$) equations:

$$q_{ij}\left(t_{j-1}\right) = a_{ij} + b_{ij}t_{j-1} + c_{ij}t_{j-1}^2 + d_{ij}t_{j-1}^3 \quad (1)$$

$$q_{ij}\left(t_j\right) = a_{ij} + b_{ij}t_j + c_{ij}t_j^2 + d_{ij}t_j^3 \quad (2)$$

- Velocity: velocities of the initial and final configurations must be zero, giving place to (*2dof*) equations:

$$\dot{q}_{i1}(t_0) = 0 \tag{3}$$

$$\dot{q}_{im}(t_m) = 0 \tag{4}$$

When passing through each configuration, the final velocity of the previous configuration should be equal to the initial velocity of the next configuration, leading to (*dof (m-2)*) equations

$$\dot{q}_{ij}(t_j) = \dot{q}_{ij+1}(t_j) \tag{5}$$

- Acceleration: For each intermediate configuration, the final acceleration of the previous configuration should be equal to the initial acceleration of the next configuration. For each intermediate configuration, the final acceleration of the previous interval must be equal to the first of the following one, giving rise to (*dof*(*m*-2)) equations:

$$\ddot{q}_{ij}(t_j) = \ddot{q}_{ij+1}(t_j) \tag{6}$$

Knowing the time required to execute the trajectory between the different configurations, the cubic polynomial coefficients can be obtained by using the above equations efficiently (by solving the inverse problem dynamic) and the normal time calculation procedure (see Suñer, J.L. et al [2]).

In addition, the minimum time trajectory must meet the following three types of constraints:

- Maximum torque on the actuators,

$$\tau_i^{\min} \le \tau_i(t) \le \tau_i^{\max} \quad \forall t \in [0, t_{\min}],\ i = 1 \ldots dof \tag{7}$$

- Maximum power on the actuators,

$$P_i^{\min} \le \tau_i(t) \le P_i^{\max} \quad \forall t \in [0, t_{\min}],\ i = 1 \ldots dof \tag{8}$$

- Maximum jerk on the actuators,

$$\dddot{q}_i^{\min} \le \dddot{q}_i(t) \le \dddot{q}_i^{\max} \quad \forall t \in [0, t_{\min}],\ i = 1 \ldots dof \tag{9}$$

- Consumed Energy,

$\sum_{j=1}^{m-1}\left(\sum_{i=1}^{dof} \varepsilon_{ij}\right) \le \mathrm{E}$, ε_{ij} being the energy consumed by the *i* actuator between configurations *j* and *j*+1

Given the large number of iterations required by the process, the technique used for obtaining the coefficients is crucial. The first task is to normalize the polynomials that define the stages (see Suñer *et al.*, 2007).In short, the optimization problem is set by using incremental time variables in each interval, so that in the interval between S_j and S_{j+1}, the time variable should be $\Delta t_j = t_j - t_{j-1}$, and the objective function,

$$\sum_{j=1}^{m-1} \Delta t_j = t_{\min} . \tag{10}$$

The solution is obtained by means of SQP procedures, so that at each iterative step it is necessary to obtain the above mentioned polynomials coefficients from the estimation of the variables of the problem.

Results

Five different examples have been solved, with sequences between 32 and 57 configurations, for a PUMA 560 robot. The constraints accounted for can be seen in Table I, where the values of maximum and minimum torque and the power for each actuator are shown.

Table I: Limits of power and torque for each actuator

i	1	2	3	4	5	6
τ_i^{max}(N m)	140	180	140	80	80	40
τ_i^{min}(N m)	-140	-180	-140	-80	-80	-40
P_i^{max}(W)	275	350	275	150	150	75
P_i^{min}(W)	-275	-350	-275	-150	-150	-75

The examples have been solved with sequences of different configurations, considering a series of cases based on using different jerk and energy consumed constraints. Example of Case 1 is shown in Figure 2, where we can see the initial and final configuration and the trajectory generated:

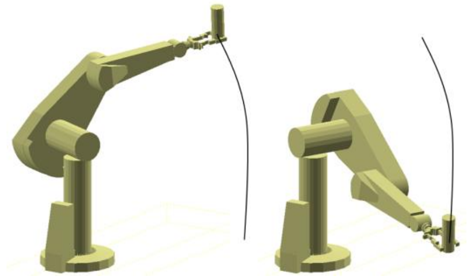


Figure 2 Initial and final configuration and trajectory generated for case 1.

Table II shows the detail of each case.

Table II: Cases solved with energy and jerk constraints.

Case*	Max. Jerk Constraint (rad/s^3)	Consumed Energy Constraint (Joules)	Execution Time (s.)	Consumed Energy (Joules)	Jerk in 3rd Actuator. (rad/s^3)	
					Máx.	Min.
1_s_s	Unconstrained**	Unconstrained**	3,79	140,1	1615	-841
1_s_75	Unconstrained**	75	22,55	75	107	-28
1_5_s	5	Unconstrained**	19,27	80,1	5	-5
1_5_75	5	75	25,76	75	5	-4
2_s_s	Unconstrained**	Unconstrained**	5,14	203,49	1098	-984
2_s_200	Unconstrained**	200	5,15	200	1096	-980
2_s_175	Unconstrained**	175	5,3	175	954	-872
2_s_150	Unconstrained**	150	5,62	150	704	-627
2_s_125	Unconstrained**	125	6,42	125	425	-418
2_s_100	Unconstrained**	100	12,25	100	167	-145
2_s_95	Unconstrained**	95	21,08	95	176	-37
2_5_s	5	Unconstrained**	23,05	103,3	5	-5
2_5_95	5	95	26,35	95	5	-5
3_s_s	Unconstrained**	Unconstrained**	2,27	87,5	897	-944
3_s_50	Unconstrained**	50	7,34	50	125	-147
3_5_s	5	Unconstrained**	14,82	51,5	5	-5
3_5_50	5	50	17,94	50	2,8	-3,6
4_5_s	5	Unconstrained**	18,28	42,14	5	-3,9
4_10_s	10	Unconstrained**	14,51	42,39	10	-7,7
4_25_s	25	Unconstrained**	10,69	43,1	25	-19,3
4_5_s	5	Unconstrained	18,28	42,14	5	-3,9
4_10_s	10	Unconstrained	14,51	42,39	10	-7,7
4_25_s	25	Unconstrained	10,69	43,1	25	-19,3
4_50_s	50	Unconstrained	8,49	43,92	50	-38,7
4_100_s	100	Unconstrained	6,74	45,36	100	-77,4
4_1000_s	1000	Unconstrained	3,21	63,37	856,8	-776,2
4_s_s	Unconstrained	Unconstrained	2,41	88,46	1361,5	-1802,9
4_5_40	5	40	18,65	40	5	-4
4_s_40	Unconstrained	40	9,94	40	337	-13,2
5_s_s	Unconstrained	Unconstrained	3,08	106,2	972	-1009
5_s_40	Unconstrained	40	9,18	40	42,2	-37,8
5_5_s	5	Unconstrained	15,91	40,3	5	-5
5_5_40	5	40	15,93	40	5	-5

*(Nomenclature used. Case: numberexample_const-ddd_const-ener). **(Nomenclature used. Unconstrained: s)

Table II presents the results, execution time required to perform the robot trajectory, energy consumed and the jerk reached on the third actuator. We only represent the values of the third actuator because it has significant behaviour and we also try to avoid generating too much information in the tables.

Effect of the energy consumed constraint in Example 2

Figure 3 shows the evolution of acceleration for the first three actuators in the case 2_s_s without constraints on the consumed energy and jerk. It can be seen that the greatest accelerations are obtained for the third actuator.

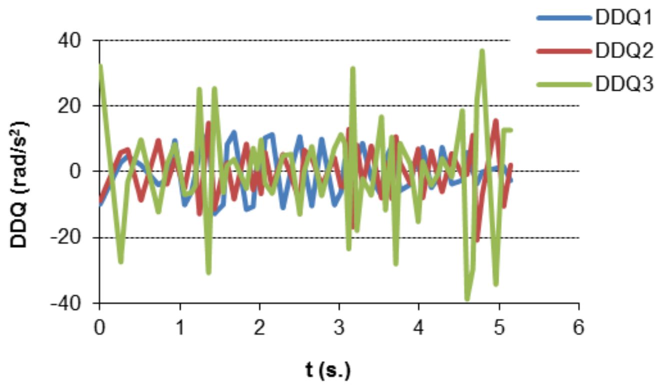


Figure 3 Acceleration in the first three actuators for the case 2_s_s

If the same example number 2 restricts the value of energy to the values indicated in each case (s_95 = 95 J ...., s_150 = 150 J), we obtain the graphs of Figure 4. It can be seen that as the energy is gradually constrained, the trajectory times become longer, especially for more severe values of the constraint. It can also be seen that in cases with more severe constraints on the energy consumed (100 J and 95 J), the variation in the time required to perform the trajectory is very important, while the maximum acceleration values retain the same order of magnitude.

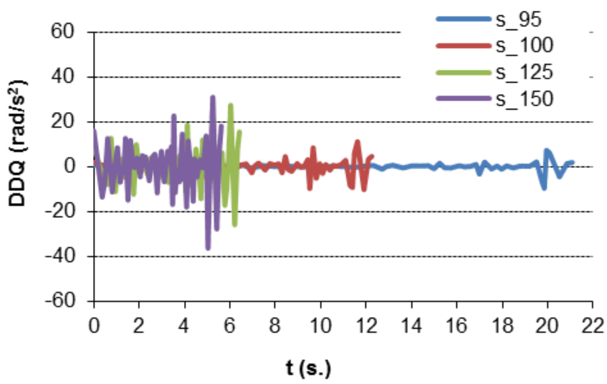


Figure 4 Acceleration of the actuator number 3 with different constraints on the consumed energy for example 2

In Figure 5 the behaviour of the acceleration of the third operator is shown, but with less severe energy constraints. The values of maximum and minimum acceleration do not suffer significant variations, resulting in greater module acceleration for the greatest energy consumed constraint case.

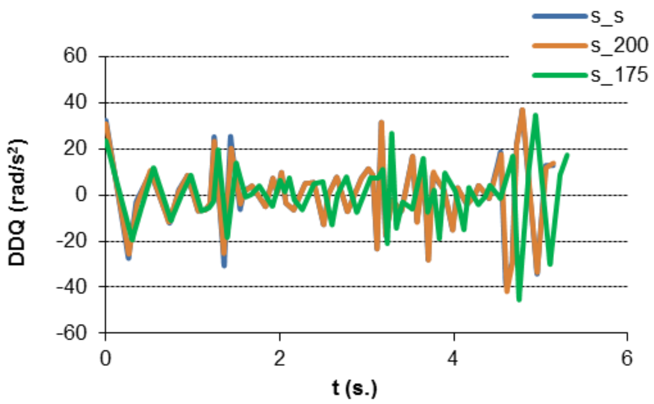


Figure 5 Acceleration of the actuator 3 for example 2, compared with the case without energy constraint (s_s) and with less severe constraints

If we analyze the power consumed in the previous cases, we obtain the graph in Figure 6. It can be seen that the power peaks decrease as the energy is constrained and that in turn the trajectory time increases.

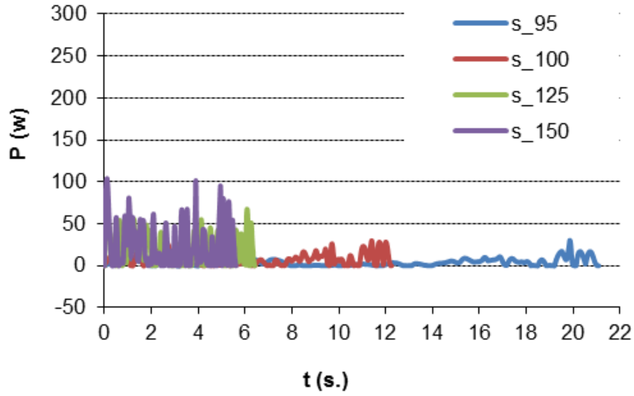


Figure 6 Consumed power for example 2 by the robot with different constraints of consumed energy

If power constraints are lenient, the results are shown in Figure 7, where it can be seen that the power variations are less pronounced.

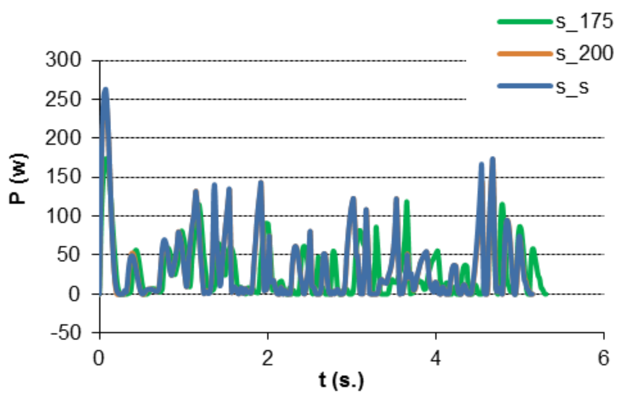


Figure 7 Consumed power for example 2 without consumed energy constraints and with less severe constraints

Effect of the jerk constraint for example number 4

Next we can see the graphs obtained for example number 4 in cases where the energy is constrained but not the jerk. Figure 8 shows that the peak accelerations are reduced while the trajectory time is increased when the jerk constraint become more severe.

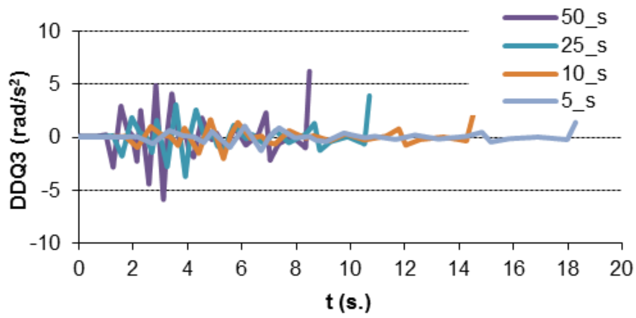


Figure 8 Acceleration of the actuator number 3 with different jerk constraints for example 4

The same behaviour is observed when the jerk constraint is less severe. See Figure 9.

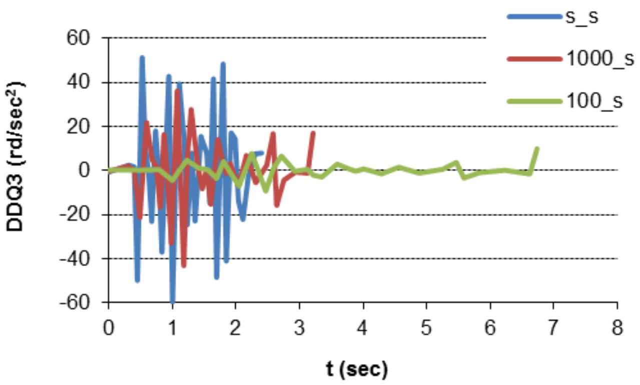


Figure 9 Acceleration of the actuator number 3 with different jerk constraints for example 4

This fact means that the graphs of velocity are smoother. See Figure 10.

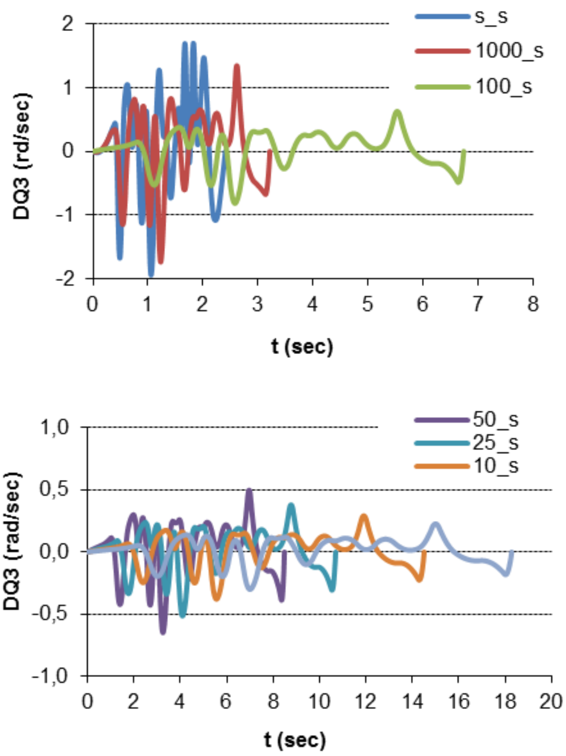
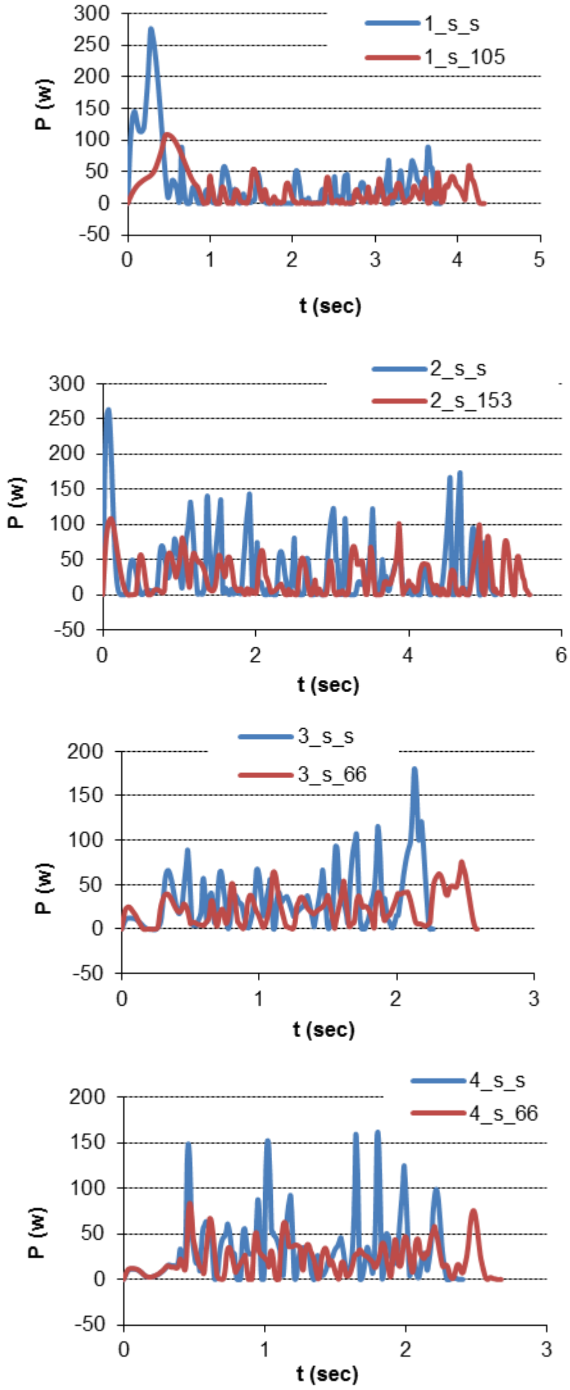


Figure 10 Velocity in the actuator number 3 for different jerk constraints for example 4

Effect of the energy consumed constraint when reduced by up to 75%.

The following diagrams show the evolution of the power consumed for a 25% reduction in the energy consumed in each example when compared with the same case without constraints on acceleration and energy. The result is an increase in the time required of between 8% and 14% when the energy consumed is reduced by 25%. See Figure 11.



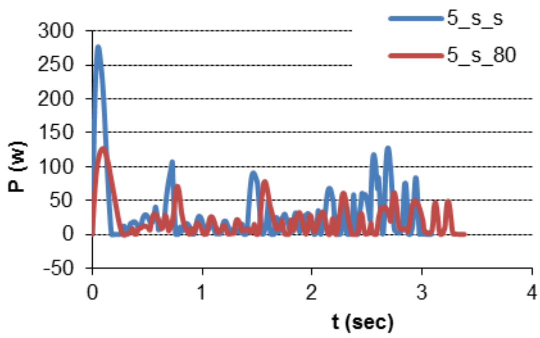


Figure 11 Consumed power in examples 1, 2, 3, 4 and 5, respectively

Table III shows the numerical results for the examples in this section.

Table III: Results obtained when imposing constraints on 75% of the energy consumed over its maximum value

Case	Max. Jerk Constraint (rad/s^3)	Consumed Energy Constraint (Joules)	Execution Time (s.)	Consumed Energy (Joules)	Jerk in the 3rd actuator. (rad/s^3)		ΔT s_s %	ΔEn s_s %
					Máx.	Min.		
1_s_105	unconstraint	105,08	4,32	105,08	945,6	-559,6	14,0	-25,0
1_s_s	unconstraint	sin	3,79	140,1	1615,1	-841,3	0,0	0,0
2_s_153	unconstraint	152,62	5,58	152,62	728,2	-659,1	8,6	-25,0
2_s_s	unconstraint	sin	5,14	203,49	1097,5	-983,7	0,0	0,0
3_s_66	unconstraint	65,63	2,59	65,63	472,2	-529,4	14,1	-25,0
3_s_s	unconstraint	sin	2,27	87,5	896,9	-944,0	0,0	0,0
4_s_66	unconstraint	66,35	2,68	66,35	732,1	-740,3	11,2	-25,0
4_s_s	unconstraint	sin	2,41	88,46	1361,5	-1802,9	0,0	0,0
5_s_80	unconstraint	79,65	3,38	79,65	604,6	-574,4	9,7	-25,0
5_s_s	unconstraint	sin	3,08	106,2	971,5	-1008,5	0,0	0,0

Effect of the combination of severe constraints

The following graph in Figure 12 shows the effect of the jerk constraints acting with acute energy consumed constraints (40 W.). First, it should be noted that a severe energy constraint does not prevent high and significant jerk values from appearing, see Table II. Notice in examples 4 and 5 how severe jerk constraints (5 rad/s^3) have a major effect on the time required to perform the trajectory. The following graph shows the effect of the jerk constraints acting with acute consumed energy constraints (40 W.).

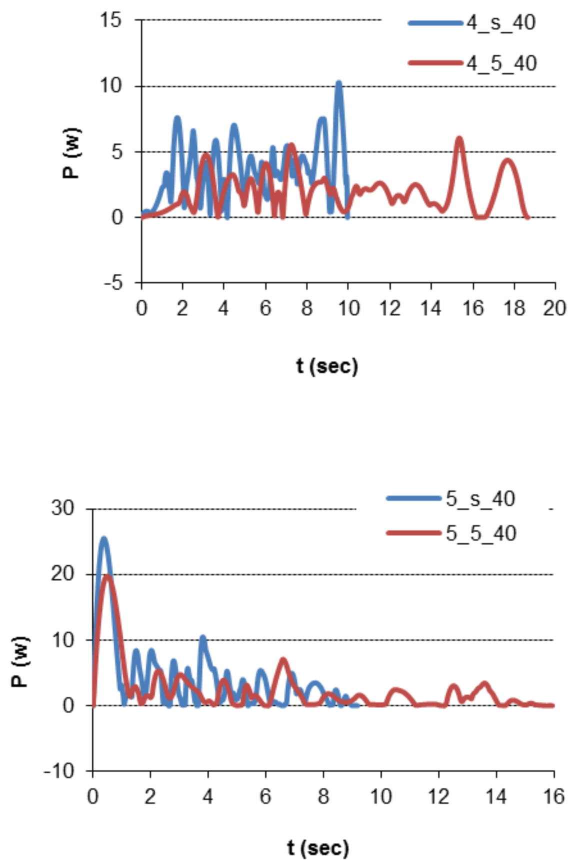


Figure 12 Consumed power in examples 4 and 5, respectively

Conclusions

A trajectory planner has been introduced in which the trajectories calculated meet constraints on torque, power, jerk and energy consumed. This paper analyzes the impact of the torque, power, jerks and energy consumed constraints on the generation of minimum time collision-free trajectories for industrial robots. The algorithm works on a discrete configuration space and the inverse dynamic problem has been solved using cubic interpolation functions between two adjacent configurations. The algorithm has been applied on a large number of examples. The results obtained have been analysed on the basis of the minimum time required to perform the trajectory and the main conclusions are:

a) Constraints on the energy consumed must enable the manipulator to exceed the requirements associated with potential energy, as the algorithm works on the assumption that the energy can be dissipated but not recovered.

b) Increase in the severity of energy constraints results in longer time trajectories with more soft power requirements (Figure 11). When constraints are not very severe, efficient trajectories can be obtained without high penalties on the working time cycle.

c) An increase in the severity of the jerk constraints involves longer time trajectories with more soft power requirements (Figure 8, 9 and 12) and lower energy consumed (Table II). When constraints are very severe, times are also severely penalized (Figure 8 and 12).

d) When dealing with power, torque and energy consumed constraints simultaneously, even with severe values of them, significant problems associated with the jerk might appear (Figure 5).

e) To obtain competitive results in the balance between time cycle and energy consumed, the actuators should work with the maximum admissible value of the jerk so that the robot can work with the desired accuracy.

Acknowledgements

This paper has been made possible thanks to support from the Spanish Ministry of Science and Innovation through the Project for Research and Technological Development with ref. DPI2010-20 814-C02-01.

References

Bobrow, J.E., Dubowsky, S. and Gibson, J.S. (1985), "Time-Optimal Control of Robotic Manipulators Along Specied Paths", *International Journal of Robotics Research*, Vol. 4, No 3, pp. 3-17.

Chen, Y. and Desrochers, A.A. (1989), "Structure of minimum time control law for robotic manipulators with constrained paths", in *IEEE Int Conf Robot Automat in Scottsdale, USA, 1989,* ISBN: 0-8186-1938-4, pp. 971-976.

Chettibi, T., Lehtihet, H. E., Haddad, M. and Hanchi, S.(2002), "Minimum cost trajectory planning for industrial robots", *Mechanism and Machine Theory,* Vol. 37, No 10, pp. 1063-1086.

Cho, B. H., Choi, B. S. and Lee, J. M. (2006), "Time-optimal trajectory planning for a robot system under torque and impulse constraints", *International Journal of Control, Automation, and Systems*, Vol. 4, No 1, pp. 10-16.

Constantinescu, D. and Croft, E.A. (2000), "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths", *Journal of Robotic Systems,* Vol 17, No 5, pp. 233-249.

Field, G. and Stepanenko, Y. (1996), "Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators", in *Proc. of the IEEE International Conference on Robotics and Automation in Minneapolis, USA, 1996,* ISBN: 0-7803-2988-0, pp. 2755-2760.

Garg, D. and Ruengcharungpong, C. (1992), "Force balance and energy optimization in cooperating manipulators", in *Proceedings of the 23rd Annual Pittsburgh Modeling and Simulation Conference in Pittsburgh, USA, 1992*, pp. 2017-2024.

Gasparetto, A. and Zanotto, V. (2007),"A new method for smooth trajectory planning of robot manipulators", *Mechanism and Machine Theory,* Vol. 42, No 4, pp. 455-471.

Gasparetto, A. and Zanotto, V. (2008), "A technique for time-jerk optimal planning of robot trajectories", *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No 3, pp. 415-426.

Gasparetto, A. and Zanotto, V. (2010), "Optimal trajectory planning for industrial robots", *Advances in Engineering Software*, vol. 41, No 4, pp. 548-556.

Hirakawa, A. and Kawamura, A. (1996), "Proposal of trajectory generation for redundant manipulators using variational approach applied to minimization of consumed electrical energy", in *Proceedings of the Fourth International Workshop on Advanced Motion Control in Mie , Japan, 1996*, ISBN: 0-7803-3219-9, pp. 687-692.

Huang, P., Xu, Y. and Liang, B. (2006), "Global Minimum-Jerk trajectory Planning of space manipulator", *International Journal of Control, Automation and Systems,* Vol. 4, pp. 405-413

Kyriakopoulos, K.J. and Saridis, G.N. (1988), "Minimum jerk path generation", *in IEEE international conference on robotics and automation in Philadelphia, USA, 1988,* ISBN: 0-8186-0852-8, pp. 364-369.

Piazzi, A. and Visioli, A. (1997), "An interval algorithm for minimum-jerk trajectory planning of robot manipulators", in *Proc. of the 36th conference on decision and control, San Diego, California, 1997,* ISBN: 0-7803-4187-2 pp. 1924-1927.

Piazzi, A. and Visioli, A. (1998), "Global minimum-time trajectory planning of mechanical manipulators using interval analysis", *International Journal of Control*, Vol. 71, No 4, pp. 631-652.

Piazzi, A. and Visioli, A. (2000), "Global minimum-jerk trajectory planning of robot manipulators", in *IEEE Transactions on Industrial Electronics,* ISSN: 0278-0046, pp. 140-149.

Rubio, F.J., Valero, F.J., Suñer, J.L. and Mata, V. (2009), "Direct step-by-step method for industrial robot path planning", *Industrial Robot: An International Journal,* Vol. 36, No 6, pp. 594-607

Saramago, S.F.P. and Steffen, V. Jr. (1998), "Optimization of the trajectory planning of robot manipulators tacking into account the dynamics of the system", *Mechanism and Machine Theory* , Vol. 33, No 7, pp. 883-894.

Saramago, S.F.P. and Steffen, V. Jr. (2000), "Optimal trajectory planning of robot manipulators in the presence of moving obstacles", *Mechanism and Machine Theory,* Vol. 35, pp. 1079-1094.

Shin, K.G. and McKay, N.D. (1985), "Minimum-time control of robotic manipulators with geometric path constraints", *in IEEE Transactions on Automatic Control*, ISSN: 0018-9286, pp. 531-541.

Suñer, J.L., Valero, F.J., Ródenas, J.J., and Besa, A. (2007), "Comparación entre procedimientos de solución de la interpolación por funciones splines para la planificación de trayectorias de robots industriales", *in VIII Congreso Iberoamericano de Ingeniería Mecánica in Cusco, Perú, 2007*, ISBN: 978-9972-2885-31.

Valero, F., Mata, V. and Besa, A. (2006), "Trajectory planning in workspaces with obstacles taking into account the dynamic robot behavior", *Mechanism and Machine Theory*, Vol. 41, pp. 525-536.