15th Triennial World Congress, Barcelona, Spain

DISTURBANCES REJECTION ON A ROBOT ARM USING AN EFFICIENT PREDICTIVE CONTROLLER

S. Torres, J.A. Méndez, L. Acosta, M. Sigut, G.N. Marichal

Automation and Control Group, Department of Applied Physics, University of La Laguna, Tenerife, SPAIN.

Abstract: This work proposes a predictive controller with interpolation in order to improve the behaviour of a typical MPC when the system presents constraints. Particularly, it is interesting to see how the interpolation, between the solution of the optimal unconstrained problem and other feasible solutions, assures the stability of the system in presence of disturbances. The system in which the controller is applied is a two-link robot manipulator arm. The predictive controller is inserted in an adaptive perturbation scheme to change adequately the nominal inputs, given by an inverse dynamics controller, in order to reject the disturbances produced. The efficiency of the proposed strategy is shown by simulation.

Keywords: predictive control, interpolation algorithms, constraints, robot arms, optimal control, robustness.

1. INTRODUCTION

Traditionally, the predictive control theory has been applied on systems in which the dynamics are slow, e.g., an industrial plant. The time spent in doing the computation of the algorithms should not be a problem in these cases, where sampling rates of several seconds can be used. However, in order to implement them in a system with fast dynamics, e.g. a robot manipulator, some new efficient and low computation time strategies must be introduced.

The original MPC unconstrained problem with infinite horizon leads to the optimal, linear and quadratic solution LQ. In order to consider the constraints, some works (Kouvaritakis, *et al.*, 1997; Scokaert and Rawlings, 1998) successfully propose several sub optimal strategies, based on reducing the dimension of the problem N. The drawback of these strategies is that high values of N are needed in order to get a feasible solution, so the computational load of the algorithm strongly increases. Recently, some strategies based on interpolations between the LQ solution and the “mean level” (ML) or the “tail” solutions are proposed (Kouvaritakis, *et al.*, 1998; Rossiter, *et al.*, 1998). However, they have some drawbacks. For the first algorithm, referred as LM (LQ+ML), cost function convergence is not guaranteed, so the system can result unstable. The second algorithm, referred as LT (LQ+Tail), presents robustness problems when uncertainty in the model is present or in the presence of disturbances, because they can make the tail unfeasible. This leads to an unfeasible solution and causes closed-loop instability. In the work of Méndez, *et al.* (2000), these algorithms are improved by interpolating between the three solutions: LQ, ML and the tail (this strategy will be referred as LMT: LQ+ML+Tail). The optimisation process leads to a two dimensional quadratic problem. This algorithm presents convergence, optimality, feasibility and better robustness properties with respect to disturbances than the other formulations.

In this work the implementation of the LMT algorithm on a two-link manipulator robot arm is performed. The goal is to drive the links along some planned trajectories by inserting the system in an adaptive perturbation scheme.

2. ROBOT MODEL

The model of the robot arm used in this work is obtained from the second and third link of a PUMA robot arm (see figure 1). The dynamics of this system are represented by the following set of high non-linear and coupled differential equations:

$$u(t) = D(\theta(t))\ddot{\theta}(t) + h(\theta(t), \dot{\theta}(t)) + c(\theta(t)) \quad (1)$$

being $D(\theta(t))$ the inertia matrix, $h(\theta(t), \dot{\theta}(t))$ the Coriolis and centrifugal force vector, $c(\theta(t))$ the gravitational force vector and $u(t) = [u_1(t) \quad u_2(t)]^T$ the applied torque to each link. The parameters of the links considered in this work are 5.0 kg and 4.5 kg for the masses of the links and 0.43 m for its lengths.

3. CONTROL SCHEME

The control technique applied here is based on the scheme shown in figure 2. This method uses a linearisation of the model around the desired (or nominal) trajectory. Then, the torque applied to the link has two contributions: a direct contribution, calculated from the inverse dynamics equations; and a feedback contribution, where a linear controller, using the linearised model of the plant, tries to correct the deviations from the nominal trajectory. 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 paper, a local linearisation, described in the next section, is used to obtain the linearised model of the robot manipulator at each instant of time. On the other hand, a predictive controller with interpolation is used to compute the feedback command contribution.

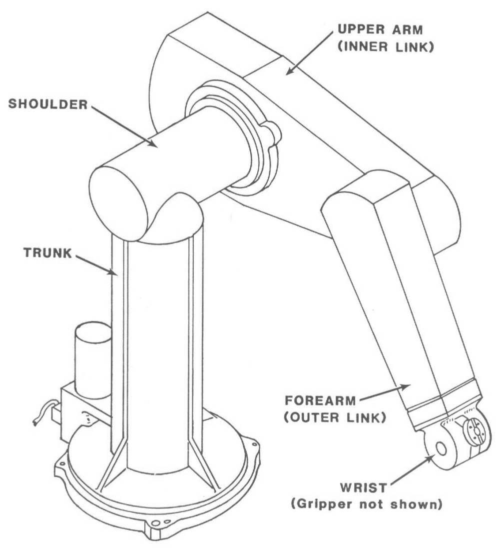

Fig. 1. PUMA robot arm from 560 series.

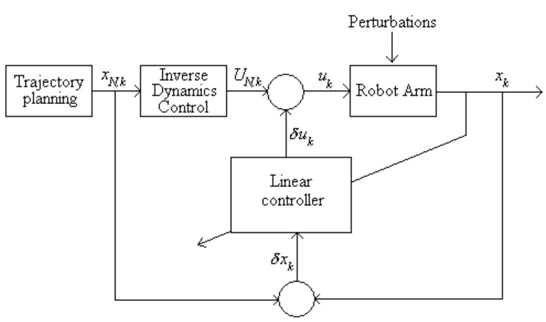

Fig. 2. Adaptive perturbation controller.

4. MODEL LINEARISATION

In order to apply the predictive control algorithms presented in this work, a linear model of the system is needed. In this case, a local linearisation is used. An approximate linear model of the non-linear system, valid for deviations of the trajectory with respect the nominal one, is obtained at each point.

To obtain it, consider the following state variables:

$$\begin{array}{ll} x_1 = \theta_1 & x_2 = \dot{\theta}_1 \\ x_3 = \theta_2 & x_4 = \dot{\theta}_2 \end{array} \quad (2)$$

where θ_1 and θ_2 are respectively the angles of the two links. In vector notation, we have:

$$\dot{\mathbf{x}} = \mathbf{F}(x_1, x_2, x_3, x_4) \quad (3)$$

with $\mathbf{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$, $\mathbf{F} = [f_1 \quad f_2 \quad f_3 \quad f_4]^T$, and being $f_i(\mathbf{x,u})$ scalar functions. Computing a first order Taylor expansion, considering the current manipulator state as the equilibrium point, the following equation is obtained:

$$\delta\dot{\mathbf{x}} = A\delta\mathbf{x} + B\delta\mathbf{u} \quad (4)$$

where A and B are the state matrixes of the approximated linear model, $\mathbf{u} = [u_1 \quad u_2]^T$ are the applied torques to the system and $\delta\mathbf{u}$ are the deviations with respect to the nominal torque $\mathbf{U_N}$. These nominal torques are computed using the inverse dynamics equations (by using the Newton-Euler recursive algorithm) and they are the inputs needed to reach the next point of the planned (nominal) trajectory. This linear approximation is valid for the trajectory deviations with respect to the nominal one, obtained by means of a trajectory planner.

Then, the inputs applied to the system are given by:

$$\boldsymbol{u}_k = \boldsymbol{U}_{N,k} + \delta\boldsymbol{u}_k \quad (5)$$

where $\delta\mathbf{u}_k$ are the feedback torques computed by the predictive control algorithms proposed in next section. If the current state of the arm and the next desired state are given, the algorithm gives the appropriate inputs to achieve it.

5. CONTROL ALGORITHMS BASICS

Let x_k be the state vector of the system at time instant k and u_k the input vector at the same instant. The control problem is to find the input sequence that minimizes the following cost function:

$$J = \sum_{k=0}^{\infty} \left(\mathbf{x}_k^t Q \mathbf{x}_k + \mathbf{u}_k^t R \mathbf{u}_k \right) \quad (6)$$

with constraints:

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

$$\begin{aligned} x_{i,min} \le x_i \le x_{i,\max} \quad &, \quad i = 1,\dots,n \\ \mathrm{u}_{i,\min} \le u_i \le \mathrm{u}_{i,\max} \quad &, \quad i = 1,\dots,m \end{aligned} \tag{8}$$

where the sub-indices *min* and *max* refers 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.

5.1 LQ controller

The control law for this controller is given by:

$$\delta\mathbf{u}_k = -K_{LQ}\delta\mathbf{x}_k \tag{9}$$

where the gain K_{LQ} is obtained by considering the minimisation of (1), subject to the constraints (2) and (3), by using a Ricatti formulation (Lewis, 1984).

5.2 LM controller: interpolating the LQ solution with a feasible solution

The strategy of the predictive controller consists of computing the input sequence $\{u_i,\}$ at each sample time that minimizes (6). 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 constraints, the optimisation leads to the LQ solution, u_{LQ}. This solution is optimal if it is feasible.

Let's call mean level solution, u_{ML} , to the solution obtained by considering the minimisation of the cost function (6) 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:

$$\delta\mathbf{u}_{k,LM} = \left(1-\alpha\right)\delta\mathbf{u}_{k,LQ} + \alpha\delta\mathbf{u}_{k,ML} \ \ , 0 \le \alpha \le 1 \tag{10}$$

and doing the minimisation of (6) with respect to $\delta u_{k,LM}$ in order to compute the α value. To use this algorithm, a division of the initial state is done:

$$x_0 = w_0 + z_0 \tag{11}$$

with

$$\begin{aligned} w_0 &= (1-\alpha)x_0 \\ z_0 &= \alpha x_0 \end{aligned} \tag{12}$$

The predictions for the input are given by:

$$\delta\mathbf{u}_{k,LM} = -K_{LQ}\phi_{LQ}^k w_0 - K_{ML}\phi_{ML}^k z_0 \tag{13}$$

where K_{LQ} is the gain obtained from the optimal control problem without constraints and K_{ML} is the gain corresponding to the ML problem. Moreover, $\phi_{LQ}=A-BK_{LQ}$ and $\phi_{ML}=A-BK_{ML}$, where A and B are the state matrixes corresponding to the system (4).

5.3 LMT Algorithm: two-dimensional interpolation

The previous algorithm has a main drawback: it does not assure the convergence of the cost function, so stability problems may occur. To solve this problem, the addition of the "tail" to the previous interpolation algorithm is proposed. Firstly, the concept of "tail" is introduced. In the current instant of time, the tail of the optimal input sequence u_i is:

$$u_{tail} = \{u_i\} \quad , \quad i = 1,2,\dots \tag{14}$$

this is, all expect for the first value u_0. As u_{tail} belongs to a feasible control law that minimises J, it is a feasible sub optimal solution for the input in the next instant of time. So this solution can be added to the solution of the previous algorithm by defining the solution:

$$\delta\mathbf{u}_{k,LMT} = \left(1-\alpha-\beta\right)\delta\mathbf{u}_{k,LQ} + \alpha\delta\mathbf{u}_{tail} + \beta\delta\mathbf{u}_{k,ML} \tag{15}$$

where $0 \le \alpha \le 1$, $0 \le \beta \le 1$ and $0 \le \alpha+\beta \le 1$.

The minimisation of the cost function (6) is done with respect to $\delta\boldsymbol{u}_{k,LMT}$ in order to obtain the interpolation parameters α and β. In this case, the initial state of the system x_0 is divided in three sub-states:

$$x_0 = w_0 + z_0 + v_0 \tag{16}$$

with:

$$\begin{aligned} w_0 &= (1-\alpha-\beta)x_0 + \beta\phi_{LQ}w_{-1} \\ z_0 &= \alpha x_0 \\ v_0 &= \beta\left(x_0 - \phi_{LQ}w_{-1}\right) \end{aligned} \tag{17}$$

and the predictions to the commands are given by:

$$\delta\boldsymbol{u}_{k,LMT} = -K_{LQ}\phi_{LQ}^k w_0 - K_{ML}\phi_{ML}^k\left(z_0 + v_0\right) \tag{18}$$

being K_{LQ}, K_{ML}, ϕ_{LQ} and ϕ_{ML} as in the previous case. This algorithm presents the following four desirable properties: convergence of the cost function, stability of the closed-loop system, feasibility and robustness against model uncertainties and/or disturbances.

5.4 Algorithm scheme

A general algorithm description is presented below.

Step 1) Set the sample time T and the final time of simulation T_{final} . Set $k = 0$.

Step 2) Compute the nominal trajectory and apply the inverse dynamics controller to compute the nominal inputs $\mathbf{U}_{\mathbf{N},k}$, $k = 1, 2, \dots$.

Step 3) $k = k + 1$

Step 4) Compute the linearised model of the system (matrices A and B) by using the current state, x_k , and the applied inputs, u_{k-1} .

Step 5)

Case i) *LQ controller*.

- Compute K_{LQ} from the unconstrained LQ problem using the weight matrices Q and R from (20).
- Compute $\delta\mathbf{u}_k$ by using (9).

Case ii) *LM controller*.

- Compute K_{LQ} and K_{ML} from the unconstrained LQ problem and, for the ML problem, using much more strong weights in R than in Q.
- Solve the linear programming problem (13) in order to compute α .
- Compute $\delta\mathbf{u}_k$ by using (10).

Case iii) *LMT controller*.

- Compute K_{LQ} and K_{ML} as in case ii.
- Solve the quadratic programming problem (18) in order to compute α and β .
- Compute $\delta\mathbf{u}_k$ by using (15).

Step 6) Form the input $\mathbf{u}_k$ by applying (5).

Step 7) GO Step 3 WHILE kT< T_{final}

The main on-line steps in the method are the computation of the matrices of the linearised model and the output of the predictive controller. Obviously, they have to be executed faster than the sampling time T.

A suitable value for this parameter is T=10 msec. The linearisation takes nearly 2 msec. Solving the unconstrained LQ problem, and the linear programming problem (if LM is considered) or the quadratic programming problem (if LMT is used), do not take much time if these routines are coded in an efficient manner (less than 6 msec). Thus T is large enough for the computations to take place.

6. RESULTS

The aim of this work is the implementation of the LMT predictive control algorithm on a two-link robot manipulator, and compares the results with those obtained from the LQ unconstrained optimal controller and with the LM controller. In the different simulations, the system is affected by an additive step perturbation on the state of each link with the same amplitude. In this case, only constraints in the inputs are considered. These constraints are:

$$\begin{aligned} \left|u_{k,1}\right| &\le 90 \quad N \cdot m \\ \left|u_{k,2}\right| &\le 15 \quad N \cdot m \end{aligned} \tag{19}$$

where the sub-indices 1 and 2 indicates respectively the links 1 and 2.

6.1 LQ Controller

First, the controller described in 5.1 is considered. The weight matrices Q and R used in the minimisation of (6) are presented in equation (20).

$$Q = \begin{pmatrix} 1500 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix} \;,\; R = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \tag{20}$$

The results obtained for the first link are shown in figure 3. The nominal contribution $\mathbf{U_N}$ to the input provides the necessary inputs to carry the link close to the desired trajectory (dotted line). When the perturbation affects to the system, this results unstable due to input constraints (8). Saturation of the applied inputs occurs to both links, so the system becomes unstable.

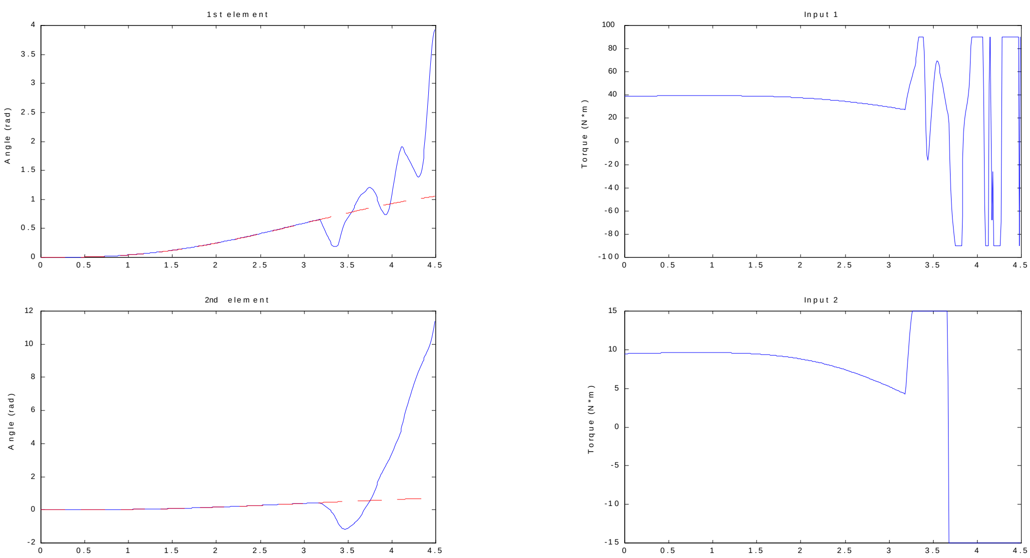

Fig. 3. LQ optimal controller applied on the two-link robot arm: evolution of the robot (left column) and resulting torques (rigth column). System affected by an additive disturbance between t=3.2 sec and t=3.4 sec.

6.2 LM Controller

Secondly, the control algorithm described in 5.2, based on an interpolation between LQ and ML solution, is used. The results obtained with this controller are shown in figure 4. The manipulator tries to reach the nominal trajectory after disturbance is produced. Input torques to the links always take values under the highest permitted ones. As shown, instability is avoided . As it can be seen, the LQ control law is applied as far as the disturbance is produced. In this moment, the parameter α varies from 0 to a higher value (see figure 6). The control law tends to the ML solution in order to get a better performance of the links without violation of the constraints. When the disturbance stops, the LQ solution is recovered because it becomes feasible again. The trajectories of both links tend to the respective nominal trajectories. The cost measured in this case is J=36442.

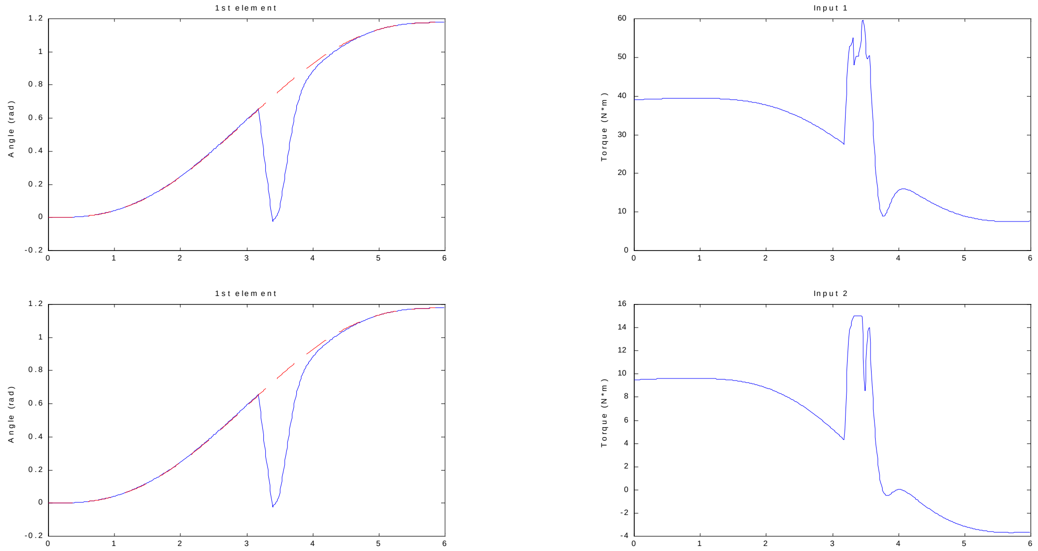

Fig. 4. LM controller applied on a two-link robot arm: evolution of the robot (left column) and resulting torques (right column). System affected by an additive disturbance between t=3.2 sec and t=3.4 sec.

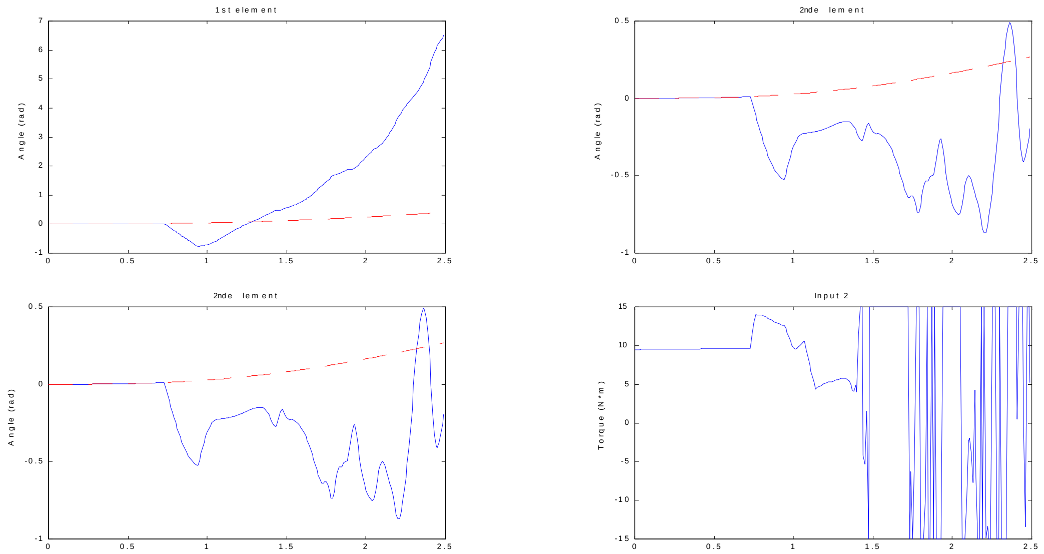

Fig. 5. LM controller applied on a two-link robot arm: evolution of the robot (left column) and resulting torques (right column). System affected by an additive disturbance between t=0.75 sec and t=0.95 sec.

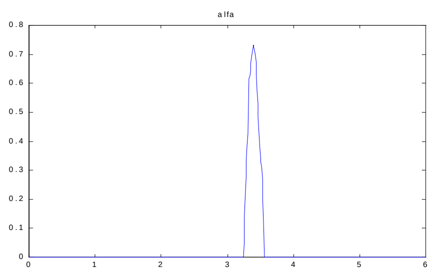

Fig. 6. Evolution of the parameter α for the simulation of figure 4.

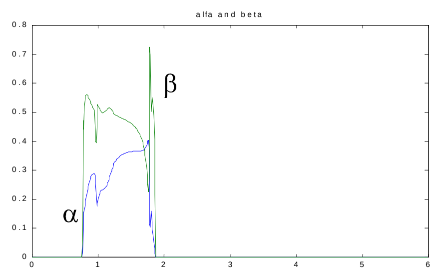

Fig. 7. Evolution of the parameters α and β for the simulation of figure 8.

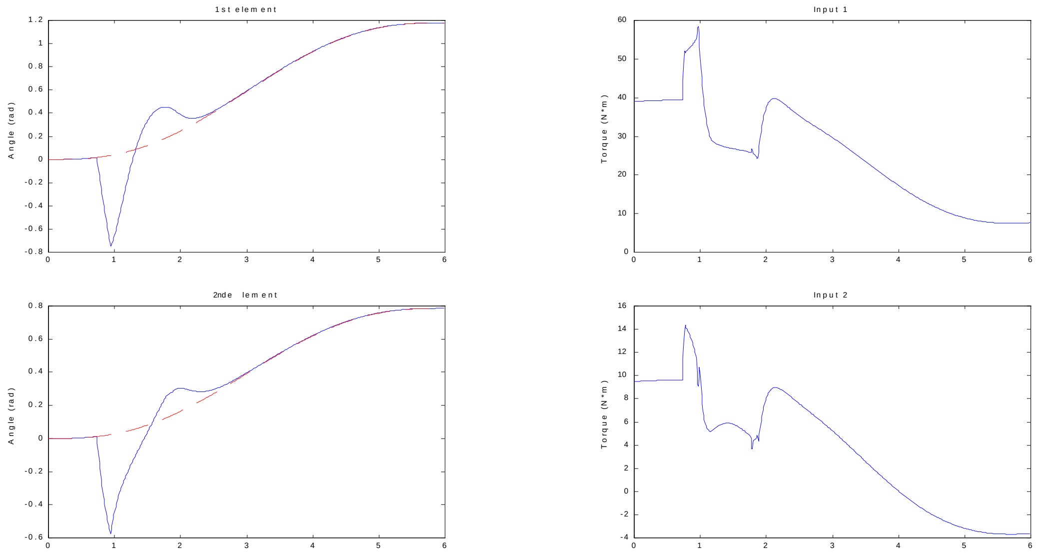

Fig. 8. LMT controller applied on the two-link robot arm: evolution of the robot (left column) and resulting torques (right column). System affected by an additive disturbance between the instants 1.0 sec. and 1.2 sec.

6.3 LMT Controller

Finally, the algorithm described in section 5.3 is considered. The performance of the system is similar to the previous case. The additive disturbance moves the system away from the desired trajectory. Once the disturbance stops, the algorithm recovers the nominal trajectory without any constraint violation. The cost measured in this case is J=35757. A reduction of 1.9% is obtained with respect the previous algorithm.

LMT algorithm assures an efficient behaviour of the system even when the LM algorithm fails. Figures 5 and 8 show a situation where the disturbance is applied between the instants t=0.75 sec and t=0.95 sec. The LM controller becomes infeasible once the disturbance is applied. Saturation of the input is produced and the system results unstable. However, with the LMT controller, both links tend efficiently to the desired trajectory once the disturbance stops. Input saturation is not observed, and then the system keeps stable during the simulation. Evolution of the parameters α and β can be seen in figure 7.

CONCLUSIONS

The aim of this work is to implement an efficient predictive controller, based on interpolating the optimal solution with feasible solutions, in a two-link robot manipulator. The control strategy is used with an adaptive perturbation scheme: a linear control law, which tries to beat off disturbances applied on the system, corrects the inputs computed by the application of inverse dynamics. This correction is carried out, in this case, by the LMT predictive controller. Advantages of this algorithm are its simplicity and the low computational cost. The optimisation is reduced to a quadratic programming problem. A linear model of the system is needed for the application of this algorithm. This linearisation was done at each point of the trajectory.

The results obtained show the advantages of LMT algorithm, with respect to the other strategies. The LQ and LM algorithms present feasibility problems, producing instability by command saturation, in presence of disturbances. However, the LMT algorithm rejects the disturbances without input constraints violations in the system. In this way, the system remains stable along the planned trajectory.

ACKNOWLEDGEMENTS

This work is financed by the Gobierno Autónomo de Canarias by the research project PI1999/140 titled "*Diseño e implementación hardware y software de un robot autónomo inteligente para la recogida de objetos. Aplicación a la recogida de fruta*".

REFERENCES

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. Rossiter and M. Cannon (1998). Linear quadratic feasible predictive control. *Automatica*, **34**(12), pp. 1583-1592.

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). In: *Optimal Control* (New York Chichester (Wiley)), 49-52.

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**, pp. 27-38.

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

P.O.M. Scokaert and B. Rawlings (1998). Constrained linear quadratic regulation. IEEE Trans. On Automatic Control, 43(8), pp. 1163-1169.