Embedded Position Control System of a Manipulator using a Robust Nonlinear Predictive Control

Arturo Rojas–Moreno
Department of Electronic Engineering
University of Engineering and Technology
Lima 43, Peru
Email: arojas@utec.edu.pe

Richard Valdivia–Mallqui
Faculty of Electronic Engineering
National University of Callao
Callao 02, Peru
Email: richvam@gmail.com

***Abstract*—This paper deals with the implementation of a embedded position control system using a robust nonlinear predictive controller, which is employed to control simultaneously angular positions of the base and arm of an angular manipulator of 2DOF (2 Degrees of Freedom). The design of such control system requires the derivation of the dynamic nonlinear model of the manipulator, as well as the determination of the corresponding predictive control law. Intensive simulation studies permitted to find out the initial values of the tuning parameters of the predictive controller. A CRIO (Compact Reconfigurable Input/Output) device was used to embed the control system. Good performance of the predictive control system was verified via experimentation.**

I. Introduction

Control laws for robotic applications require to be effective for achieving accurate tracking of fast motion in the presence of uncertainties and variable inertia and gravitational load of the manipulator during operation. In this context, nonlinear robust predictive controllers constitute a real alternative due to its robustness in the presence of model uncertainties and disturbances.

The number of research works related to the application of nonlinear predictive control to manipulators is continuously growing. In reference [1] an efficient approaches for nonlinear model predictive control, derived via minimization of adequate cost functions is verified by simulation. A nonlinear receding–horizon controller for rigid link manipulators, determined from a quadratic performance index of the predicted tracking error and the predicted control effort, is treated in [2]. In reference [4], a PUMA 560 manipulator robot was controlled using a predictive computed–torque control, while a tracking controller for a two-link planar manipulator on the horizontal space via nonlinear model predictive control is proposed in paper [5].

In this paper, an embedded robust nonlinear predictive control system, based on the algorithm develops in [7], is implemented. The algorithm, designed to tracking control of a n–link manipulator, is based on a prediction model, which is carried out via a truncated Taylor series expansion. The optimal control is computed directly from the minimization of a receding horizon cost function. The control law is running with a simple speed observer instead of the complex nonlinear state observer proposed in [7]. The implemented embedded controller is applied to an angular manipulator for controlling simultaneously angular positions of its base and arm. Experimental results demonstrates that the established design specifications for base and arm: steady state error, percentage overshoot less than 3%, and settling time less than 0.5 s, are met.

This paper is organized as follows. Section II describes the nonlinear model of the manipulator. Derivation of the robust predictive controller is performed in section II, while in section IV is carried out the simulation of the designed control system. The experimental setup, the embedded control software, and the experimental results are presented in sections V, VI, and VII, respectively. Some relevant conclusions of this work are discussed in section VIII. A non embedded version of this work is treated in [8].

II. Manipulator Modelling

The 2DGOF (2 Degrees of Freedom) angular manipulator depicted in Fig. 1 consists of two DC servomotors driving the base and arm of such manipulator. Each servomotor possesses a reduction mechanism (a gear train) and an encoder to sense the angular position of the servo shaft. The described manipulator is a MIMO (Multiple Input Multiple Output) system due to its two inputs: control voltages u_1 and u_2 applied to the armature of each servo, and two outputs: angular positions q_1 and q_2 of the servomotor axes of. Table I describes the variables and parameters of the manipulator.

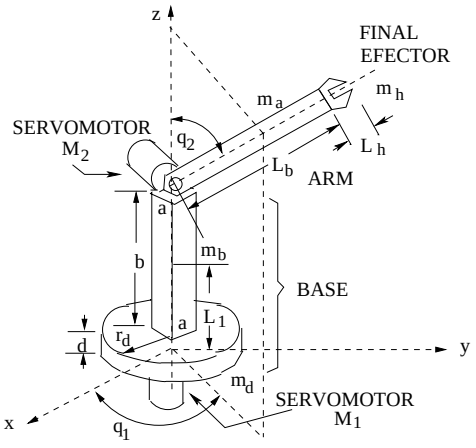

Fig. 1. Angular manipulator scheme.

Neglecting joint frictions (reason for that is explained in equation (14)), the dynamic model of the manipulator (the

TABLE I. VARIABLES AND PARAMETERS OF THE MANIPULATOR. INDEX i TAKES VALUES 1 AND 2 FOR THE BASE AND ARM, RESPECTIVELY.

Symbol	Description	Value	Units
q_i	Angular position		rad
M_i	DC servomotor		
T_i	Generated torque by M_i		N-m
m_d	Disc mass	0.55	kg
m_b	Bar mass	0.9	kg
m_a	Arm mass	0.8	kg
m_h	Final effector mass	0.15	kg
m_2	Arm equivalent mass		kg
d	Disc width	0.01	m
r_d	Disc radioo	0.06	m
b	Bar length	0.25	m
a^2	Area of the bar section	$0.044^{\ 2}$	m^2
L_b	Arm length	0.3	m
L_h	Final effector length	0.05	m
L_1	MC (Mass Center) of the base		m
L_2	Arm equivalent length		m
J_h	Moment de inertia (MI) of the effector		kg-m^2
J_m	MI of M_i	0.0003	kg-m^2
J_{eq}	Equivalent MI		kg-m^2
J_{gi}	MI of the gear train	0.053	kg-m^2
J_i	MI of base and arm		kg-m^2
B_m	Friction constant of M_i	0.0001	N-m-s/rad
B_{eq}	Equivalent friction constant		N-m-s/rad
B_{gi}	Friction constant of a gear train	0.01	N-m-s/rad
n	Gear train ratio of M_i	12.5	
R_a	Armature resistance of M_i	3.5	Ω
L_a	Armature inductance of M_i	0.00015	H
V_{bi}	EMF voltages		V
V_{ai}	Atrmature voltages		V
i_{ai}	Armature currents		A
K_A	Amplifier gain	8.5	
K_m	Servomotor constant	0.0421	N-m/A
K_b	EMF constant	0.0565	V-s/rad
u_i	Control voltages		V
g	Gravitational constant	9.81	m/s^2

Lagrange model) was derived in [6] by mean of the Lagrange equations method, which consists in determining the kinetic (V_i) an potential (U_i) energies for each DOF: angular positions q_1 and and q_2. The resulting Lagrange model of the angular manipulator takes on the form

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})=\mathbf{u};\quad \mathbf{q}=\begin{bmatrix} q_1 \\ q_2 \end{bmatrix};\quad \mathbf{u}=\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{1}$$

$$\mathbf{D}=\begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix};\quad \mathbf{C}=\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix};\quad \mathbf{d}=\begin{bmatrix} 0 \\ d_{21} \end{bmatrix}$$

$$\begin{aligned}
D_{11} &= \frac{R_a}{nK_mK_A}\left(J_1+J_{eq}+\frac{1}{4}m_2L_2 sen^2 q_2\right)\\
D_{22} &= \frac{R_a}{nK_mK_A}\left(J_2+J_{eq}+\frac{1}{4}m_2L_2\right)\\
C_{11} &= \frac{R_a}{nK_mK_A}\left(B_{eq}+\frac{n^2K_mK_b}{R_a}\right)\\
C_{12} &= \frac{R_a}{nK_mK_A}\left(\frac{1}{2}\,m_2L_2\dot{q}_1 senq_2\cos q_2\right)\\
C_{21} &= \frac{R_a}{nK_mK_A}\left(-\frac{1}{4}\,m_2L_2\dot{q}_1 senq_2\cos q_2\right)\\
C_{22} &= \frac{R_a}{nK_mK_A}\left(B_{eq}+\frac{n^2K_mK_b}{R_a}\right)\\
G_{21} &= \frac{R_a}{nK_mK_A}\left(-\frac{1}{2}\,m_2L_2 g senq_2\right)
\end{aligned}$$

In (1), $\mathbf{D}=\mathbf{D}^T>0$ of dimension 2×2 (the notation >0 means that $\mathbf{D}$ is a definite positive matrix) is the inertia matrix of the manipulator, $\mathbf{C}$ of order 2×2 contains Coriolis and centripetal forces, $\mathbf{G}$ is the order two vector of gravitational torques, and $\mathbf{u}$ represents the control control voltage vector of order two.

A. State Equation of the Manipulator

Defining the following state variables: $x_1=q_1$, $x_2=q_2$, $x_3=\dot{q}_1$, and $x_4=\dot{q}_2$, the dynamic model given in (1) can be transformed into the following state equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}=\begin{bmatrix} f_1(\mathbf{x},\mathbf{u}) \\ f_2(\mathbf{x},\mathbf{u}) \\ f_3(\mathbf{x},\mathbf{u}) \\ f_4(\mathbf{x},\mathbf{u}) \end{bmatrix} \tag{2}$$

$$=\begin{bmatrix} x_3 \\ x_4 \\ D_{11}^{-1}(-C_{11}x_3-C_{12}x_4+u_1) \\ D_{22}^{-1}(-C_{21}x_3-C_{22}x_4-G_{21}+u_2) \end{bmatrix}$$

III. PREDICTIVE CONTROLLER DESIGN

As an extension of the 2–dimensional equation (1), a manipulator with rigid arms and n DOF can be described by the following n–dimensional matrix equation

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})=\mathbf{u} \tag{3}$$

This work employs the robust nonlinear predictive control algorithm developed in [7]. Such predictive control algorithm is based on optimization of the following receding-horizon cost function

$$J=\frac{1}{2}\int_{T_1}^{T_2}\mathbf{e}_q(t+\tau)^T\mathbf{e}_q(t+\tau)d\tau \tag{4}$$

$$\mathbf{e}_q(t+\tau)=\mathbf{q}(t+\tau)-\mathbf{q}_r(t+\tau)$$

where $\mathbf{e}_q(t+\tau)$ is the tracking vector at the time $(t+\tau)$, τ is the prediction time, $\mathbf{q}(t+\tau)$ is the τ–step ahead prediction vector of angular positions, and $\mathbf{q}_r(t+\tau)$ represents the vector of future reference trajectories. The control objective of the predictive control system is to track some reference trajectories fulfilling the design specifications previously established.

Applying the truncated Taylor series expansion, the prediction of the output vector $\mathbf{q}$ is found to be

$$\mathbf{q}(t+\tau)=\mathbf{q}(t)+\tau\dot{\mathbf{q}}(t+\tau)+\frac{\tau^2}{2}\ddot{\mathbf{q}}(t) \tag{5}$$

Using the manipulator model given in (3), successive diferentiation of the output $\mathbf{q}(t)$ produces the following $\mathbf{Q}(t)$ vector

$$\mathbf{Q}(t)=\begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \\ \ddot{\mathbf{q}}(t) \end{bmatrix}=\begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \\ -\mathbf{D}(\mathbf{q})^{-1}(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})) \end{bmatrix}$$

$$+\begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \\ \mathbf{D}(\mathbf{q})^{-1}\mathbf{u}(t) \end{bmatrix} \tag{6}$$

Note that $\ddot{\mathbf{q}}(t)$ was obtained from (3). Therefore, the prediction model can be formulated as

$$\mathbf{q}(t+\tau) = \mathbf{T}(t)\mathbf{Q}(t) \qquad \mathbf{T}(t) = [\mathbf{I}_{n\times n} \quad \tau\mathbf{I}_{n\times n} \quad \frac{\tau^2}{2}\mathbf{I}_{n\times n}] \tag{7}$$

where $\mathbf{I}_{n\times n}$ is the identity matrix of dimension $n \times n$. The $\mathbf{q}_r(t+\tau)$ vector can be calculated as before

$$\mathbf{q}_r(t+\tau) = \mathbf{T}(t)\mathbf{Q}_r(t) \qquad \mathbf{Q}_r(t) = [\mathbf{q}_r(t) \quad \dot{\mathbf{q}}_r(t) \quad \ddot{\mathbf{q}}_r(t)] \tag{8}$$

The predictive error is defined as

$$\mathbf{e}_q(t+\tau) = \mathbf{q}(t+\tau) - \mathbf{q}_r(t+\tau) = \mathbf{T}(t)(\mathbf{Q}(t) - \mathbf{Q}_r(t)) \tag{9}$$

Then, cost function (4) takes on the form:

$$J = \frac{1}{2}(\mathbf{Q}(t) - \mathbf{Q}_r(t))^T \mathbf{\Pi}(\mathbf{Q}(t) - \mathbf{Q}_r(t)) \tag{10}$$

where the expression of matrix $\mathbf{\Pi}$ is found to be

$$\begin{aligned}\mathbf{\Pi} &= \int_{T_1}^{T_2} \mathbf{T}(t)^T\mathbf{T}(t)d\tau = \begin{bmatrix} \mathbf{\Pi}_1 & \mathbf{\Pi}_2 \\ \mathbf{\Pi}_2^T & \mathbf{\Pi}_3 \end{bmatrix} \\ &= \begin{bmatrix} T\mathbf{I}_{n\times n} & (T^2/2))\mathbf{I}_{n\times n} & \vdots & (T^3/6))\mathbf{I}_{n\times n} \\ (T^2/2))\mathbf{I}_{n\times n} & (T^3/3))\mathbf{I}_{n\times n} & \vdots & (T^4/8))\mathbf{I}_{n\times n} \\ \cdots & \cdots & \vdots & \cdots \\ (T^3/6))\mathbf{I}_{n\times n} & (T^4/8))\mathbf{I}_{n\times n} & \vdots & (T^5/20))\mathbf{I}_{n\times n} \end{bmatrix}\end{aligned}$$

where $T = T_2 - T_1$. The control law $\mathbf{u}$ can be obtained from $\partial J/\partial \mathbf{u} = 0$ (J is the cost function given in (10))

$$\mathbf{u}(t) = -\mathbf{D}(\mathbf{q})\{[\mathbf{\Pi}_3^{-1}\mathbf{\Pi}_2^T \qquad \mathbf{I}_{n\times n}][\mathbf{M}(t) - \mathbf{Q}_r(t)]\} \tag{11}$$

$$\mathbf{M}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \\ -\mathbf{D}(\mathbf{q})^{-1}(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})) \end{bmatrix}$$

$$[\mathbf{\Pi}_3^{-1}\mathbf{\Pi}_2^T \qquad \mathbf{I}_{n\times n}] = [(\frac{10}{3T^2})\mathbf{I}_{n\times n} \qquad (\frac{5}{2T})\mathbf{I}_{n\times n} \qquad \mathbf{I}_{n\times n}]$$

Therefore, the control law given by (11) is expressed as

$$\begin{aligned}\mathbf{u}(t) = -\mathbf{D}(\mathbf{q})\{&\mathbf{K}_1(\mathbf{q} - \mathbf{q}_r) + \mathbf{K}_2(\dot{\mathbf{q}} - \dot{\mathbf{q}}_r) \\ &-\mathbf{D}(\mathbf{q})^{-1}[\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})] - \ddot{q}_r\}\end{aligned} \tag{12}$$

$$\mathbf{K}_1 = k_1\mathbf{I}_{n\times n}; \quad \mathbf{K}_2 = k_2\mathbf{I}_{n\times n}; \quad k_1 = \frac{10}{3T^2}; \quad k_2 = \frac{5}{2T}$$

A. The Robust Nonlinear Predictive Controller

Various source of uncertainties, like modeling and computation errors, as well as unknown loads, need to be considered for real–time implementation purposes. Such uncertainties can be formulated as increments $\Delta(.)$, which will be added to nominal matrices and vectors of the manipulator model described in (3); that is

$$\begin{aligned}\mathbf{D} &\Rightarrow \mathbf{D} + \mathbf{\Delta D} \\ \mathbf{C} &\Rightarrow \mathbf{C} + \mathbf{\Delta C} \\ \mathbf{G} &\Rightarrow \mathbf{G} + \mathbf{\Delta G}\end{aligned} \tag{13}$$

Then, the dynamic model of the manipulator results

$$(\mathbf{D}(\mathbf{q})+\mathbf{\Delta D})\ddot{\mathbf{q}}+(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{\Delta C})\dot{\mathbf{q}}+(\mathbf{G}(\mathbf{q})+\mathbf{\Delta G})+\mathbf{F}_r = \mathbf{u}+\mathbf{b} \tag{14}$$

Note that $\mathbf{F}_r$ and $\mathbf{b}$ vectors, both of order n, have been added to (1). Vectors $\mathbf{F}_r$ and $\mathbf{b}$ represent non modelled frictions (like the joint frictions of the manipulator under study) and system disturbances, respectively. Operating on equation (14), we obtain

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{u} + \eta(\ddot{\mathbf{q}},\dot{\mathbf{q}},\mathbf{q},\mathbf{b}) \tag{15}$$

where η represents the system uncertainty

$$\eta = -\mathbf{\Delta D}\ddot{\mathbf{q}} + \mathbf{\Delta C}\dot{\mathbf{q}} + \mathbf{\Delta G} + \mathbf{F}_r - \mathbf{b} \tag{16}$$

Observe in (16) that the η parameter includes non modelled quantities, parametric uncertainties, and external disturbances.

To make robust the control law (12), the effects of the uncertainties need to be included in the control loop. In general, the uncertainty parameter η is unknown. Then, its estimate η_{est} is required to compute the robust control law. Taken η_{est} into consideration, the control law (12) results

$$\begin{aligned}\mathbf{u}(t) = -\mathbf{D}(\mathbf{q})\{&\mathbf{K}_1(\mathbf{q} - \mathbf{q}_r) + \mathbf{K}_2(\dot{\mathbf{q}} - \dot{\mathbf{q}}_r) \\ &-\mathbf{D}(\mathbf{q})^{-1}[\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})] - \ddot{\mathbf{q}}_r\} - \eta_{est}\end{aligned} \tag{17}$$

B. Uncertainty Estimation that guaranties stability

Replacing (17) in (15), we can obtain the dynamic equation of the tracking error

$$\begin{aligned}\ddot{\mathbf{e}}_q(t) + \mathbf{K}_2\dot{\mathbf{e}}_q(t) + \mathbf{K}_1\mathbf{e}_q(t) &= \mathbf{D}^{-1}(\mathbf{q})\mathbf{e}_\eta(t) \\ \mathbf{e}_\eta = \eta - \eta_{est} \qquad \mathbf{e}_q &= \mathbf{q} - \mathbf{q}_r\end{aligned} \tag{18}$$

In terms of the tracking error $\mathbf{e}_\eta$, the state space model of the dynamic system given in (17) can be expressed as

$$\dot{\mathbf{e}} = \mathbf{A}_1\mathbf{e} + \mathbf{B}\mathbf{D}^{-1}\mathbf{e}_\eta \tag{19}$$

$$\dot{\mathbf{e}} = [\mathbf{e}_q \quad \dot{\mathbf{e}}_q] \qquad \mathbf{A}_1 = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_{n\times n} \\ -\mathbf{K}_1 & -\mathbf{K}_2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n\times n} \\ \mathbf{I}_{n\times n} \end{bmatrix}$$

Knowing that $\mathbf{K}_1 > 0$ and $\mathbf{K}_2 > 0$, then matrix $\mathbf{A}_1$ is Hurwitz. Hence, for any $\mathbf{Q} = \mathbf{Q}^T > 0$, exists a finite matrix $\mathbf{P} = \mathbf{P}^T > 0$ that satisfies the following Lyapunov equation

$$\mathbf{A}_1\mathbf{P}\mathbf{A}_1 = -\mathbf{Q} \tag{20}$$

The Lyapunov method requires to define a Lyapunov function candidate to determine the system stability

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{P}\mathbf{e} + \mathbf{e}_\eta^T\mathbf{\Gamma}\mathbf{e}_\eta \tag{21}$$

In (21), matrix $\mathbf{\Gamma} > 0$ is symmetric. The total derivative of $\dot{V}$ along the trajectory of the system given by (18) results

$$\dot{V} = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}_\eta^T\{(\mathbf{D}^{-1})^T\mathbf{B}^T\mathbf{P}\mathbf{e} + \mathbf{\Gamma}\dot{\mathbf{e}}_\eta\} \tag{22}$$

Let us define

$$\dot{\mathbf{e}}_\eta = -\mathbf{\Gamma}^{-1}(\mathbf{D}^{-1})^T\mathbf{B}^T\mathbf{P}\mathbf{e} = \dot{\eta} - \dot{\eta}_{est} \tag{23}$$

We do not how η varies in time. Therefore, let us assume $\dot{\eta} = 0$, also, for the case of slowing time–variation of η. For each sampling period, from (23), η_{est} is expressed as

$$\dot{\eta}_{est} = \mathbf{\Gamma}^{-1}(\mathbf{D}^{-1})^T\mathbf{B}^T\mathbf{P}\mathbf{e} \tag{24}$$

Substitution of (23) in (22) produces

$$\dot{V} = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} \tag{25}$$

According to Lyapunov's stability theory, (25) guarantees that $\mathbf{e}(t)$ and $\mathbf{e}_\eta(t)$, and therefore $\eta_{est}(t)$ are bounded. The uncertainty law given in (24) can be written as

$$\eta_{est} = \int \mathbf{\Gamma}^{-1}(\mathbf{D}^{-1})^T\mathbf{B}^T\mathbf{P}\mathbf{e}dt \tag{26}$$

An uncertainty estimator developed in [9] can be designed from the manipulator model (15). That is

$$\begin{aligned}\dot{\eta}_{est} &= \mathbf{L}(\mathbf{D}(\mathbf{q}))^{-1}(\eta - \eta_{est}) \qquad \mathbf{L} = \ell\mathbf{I}_{n\times n} \\ &= -\mathbf{L}(\mathbf{D}(\mathbf{q}))^{-1}\eta_{est} + \mathbf{L}(\ddot{\mathbf{q}} + (\mathbf{D}(\mathbf{q}))^{-1}\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\ &\quad +(\mathbf{D}(\mathbf{q}))^{-1}\mathbf{G}(\mathbf{q}) - (\mathbf{D}(\mathbf{q}))^{-1}\mathbf{u}(t)) \end{aligned} \tag{27}$$

Note in (27) that $\mathbf{L} = \ell\mathbf{I}_{n\times n}$ is a gain diagonal matrix, where ℓ is positive. Considering that $\dot{\eta}(t) = 0$, from (27) can be determined the estimator dynamics; that is

$$\dot{\mathbf{e}}_q + \mathbf{L}(\mathbf{D}(\mathbf{q}))^{-1}\mathbf{e}_q = 0 \tag{28}$$

Using the fact that $\mathbf{L} > 0$ and $\mathbf{D} > 0$, then it is verifiable that the estimator tracking error converge to zero. On replacing the control law (16) into equation (28), we obtain the dynamic of the uncertainties estimation. That is

$$\dot{\eta}_{est}(t) = \mathbf{L}\ddot{\mathbf{e}}_q(t) + \mathbf{K}_2\dot{\mathbf{e}}_q(t) + \mathbf{K}_1\mathbf{e}_q(t)) \tag{29}$$

Integration of (29) produces

$$\eta_{est}(t) = \mathbf{L}\dot{\mathbf{e}}_q(t) + \mathbf{K}_2\mathbf{e}_q(t) + \mathbf{K}_1\int \mathbf{e}_q(t)dt) \tag{30}$$

Comparison of estimators (30) and (26) tell us that the estimator (30) is better due to the presence of the integral operator, which permit to achieve null steady state errors.

C. The Speed Observer

Instead of the nonlinear state observer proposed in [7], this work uses for simplicity a speed observer. Taking into account that only the angular position $\mathbf{q}$ is available, such observer to estimate $\dot{\widehat{\mathbf{q}}}$ has the form

$$\dot{\widehat{\mathbf{q}}} = \dot{\mathbf{q}}_r + \mathbf{L}_d(\mathbf{q} - \widehat{\mathbf{q}}) \tag{31}$$

In (31), $\mathbf{L_d} = \ell_d\mathbf{I}_{n\times n} > 0$ is the observer's gain diagonal matrix, where ℓ_d is a positive constant.

IV. Design Procedure and Simulation

The following procedure can be applied to implement the robust nonlinear predictive controller of III-A:

1) Determine the model of the manipulator (section II)

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{u} \tag{32}$$

2) Set initial conditions of vectors $\mathbf{q}$, $\dot{\mathbf{q}}$, $\widehat{\mathbf{q}}$, and $\dot{\widehat{\mathbf{q}}}$.
3) Set initial values to the tuning parameters ℓ_d (speed observer gain), ℓ (uncertainties estimator gain), T_1, T_2, $T = T_1 - T_2$, $k_1 = 10/(3T^2)$, $k_2 = 5/(2T)$, $\mathbf{K}_1 = k_1\mathbf{I}_{n\times n}$, and $\mathbf{K}_2 = k_2\mathbf{I}_{n\times n}$, where $n = 2$ is the system order.
4) Design of the control loop for a sampling time T_s and for a reference vector $\mathbf{q}_r(t)$. This work employs step wise reference signals for simulation purposes. However, $\mathbf{q}_r(t)$ could be any trajectory.
5) Formulate the speed observer given by (31) to determine the estimate vector $\dot{\widehat{\mathbf{q}}}$

$$\dot{\widehat{\mathbf{q}}} = \dot{\mathbf{q}}_r + \mathbf{L}_d(\mathbf{q} - \widehat{\mathbf{q}}) \tag{33}$$

The $\widehat{\mathbf{q}}$ vector is obtained by integration of $\dot{\widehat{\mathbf{q}}}$.
6) Formulate equations $\mathbf{e}_q = \mathbf{q} - \mathbf{q}_r$, $\dot{\mathbf{e}}_q = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r$, and $\int \mathbf{e}_q(t)dt$, to determine the uncertainties estimator (equation (30))

$$\eta_{est}(t) = \mathbf{L}\dot{\mathbf{e}}_q(t) + \mathbf{K}_2\mathbf{e}_q(t) + \mathbf{K}_1\int \mathbf{e}_q(t)dt) \tag{34}$$

7) Calculate the control law given in (17)

$$\begin{aligned}\mathbf{u}(t) = -\mathbf{D}(\mathbf{q})\{\mathbf{K}_1(\mathbf{q} - \mathbf{q}_r) + \mathbf{K}_2(\dot{\mathbf{q}} - \dot{\mathbf{q}}_r) \\ -\mathbf{D}(\mathbf{q})^{-1}[\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})] - \ddot{\mathbf{q}}_r\} - \eta_{est}\end{aligned} \tag{35}$$

For the simulation phase, the control law (35) is applied to the dynamic model of the manipulator, whereas for real–time implementation, such control law drives the manipulator.

The simulation program cprmrar.m, written in MATLAB code, is listed below. It simulates the predictive control system. Fig. 2 shows the result of the simulation. Observe that angular positions $q_1(t)$ for the base and $q_2(t)$ for the arm tracks the corresponding step type references $q_{1r}(t)$ and $q_{2r}(t)$. Note that the design specifications null steady state, settling time less than 1 s, and percentage overshoot less than 3% are met.

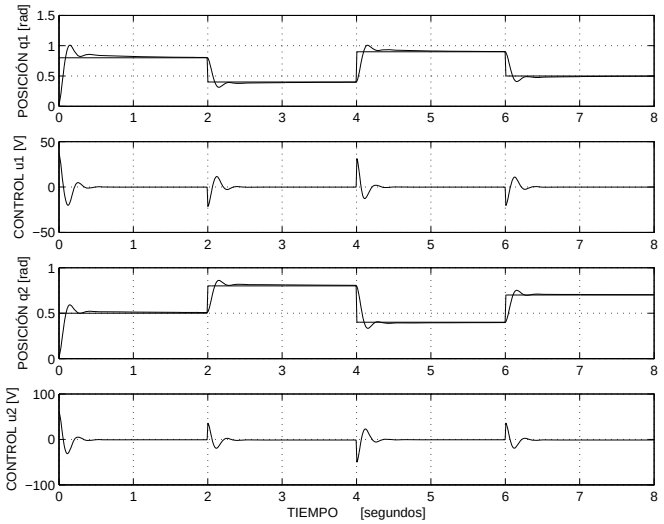

Fig. 2. Controlled angular positions q_1 y q_2 of the manipulator using nonlinear robust predictive control signals u_1 y u_2.

```
% cprmrar.m ROBUST NONLINEAR PREDICTIVE CONTROL
  clear all; close all; clc;
% DATA: MECHANICAL SUBSYSTEM
  d=0.01; rd=0.06; b=0.25; a=0.044; La=0.3;
  Lh=0.05; L2=La+Lh; md=0.55; mb=0.9; ma=0.8;
  mh=0.15; g=9.81; n=12.5; m1=md+mb; m2=ma+mh;
  L1=(mb*b+md*d)/(mb+md); Jm=0.0003; Jg=0.053;
  Jd= md*rd^2/2; Jb=mb*a^2/6; J1=Jd+Jb;
```

```
  J2=m2*L2/3; Jeq=n^2*Jm+Jg; Bm=0.0001; Bg=0.01;
  Beq=n^2*Bm+Bg; no=2; % SYSTEM ORDER
% DATA: ELECTRICAL SUBSYSTEM
  Km=0.0421; Kb=0.0564; Larm=0; Ra=5.3;
% KA=8.5; co=Ra/(n*Km*KA);
% CONSTANT ELEMENTS OF D, C AND G MATRICES
  D22=Ra*(J2+Jeq+m2*L2/4)/(n*Km*KA); D22e=D22;
  C11=Ra*(Beq+n^2*Km*Kb/Ra)/(n*Km*KA); C11e=C11;
  C22=Ra*(Beq+n^2*Km*Kb/Ra)/(n*Km*KA); C22e=C22;
% UNCERTAINTIES
  DeltaD=[.01 0;0 .02]; DeltaG=[.0;-.1];
  DeltaC=[.01 -.02;.001 -.03]; bu=[.1;-.1];
% INITIAL CONDITIONS
  q1=0; q2=0; dq1=0; dq2=0; qe1=0; qe2=0;
  dqe1=0; dqe2=0; ieq=[0;0]; q=[q1;q2];
  dq=[dq1;dq2]; qe=[qe1;qe2]; dqe=[dqe1;dqe2];
% TUNING PARAMETERS
  Ld=35; T1=0; T2=0.1; T=T2-T1; el=0.2;
  L=el*eye(no); k1=10/(3*T^2); k2=5/(2*T);
  K1=k1*eye(no); K2=k2*eye(no);
% CONTROL LOOP
  nn=8000;  Td=10; A=1; Ts=0.01; % SAMPLING
  for k=1:nn
% DESIRED REFERENCES
  if(k>=0 && k<=nn/4)
     qr1=0.8*A; qr2=0.5*A;
  elseif(k>=nn/4&&k<=nn/2)
     qr1=0.4*A; qr2=0.8*A;
  elseif(k>=nn/2&&k<=3*nn/4)
     qr1=0.9*A; qr2=0.4*A;
  elseif(k>=3*nn/4&&k <= nn)
     qr1=0.5*A; qr2=0.7*A; end
  Qr1(k)=qr1; dqr1=0; ddqr1=0;
  Qr2(k)=qr2; dqr2=0; ddqr2=0; qr=[qr1;qr2];
  dqr=[dqr1;dqr2]; ddqr=[ddqr1;ddqr2];
% SPEED OBSERVER
  dqe=dqr+Ld*[q-qe]; qe=qe+Ts*dqe;
  eq=qe-qr; deq=dqe-dqr;
% CONTROL LAW
  D11e = Ra*(J1 + Jeq)/(n*Km*KA) + ...
  (Ra*m2*L2/(4*n*Km*KA))*sin(qe(2))^2;
  C12e = Ra*m2*L2^2/(2*n*Km*KA)*dqe(1)* ...
  sin(qe(2))*cos(qe(2));
  C21e = -(Ra*m2*L2^2/(2*n*Km*KA)/4)* ...
  dqe(1)*sin(qe(2))*cos(qe(2));
  De=[D11e 0;0 D22e]; Ce=[C11e C12e;C21e C22e];
  G21e=-(Ra*m2*L2*g/(2*n*Km*KA))*sin(qe(2));
  Ge=[0;G21e];
  ieq = ieq + Ts*eq;
  etae = L*(deq + K2*eq + K1*ieq);
  u =-De*(K1*eq+K2*deq-inv(De)* ...
  (Ce*dqe+Ge-ddqr)-etae;
  U1(k) = u(1); U2(k) = u(2);
% DYNAMIC MODEL OF THE MANIPULATOR
  D11 = Ra*(J1 + Jeq)/(n*Km*KA) + ...
  (Ra*m2*L2/(4*n*Km*KA))*sin(q(2))^2;
  C12 = Ra*m2*L2^2/(2*n*Km*KA)*dq(1)* ...
  sin(q(2))*cos(q(2));
  C21 = -(Ra*m2*L2^2/(2*n*Km*KA)/4)* ...
  dq(1)*sin(q(2))*cos(q(2));
  D=[D11 0;0 D22]; C=[C11 C12;C21 C22];
  G21=-(Ra*m2*L2*g/(2*n*Km*KA))*sin(q(2));
  G=[0;G21]; dq=dq+Ts*inv(D+DeltaD)*((u+bu)-...
  (C+DeltaC)*dq - (G+DeltaG));
  q =q+Ts*dq; Q1(k)=q(1); Q2(k)=q(2); end
% IT FOLLOWS PLOTS SENTENCES
```

V. Experimental Setup

Fig. 3 shows the experimental setup of the predictive position control system. The base and arm of the angular manipulator (Fig. 1) are driven by two DC servomotors. Each of them possesses a reduction mechanism and a quadrature encoder to sense the angular position of the servomotor shaft. A NI cRIO–9073 (Cmpact Reconfigurable Input/Output) device was used to embed the control algorithm. Module NI 9263 14–Ch $\pm$ 10 V 16 bit was employed to output the two control voltages, while module NI 9401 8–Ch TTL High Speed Digital I/O had the task to process the two angular position signals (in the form of two pulse trains) from the quadrature encoders. Each control voltage is in turn amplified by a PWM (Pulse Width Modulation) Galil Motion Control Amplifier. In turn, each amplifier outputs a control DC voltage to the armature of its corresponding DC servomotor.

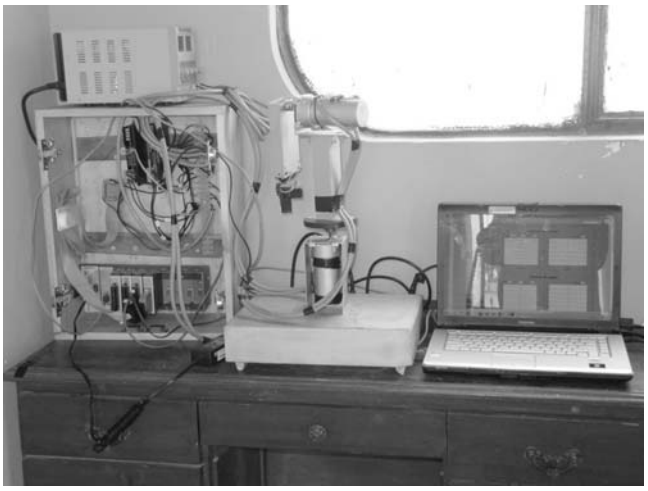

Fig. 3. The experimental setup.

VI. Control Software

The control software, written in LabVIEW code, comprises several parts. A work computer (e.g., a laptop) is used to run the Human Machine Interface part in order to observe running variables and introduce tuning parameters. There are also the part to storage historic data, the communication loop part to connect variables between the work computer and the cRIO device, and, the part to display historic data. The cRIO device executes another communication loop: the control algorithm (Fig. 4), and, all the data required by the implementation. The control software is based on the simulation program cprmrar.m (section IV).

VII. Experimental Results

The performance of the designed robust nonlinear predictive control system was tested via experimentation for a sampling time T_s = 10 ms. Design specifications were set to null steady state error, overshoot percentage less than 3%, and settling time less than 1 s.

Figs. 5 depicts the simultaneous control of angular positions of the base and arm of the manipulador. Tuning parameters were set to L_d = 50, T_1 = 0, T_2 = 0.15 and ℓ

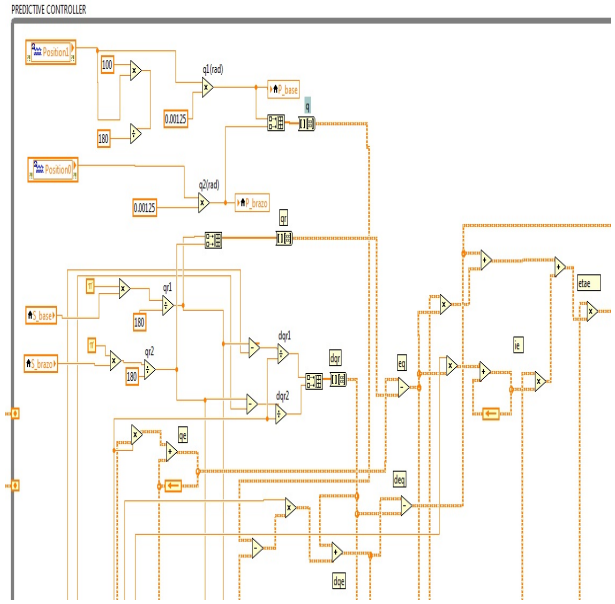

Fig. 4. Portion of the program showing the control algorithm.

=0.5. Such parameters permit to calculate the other tuning parameters: $T = T_1 - T_2$, $k_1 = 10/(3T^2)$, $k_2 = 5/(2T)$, $\mathbf{K}_1 = k_1\mathbf{I}$, and $\mathbf{K}_2 = k_2\mathbf{I}$, where $\mathbf{I}$ is the Identity matrix of order 2. Note that the required design specifications for the angular positions of base and arm: percentage overshoot less than 3%, settling time less than 1 s, and null steady state error, are satisfied.

VIII. Conclusions

In view of the results of section VII, the main goal of this work has been achieved: to control simultaneously the base and arm positions of an angular manipulator using a robust nonlinear predictive control systems. Fig. 5 demonstrates that the established design specifications were met: null steady state error, percentage overshoot less than 2%, and settling time less than 0.5 seconds.

The determination of the dynamic model of the manipulator (section II) was required to simulate the behavior of the designed predictive control systems (Fig. 2) and to determine by trial–and–error the initial values of the tuning parameters.

The proposed design procedure to implement an embedded nonlinear predictive control system, can be applied to any nonlinear system whose dynamic model can be represented as in equation (3).

Acknowledgment

The authors would like to thank to the University of Engineering and Technology (www.utec.edu.pe) for supporting the presentation of this work to the scientific community.

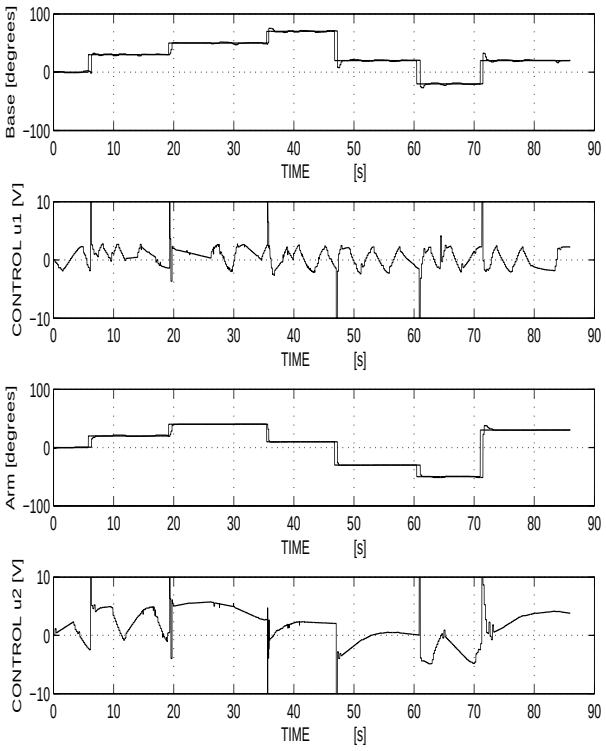

Fig. 5. Experimental result: simultaneous control of angular positions of the base and arm of the manipulator employing a robust nonlinear predictive controller.

References

[1] P. H. poignet, *Nonlinear model predictive control of a robot manipulator*, Advanced Motion Control, 2000. Proceedings. 6th International Workshop on.

[2] R. Hedjar and P. Boucher, *Nonlinear Receding-Horizon Control of Rigid Link Robot Manipulators*, International Journal of Advanced Robotic Systems, Volume 2, Number 1, pp. 015–024, 2005.

[3] V. M. Becerra, S. Cook, and J. Deng, *Predictive computed-torque control of a PUMA 560 manipulator robot*, In: P. Piztek (ed.) Proceedings of the 16th IFAC World Congress, Elsevier, 2006.

[4] K. Bdirina, D. Djoudi, M. Lagoun, and N. Hamani, *Nonlinear Predictive Control of Rigid Link Manipulator*, International Conference on Trends in Industrial and Mechanical Engineering (ICTIME'2012), March 24–25, Dubai.

[5] T. Henmi, T. Ohta, M. Deng, and A. Inoue, *Tracking Control of a two–link planar manipulator using nonlinear model predictive control*, International Journal of Innovative, Computing, Information and Control, Volume 6, Number 7, July 2010.

[6] A. Rojas Moreno, *Control No Lineal Multivariable. Aplicaciones en Tiempo Real*, Editorial Universitaria UNI (EDUNI), ISBN 978–612–4072–18–5, 2012. Ingeniería, FIEE, Lima, 2012.

[7] A. Merabet and J. Gu, *Robust nonlinear predictive control based on state estimation for robot manipulator*, Int. J. of Appl. Math and Mech., pp 48–49, V 5, N 1, 2009.

[8] H. Hurtado, *Implementación de un Sistema de Control Predictivo para un Manipulador de 2GDL*, Tesis de titulación (por sustentarse), Universidad Católica Santa María, Facultad de Ingeniería Electrónica, 2013.

[9] W. Feng, J. O'Reilly, and DJ Balance, *MIMO nonlinear PID predictive control*, IEE Proceedings Control Theory Applications, pp 201–208, V 149, N 3, 2002.