Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Efficient Low-cost Controllers for Constrained Manipulators with Uncertainties and Disturbances

S. Torres*, J.A. Méndez, L. Acosta, E.J. González, J. Toledo

**Departamento de Ingeniería de Sistemas y Automática y ATC, Universidad de La Laguna, 38271, Islas Canarias, Spain (Tel:+34 922318280; e-mail: storres@ull.es).*

Abstract: This work presents an efficient control strategy for robot manipulators with constraints and affected by uncertainties and disturbances. The controller uses a combination of model predictive control (MPC) with an adaptive robust feedforward term. The predictive controller is based on interpolations between different simple solutions to guarantee the feasibility of the final solution applied to the manipulator. The proposed method improves the existing techniques in terms of robust capabilities. Feasibility is preserved with the MPC and applicability is also guaranteed as the computational load of the interpolation algorithm is low. The benefits of the strategy, compared with other existing controllers, are shown with simulation results obtained with a PUMA-560 manipulator.

1. INTRODUCTION

The problem of controlling robot manipulators is a well-known topic that has been dealt since the 80's with satisfactory performance for the regulation and tracking problems. Several strategies based on adaptive (Whitcomb *et al.*, 1993; Spong, 2003) and robust control (Spong, 1992; Alonge *et al.*, 2004) have been designed to take into account the model uncertainties. Their main drawback is that they assume some bounds in the uncertainties and disturbances, in order to tune the controller parameters. In the case of a manipulator with rigid elements, the defined techniques are efficient, but they have to be revised with flexible and low-mass structures, where uncertainties and disturbances affect more decisively the global performance of the structure. The constraints affecting the manipulator are also an important point for global performance. This problem is traditionally accomplished by the trajectory planner. However, constraints violation can occur if, for instance, unbounded uncertainties, unknown payload mass or disturbances are present.

This work gives a global solution to these control problems, thus increasing the performance of the closed-loop system. Particularly, some new control strategies for constrained manipulators in presence of unmodelled uncertainties and unbounded disturbances are defined. They are based on the application of an efficient and low-cost predictive controller and an adaptive robust term for disturbances and uncertainties rejection

This paper is presented as follows: next section shows MPC definition and its main drawbacks. Section 3 describes a low-cost solution to avoid constraints violation. Sections 4 and 5 present the new proposed strategies: IAPC and RIAPC. The results obtained with these controllers are shown and compared with existing controllers in section 6. Finally, some conclusions are presented.

2. ADDING CONSTRAINTS TO CONTROL ALGORITHM FOR MANIPULATORS: MPC

As it is well known, the state of the robot is restricted by mechanical constraints (maximum values for the positions and velocities of the links). The maximum input to the actuators is also important because input saturations could cause instability of the closed-loop system or poor tracking performance. Although several approaches exist in the literature, constrained control of robot manipulators has not been solved efficiently in terms of computational load.

MPC is a simple and efficient way to take into account the constraints in the control algorithm. The algorithm has to solve on-line an optimisation problem which involves system constraints. Due to the need of small sampling times, in order to get an accurate prediction model, the optimisation problem has to be solved with a low-cost algorithm.

The two main drawbacks of the MPC formulation are the feasibility of the MPC solution and the computational load of the strategy. In one hand, feasibility is assured if an infinite horizon problem is solved, but this is not applicable in a real system. The other problem is the computational load due to complexity and high dimension of the optimisation problem.

3. LOW-COST MPC ALGORITHM

In order to solve the drawbacks seen in the previous section, this work proposes an efficient, low-cost strategy, based on interpolation between two or more simple solutions, which reduces the MPC computational cost without changing its main properties like optimality, performance and feasibility.

The original MPC unconstrained problem with infinite horizon leads to the optimal, linear and quadratic solution LQ. In order to consider the constraints, several suboptimal strategies have been proposed (Kouvaritakis *et al.*, 1997; Scokaert and Rawlings, 1998) based on the reduction of the

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.3758

dimension of the problem *N*. Its main drawback is that high values of N are needed in order to get a feasible solution, so the computational load of the algorithm strongly increases. To accomplish this, some strategies based on interpolations between the LQ solution and the, so called, "mean level" (ML) or the "tail" solutions are proposed (Rossiter *et al.*, 1998; Méndez *et al.*, 2000). This results on several algorithms, referred as LM (LQ+ML), LT (LQ+Tail), and LMT (LQ+ML+Tail).

3.1 Interpolation based predictive control

Some basics of these interpolation algorithms are exposed below. The control problem is, given the state vector at time instant k ($\mathbf{x}_k$), and the input vector ($\mathbf{u}_k$), to find the input sequence that minimizes a cost function like this:

$$J = \sum_{k=0}^{\infty}\left(\mathbf{x}_{k+1}^T V \mathbf{x}_{k+1} + \mathbf{u}_k^T W \mathbf{u}_k\right) \tag{1}$$

with constraints:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{2}$$

$$\begin{aligned} x_{i,\min} \le x_{i,k} \le x_{\mathrm{i,max}} \quad , \quad i = 1,\ldots,n \\ \mathrm{u}_{\mathrm{i,min}} \le u_{i,k} \le \mathrm{u}_{\mathrm{i,max}} \quad , \quad i = 1,\ldots,m \end{aligned} \tag{3}$$

where the sub-indexes *min* y *max* refer to the minimal and maximum values of each variable. The linear model of the system, needed to compute the predictions for the inputs, is added as an equality constraint. The strategy of the predictive controller consists of computing the input sequence $\{\mathbf{u}_k, k = 0,\ldots,n_p\}$ at each sample time that minimizes (1), being n_p the prediction horizon. Once the optimisation is performed, only the first input value of the sequence u_0 is applied and the procedure is repeated at the next sample time.

Without inequality constraints, the optimisation leads to the solution $\mathbf{u}_k = -\mathbf{K}_{LQ}\mathbf{x}_k$, where the gain $\mathbf{K}_{LQ}$ is obtained by considering the minimisation of (1), subject to the constraints (2), by using a Ricatti formulation (Lewis, 1984).

3.2 LQ+ML interpolation (LM)

The LQ is the optimal solution if it is feasible, but this does not usually occur. Let's call mean level solution, $\mathbf{u}_{ML}$, to the solution obtained by considering the minimisation of the cost function (1) when the weight of the command, R, is much higher than the weight of the state Q. In this way, the feasibility of the solution is always guaranteed. The LM algorithm consists of doing an interpolation between the LQ solution and the ML solution as follows:

$$\mathbf{u}_{k,LM} = (1-\alpha)\mathbf{u}_{k,LQ} + \alpha\mathbf{u}_{k,ML} \quad , \quad 0 \le \alpha \le 1 \tag{4}$$

and doing the minimisation of (1) with respect to $u_{k,ML}$ to compute the α value. The input predictions are given by:

$$\mathbf{u}_{k,LM} = -\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}^k(1-\alpha)\mathbf{x}_0 - \mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}^k\alpha\mathbf{x}_0 \tag{5}$$

where $\mathbf{K}_{LQ}$ is the gain obtained from the optimal control problem without inequality constraints and $\mathbf{K}_{ML}$ is the gain corresponding to the ML problem. Moreover, $\boldsymbol{\varphi}_{LQ} = \mathbf{A} - \mathbf{B}\mathbf{K}_{LQ}$ and $\boldsymbol{\varphi}_{ML} = \mathbf{A} - \mathbf{B}\mathbf{K}_{ML}$, where $\mathbf{A}$ y $\mathbf{B}$ are the state matrixes corresponding to the system (2).

3.3 Other interpolation algorithms

To solve the problem of the convergence of the cost function, the addition of the "tail" to the previous controllers has been proposed. The tail is the solution (predicted inputs) obtained in the previous time instant, except for the applied input. As it belongs to a feasible control law that minimises (1), it is a feasible sub-optimal solution for the input in the next time instant. This helps cost function to converge.

With this solution, two interpolation methods have been proposed: the LT (LQ+Tail) and the LMT (LQ+ML+Tail) algorithms. These interpolation algorithms (Mendez *et al.*, 2000, Kouvaritakis *et al.*, 2000) are not used in this work because the LM algorithm is enough to achieve the control specifications.

4. EFFICIENT CONTROL ALGORITHM FOR MANIPULATORS WITH CONSTRAINTS: IAPC

This section shows the controller used in this work to accomplish the control of the manipulator in the presence of disturbances, uncertainties and constraints. This method is called IAPC: Interpolation based Adaptive Perturbation Controller. It is based on the well known adaptive perturbation controller, where the linear contribution to the control law is given by an interpolation predictive controller.

4.1 Adaptive perturbation scheme

The control technique applied in this work is based on the scheme shown in figure 1. This method uses a linearisation of the model around the desired (or nominal) trajectory. Then, the torque applied to the links has two contributions: a direct contribution, calculated from the computed torque controller, and a feedback contribution, where a linear interpolation predictive controller, using the linearised model of the plant, tries to correct the deviations from the nominal trajectory, obtained by means of a trajectory planner. The non-linear control problem of the robot arm is then reduced to a linear control problem with respect to a nominal trajectory.

In this scheme, $\boldsymbol{\tau}_k = [\tau_{1,k},\ldots,\tau_{n,k}]^T$ are the input torques to the system and $\boldsymbol{\tau}_{\mathbf{N},k}$ are the nominal torques. The inputs applied to the system are given by:

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{\mathbf{N},k} + \tilde{\boldsymbol{\tau}}_k \tag{6}$$

where $\tilde{\boldsymbol{\tau}}_k$ are the feedback torques computed by the predictive control algorithms proposed in 4.3.

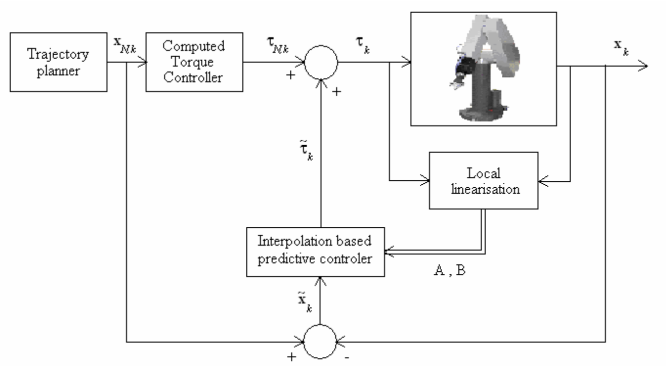

Fig. 1: Control scheme applied in this work: adaptive perturbation controller.

4.2 Prediction model

To obtain the linear model of the system, a local linearisation method is used. Following this method, once defined the state of the manipulator $\mathbf{x} = [x_1 \ldots x_{2n}]^T$, with $x_1 = \theta_1,\ x_2 = \dot{\theta}_1, \ldots, x_{2n-1} = \theta_n,\ x_{2n} = \dot{\theta}_n$, being n the number of degrees of freedom of the structure, $\theta_i, i = 1 \ldots n$, the positions, and $\dot{\theta}_i, i = 1 \ldots n$ the velocities of the links, and considering the current manipulator state as the equilibrium point, a first order Taylor expansion gives the following equation for the linearised model:

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}\tilde{\mathbf{x}}_k + \mathbf{B}\tilde{\boldsymbol{\tau}}_k \tag{7}$$

where $\mathbf{A}$ y $\mathbf{B}$ are the state matrixes of the approximated linear model. This linear approximation is valid for the deviations with respect to the nominal trajectory.

4.3 Optimisation problem

The feedback torques are computed by applying a linear predictive controller based on interpolation. Avoiding the notation for the time instant, the control law is:

$$\tilde{\boldsymbol{\tau}} = (1-\alpha)\tilde{\boldsymbol{\tau}}_{LQ} + \alpha\tilde{\boldsymbol{\tau}}_{ML} = (1-\alpha)\mathbf{K}_{LQ}\tilde{\mathbf{x}} + \alpha\mathbf{K}_{ML}\tilde{\mathbf{x}} \tag{8}$$

Input and state predictions are given by:

$$\tilde{\tau}(k+j|k) = (1-\alpha)\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} + \alpha\mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}} \tag{9}$$

$$\tilde{\mathbf{x}}(k+j|k) = (1-\alpha)\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} + \alpha\boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}} \tag{10}$$

with $j = 0,\ldots,n_c$, being n_c the control horizon. In this case, the constraints are the maximal input to the manipulator and the maximal allowed deviation with respect to the reference trajectory. Let $\tilde{\mathbf{x}}^{+}_{\max} = \max\{\mathbf{x} - \mathbf{x}_N\}$ be the maximal positive overshooting allowed for the state with respect to the reference trajectory $\mathbf{x}_N$ and $\tilde{\mathbf{x}}^{-}_{\max} = \max\{\mathbf{x}_N - \mathbf{x}\}$ the maximal negative overshooting. Let $\boldsymbol{\tau}_M$ be the maximal value for the input (and $-\boldsymbol{\tau}_M$ the minimal). After some simple steps, the optimisation problem to solve is the following linear programming problem:

$$\begin{aligned}
&\min_{\alpha} \alpha \\
&\textit{subject to}: \\
&\left(-\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}}\right)\ \alpha \le \tilde{\mathbf{x}}^{+}_{\max} - \boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} \\
&-\left(-\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} + \boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}}\right)\ \alpha \le -\tilde{\mathbf{x}}^{-}_{\max} + \boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} \\
&\left(-\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} - \mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}}\right)\ \alpha \le \\
&\qquad\qquad \boldsymbol{\tau}_M - \boldsymbol{\tau}_N - \mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} \\
&-\left(-\mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} - \mathbf{K}_{ML}\boldsymbol{\varphi}_{ML}{}^{j}\tilde{\mathbf{x}}\right)\ \alpha \le \\
&\qquad\qquad -\boldsymbol{\tau}_M + \boldsymbol{\tau}_N + \mathbf{K}_{LQ}\boldsymbol{\varphi}_{LQ}{}^{j}\tilde{\mathbf{x}} \\
&0 \le \alpha \le 1
\end{aligned} \tag{11}$$

The value α_o obtained when solving (11) is used in the current time instant in the control law (8). This assures an optimal performance of the controller without violating the constraints. The full algorithm IAPC is exposed below:

Step 0) Fix the constraints values $\tilde{\mathbf{x}}^{+}_{\max}$, $\tilde{\mathbf{x}}^{-}_{\max}$ and $\boldsymbol{\tau}_M$.

Step 1) Compute the nominal torques $\boldsymbol{\tau}_N$ using the reference trajectory given by $\mathbf{x}_N = \left[x_{1,d} \ldots x_{2n,d}\right]^T$ and $\ddot{\boldsymbol{\theta}}_d$. Set k=1.

Step 2) Measure $\boldsymbol{\xi}_k = \begin{bmatrix}\tilde{\boldsymbol{\theta}}_k & \dot{\tilde{\boldsymbol{\theta}}}_k\end{bmatrix}^T$, with $\tilde{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_{k,d} - \boldsymbol{\theta}_k$ and $\dot{\tilde{\boldsymbol{\theta}}}_k = \dot{\boldsymbol{\theta}}_{k,d} - \dot{\boldsymbol{\theta}}_k$.

Step 3) Evaluate the linearised model at the current time.

Step 4) Obtain the gains $\mathbf{K}_{LQ}$ and $\mathbf{K}_{ML}$ solving the corresponding unconstrained infinite-time optimisation problems.

Step 5) Obtain $\boldsymbol{\varphi}_{LQ} = \mathbf{A} - \mathbf{B}\mathbf{K}_{LQ}$ and $\boldsymbol{\varphi}_{ML} = \mathbf{A} - \mathbf{B}\mathbf{K}_{ML}$.

Step 6) Solve (11) to obtain the optimal value α_o of the interpolation parameter α.

Step 7) Evaluate the input (8) and apply to the system.

Step 8) Set k=k+1 and wait until the following sampling time. Go to Step 2.

As can be observed, nominal torques are computed off-line using the reference trajectory. This reduces the computational cost of the algorithm. Once the linear model is obtained, the

$\mathbf{K}_{LQ}$ and $\mathbf{K}_{ML}$ gains have to be computed. This does not add computational cost because the simple and efficient Riccati algorithm is employed. The problem with respect to the computational cost is the solution of the linear programming problem. Its complexity increases with the control horizon n_c because of the number of constraints to evaluate. One strategy to reduce the computational cost of the algorithms consist of evaluating the linearisation every several sampling times, maintaining the current linearised model over a period of time in which the gains and the variables $\boldsymbol{\varphi}_{LQ}$ and $\boldsymbol{\varphi}_{ML}$ are not changed. Thus, the linear programming problem only changes in the new state values.

5. ADDING ROBUSTNESS TO CONTROLLER: RIAPC

The previous method can be improved by adding a term to reinforce the robustness of the controller. The typical robust control algorithms for uncertainties and disturbances rejection are based on the Lyapunov Direct Method and also on the Variable Structure Sliding strategy. First of them gives an expression for the controller as the following:

$$\tilde{\boldsymbol{\tau}}_k = \tilde{\boldsymbol{\tau}}_{k,s} + \tilde{\boldsymbol{\tau}}_{k,r}$$

$$\tilde{\boldsymbol{\tau}}_{k,r} = \begin{cases} \dfrac{\rho_k}{\left\| \mathbf{D}^t \mathbf{Q} \boldsymbol{\xi}_k \right\|} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k & , \ if \ \left\| \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k \right\| \geq \varepsilon \\ \dfrac{\rho_k}{\varepsilon} \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k & , \ if \ \left\| \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k \right\| < \varepsilon \end{cases} \tag{12}$$

where $\mathbf{D} = [\mathbf{0} \quad \mathbf{I}]^T$, being $\mathbf{0}$ and $\mathbf{I}$, respectively, a zero matrix and the identity matrix of adequate dimensions, $\mathbf{Q}$ is a definite positive matrix and ρ is referred as the uncertainty bound parameter. The robust term $\tilde{\boldsymbol{\tau}}_{k,r}$ is added to a stabilising control law $\tilde{\boldsymbol{\tau}}_{k,s}$ (PD, PD plus gravity compensation, LQ, etc.) to add robustness to the controller in presence of model uncertainties (Spong, 1992). The discontinuous definition of the robust term comes from the avoidance of the problem of chattering when the tracking error tends to zero. This law assures that the trajectories tend to a hyperplane around the sliding subspace $\mathbf{z}_k = \mathbf{D}^T \mathbf{Q} \boldsymbol{\xi}_k = \mathbf{0}$. The hyperplane dimensions depend on the parameter ε.

The performance of this controller depends on the uncertainty bound parameter ρ_k. This parameter is tuned using information of the bounds for the model uncertainties. But in certain circumstances this is not possible, for instance, when an unknown payload mass is manipulated by the robot.

A recent work (Torres *et al*, 2006) proposes a new method for on-line tuning of the parameter ρ_k. The adaptation law of the parameter ρ_k is based in a descendent gradient method, which updates it to get a good rejection of the uncertainties and disturbances. It is given by:

$$\rho_k = \rho_{k-1} - \gamma \frac{\partial J_k}{\partial \rho_{k-1}} \tag{13}$$

where $J_k(\rho_{k-1})$ is an adequate quadratic cost function and γ is the learning rate of the adaptation law. The steps to obtain the expression for the adaptation law can be seen in Torres *et al* (2006). Finally, the adaptation law is given by:

$$\rho_k = \rho_{k-1} - \gamma \left[\boldsymbol{\xi}_k{}^T \mathbf{Q}_{ad} \frac{\partial \boldsymbol{\xi}_k}{\partial \rho_{k-1}} + \tilde{\boldsymbol{\tau}}_{k-1}{}^T \mathbf{R}_{ad} \frac{\partial \tilde{\boldsymbol{\tau}}_{k-1}}{\partial \rho_{k-1}} \right] \tag{14}$$

where $\mathbf{Q}_{ad}$ and $\mathbf{R}_{ad}$ are the matrixes of $J_k(\rho)$, and:

$$\frac{\partial \boldsymbol{\xi}_k}{\partial \rho_{k-1}} = -\begin{bmatrix} \mathbf{CBM}\boldsymbol{\xi}_{k-1} \\ \dfrac{1}{h}\mathbf{CBM}\boldsymbol{\xi}_{k-1} \end{bmatrix} \quad ; \quad \frac{\partial \tilde{\boldsymbol{\tau}}_{k-1}}{\partial \rho_{k-1}} = \mathbf{M}\boldsymbol{\xi}_{k-1} \tag{15}$$

$$\mathbf{M} = \mathbf{D}^T \mathbf{Q} / \varepsilon \tag{16}$$

being $\mathbf{B}$ and $\mathbf{C}$ the matrixes obtained from the state equation (7) and the output equation $\tilde{\boldsymbol{\theta}}_k = \mathbf{C}\tilde{\mathbf{x}}_k$.

In this work, the term $\tilde{\boldsymbol{\tau}}_{k,s}$ is computed by the IAPC algorithm. The resulting control algorithm, referred as RIAPC (Robust IAPC), rejects efficiently the uncertainties maintaining the properties of feasibility and low computational cost.

6. RESULTS

The controller defined in this work gives an efficient solution to the problem of a manipulator in presence of uncertainties, disturbances and constraints. The linear predictive controller takes into account the constraints which are present in the system, normally input saturations due to actuators and state constraints due to mechanical aspects.

In the standard solution, the trajectory planner gives a reference far enough from the constraints, even in presence of uncertainties of the assumed model of the manipulator. But if there are unmodelled or unbounded uncertainties (i.e. very different and unknown values of the payload mass), or in presence of disturbances, the constraints can be violated. In this case, the proposed IAPC and RIAPC strategies give some low-cost, feasible and efficient solutions.

Next subsections show the benefits of this strategy, comparing the results between different controllers with the results obtained with the IAPC and RIAPC. First, an LQ controller with a robust term is used. Then, the IAPC strategy is used. Finally, in case of uncertainties, RIAPC strategy is used to improve the previous controller. To verify the performance of these strategies, some simulations on a PUMA-560 robot, with and without uncertainties, are shown.

6.1 Robot model without uncertainties

In this case, the model used to compute the manipulator dynamics is also used as prediction model. The disturbances affecting the system lead to constraints violation in the standard solution. This simulation shows that the IAPC (and

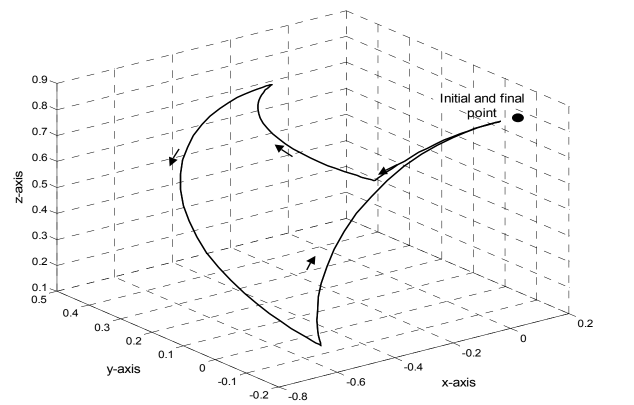

Fig. 2: Reference trajectory represented in the task space of the PUMA-560 manipulator.

RIAPC) algorithms avoid the constraints violation. The parameters of the controller are the following. For the LQ controller, the weight matrices are **V** = diag(0.04*[120 30 120 30 120 30 80 20 80 20 80 20]) and **W** = diag(0.0005*[5 5 5 1 1 1]). These values are relaxed with respect the optimal values in order to clearly show the comparison between the controllers. For the LM controller, the weight matrixes are **V**'=0.1***V** and **W**'=1e4***W**, to guarantee a feasible solution. For the robust term, the parameters are ε=1 and **Q** =diag([0 0 0 0 0,1 1 1 1 1 1]). This term is adapted on-line, using the parameters: γ =1, $\mathbf{R}_{ad}$ =h*diag(25*[1 1 1 1 1 1]), and $\mathbf{Q}_{ad}$ =h*diag(100*[1 3.75 2.5 2.5 2.5 2.5 2.5 6 5 6 6 6]), being h=0.001 the sampling time. The reference trajectory is a typical pick-and-place trajectory and has been generated by joining four points in the task space of the PUMA manipulator. This trajectory can be seen in figure 2. And additive disturbance in the positions is added at t=4.8 sec and t=16.0 sec. The input and state constraints values are $\boldsymbol{\tau}_M = \begin{bmatrix}75 & 36 & 11 & 8 & 8 & 8\end{bmatrix}^T$ and $\tilde{\mathbf{x}}^+_{\max} = -\tilde{\mathbf{x}}^-_{\max} = \begin{bmatrix}0.4 & 0.4 & 0.4 & 0.1 & 0.1 & 0.1\end{bmatrix}^T$.

Figures 3 and 4 show respectively the results obtained with the second and third links of the manipulator. The other links have a similar performance. It can be seen that, around t = 5.0 sec., these links have a better performance with the IAPC than the LQ controller because input saturation is avoided. That saturation is produced by the presence of disturbances in the position values. LQ controller gives input saturation which carries the link to a maximum tracking error of 0.20 rads. In the case of IAPC controller, the input saturation is avoided by means of the interpolation with the ML solution. The tracking error is reduced nearly 25% (around 0.15 rads.). RIAPC strategy offers a little better performance due to the efficiency of the controller against the perturbations. But this effect will be clearly shown in the next subsection, where parametric uncertainties are taken into account.

6.2 Robot model with parametric uncertainties

This section shows the results with the same controller but considering parametric uncertainties in the model. These are deviations of the masses and lengths with respect to the real values, and the presence of payload unmodelled masses. The simulation is quite similar to the previous case, so the constraints violation is again produced by the disturbances during the experiment.

Figures 5 and 6 show a different performance with respect to the previous section. Again in the second and third links, around t=5.0 sec., IAPC and RIAPC strategies offer a better performance than the LQ controller because they avoid the input saturation. But, in this case, the system with RIAPC strategy has a better performance due to the action of the robust term over the deviations produced by the uncertain predictions computed by the model. With IAPC strategy, the tracking errors produced after t=5.0 sec are higher than with the robust strategy RIAPC.

Figure 7 shows the evolution of the uncertainties bound parameter in case of LQ controller and RIAPC strategy. As it can be seen, they are very similar but, due to the interpolation strategy produces smaller tracking error values, the evolution of this parameter in the second case is smoother.

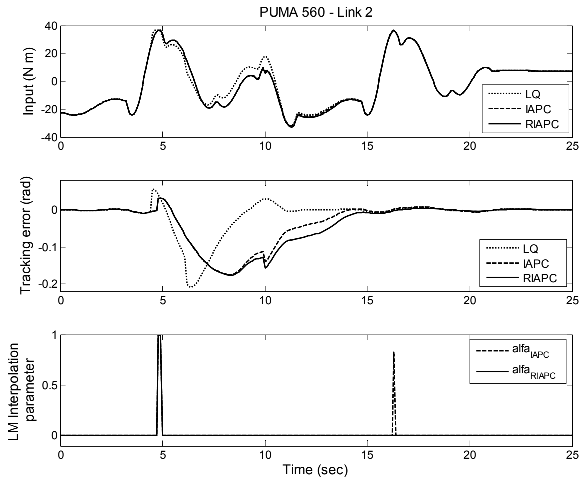

Fig. 3: Evolution of the input, tracking error and interpolation parameter for the second link. No uncertainties are present in the model.

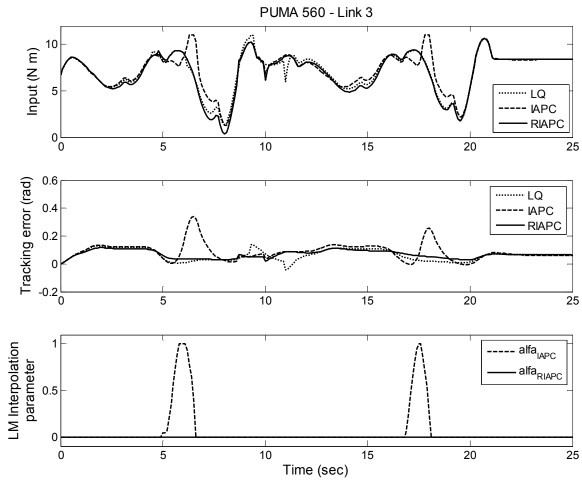

Fig. 4: Evolution of the input, tracking error and interpolation parameter for the third link. No uncertainties are present in the model.

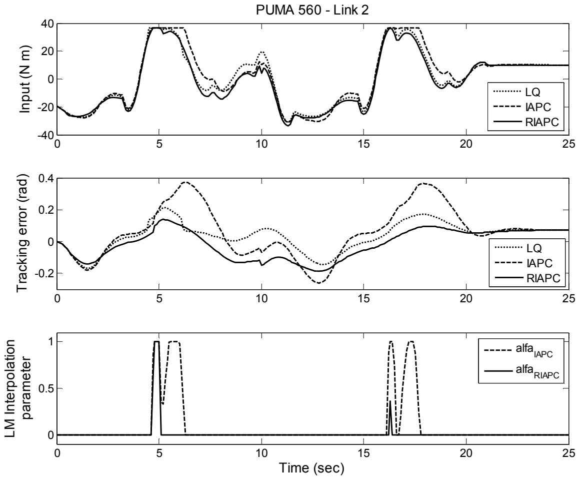

Fig. 5: Evolution of the input, tracking error and interpolation parameter for the second link. The model presents parametric uncertainties.

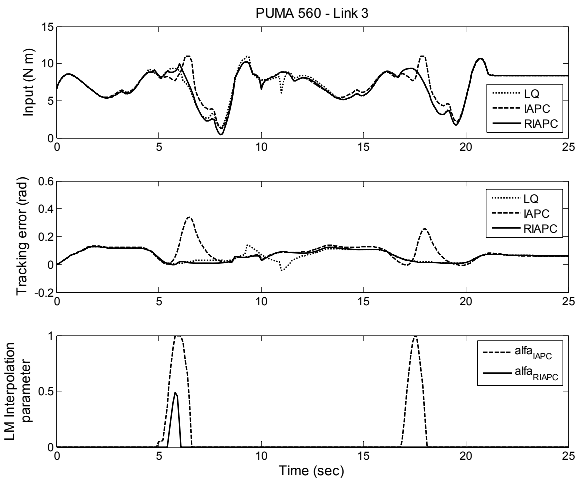

Fig. 6: Evolution of the input, tracking error and interpolation parameter for the third link. The model presents parametric uncertainties.

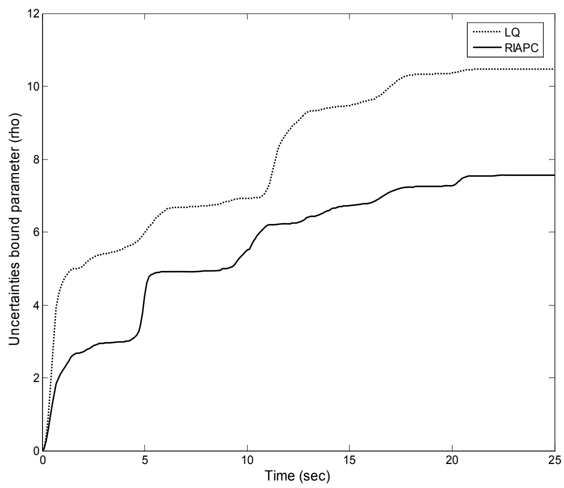

Fig. 7: Evolution of the uncertainties bound parameter for the LQ controller and RIAPC strategy in the simulations of figures 5 and 6.

7. CONCLUSIONS

Most of the high performance solutions proposed for manipulators control are difficult to implement. Their main drawbacks are the computational load and the poor performance in presence of unbounded uncertainties and disturbances.

The algorithm proposed here combines the following three main properties: constraints handling, uncertainties and disturbances rejection, and low computational cost. The controller incorporates MPC techniques into an adaptive perturbation controller. The predictive controller is based on interpolation. A local linearisation algorithm is used in order to obtain a linear model of the system. With this scheme, the violation of the constraints is avoided. Additionally, a robust term is added to the control law in order to reject the model uncertainties. This term is adaptively changed for a better performance of the closed-loop where the uncertainties and disturbances are completely unknown and/or unbounded.

Results included in this work show satisfactory performance of this strategy compared with other controllers in a PUMA-560 manipulator. For the proposed simulations, the tracking errors were significantly reduced with respect to the alternative methods. The feasibility of the solution was also verified in the simulations.

REFERENCES

Alonge, F., F. D'Ippolito and F.M. Raimondi (2004). Globally Convergent Adaptive and Robust Control of Robotic Manipulators for Trajectory Tracking. *Control Engineering Practice*, **12** (9), 1091-1100.

Kouvaritakis, B., J.A. Rossiter, M. Cannon and C.K.J. Wong (1997). Constrained stable predictive control: an l_2-optimal approach. *5th IEEE MED Conference*, Cyprus.

Kouvaritakis, B., J.A.Méndez and J.A.Rossiter, (2000) Application of MPC with interpolation to a Crane Problem. PCI 2000. Scotland. UK.

Lewis,F.L.(1984). *Optimal Control*, 49-52. Wiley, NewYork.

Méndez, J.A., B. Kouvaritakis and J.A. Rossiter (2000) State-space approach to interpolation in MPC. *Int. Journal of Robust and Nonlinear Control*, **10**, 27-38.

Rossiter, J.A., M.J. Rice, J. Schuurmans and B. Kouvaritakis (1998). A computationally efficient constrained predictive control law. *ACC 98*, Philadelphia, USA.

Scokaert, P.O.M. and B. Rawlings (1998). Constrained linear quadratic regulation. IEEE Trans. *AC*, **43**(8), 1163-1169.

Spong, M.W. (1992). On the robust control of robots manipulators. *IEEE Trans. Aut. Control*, **37,** 1782-1786.

Spong, M.W. (2003). Robust and Adaptive Motion Control of Manipulators. Personal Communication.

Torres, S., J.A. Méndez, L. Acosta and V.M. Becerra (2006). Adaptive robust controller for robot manipulators: experiments on a PUMA 560 robot. *8th Int. IFAC SYROCO 2006*, Bologna, Italy.

Whitcomb, L.L., A.A. Rizzi and D.E. Koditschek (1993). Comparative Experiments with a New Adaptive Controller for Robot Arms. *IEEE Transactions on Robotics and Automation*, **9**(1), 59-70.