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.

![Angular manipulator scheme.](image)

Neglecting joint frictions (reason for that is explained in equation (14)), the dynamic model of the manipulator (the
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_1 )</td>
<td>Angular position</td>
<td></td>
<td>rad</td>
</tr>
<tr>
<td>( M_1 )</td>
<td>DC servomotor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \tau_1 )</td>
<td>Generated torque by ( M_1 )</td>
<td></td>
<td>N-m</td>
</tr>
<tr>
<td>( m_d )</td>
<td>Disc mass</td>
<td>0.55</td>
<td>kg</td>
</tr>
<tr>
<td>( m_b )</td>
<td>Bar mass</td>
<td>0.9</td>
<td>kg</td>
</tr>
<tr>
<td>( m_a )</td>
<td>Arm mass</td>
<td>0.8</td>
<td>kg</td>
</tr>
<tr>
<td>( m_{eff} )</td>
<td>Final effector mass</td>
<td>0.15</td>
<td>kg</td>
</tr>
<tr>
<td>( d_2 )</td>
<td>Arm equivalent mass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( d )</td>
<td>Disc width</td>
<td>0.01</td>
<td>m</td>
</tr>
<tr>
<td>( t_d )</td>
<td>Disc radius</td>
<td>0.06</td>
<td>m</td>
</tr>
<tr>
<td>( d )</td>
<td>Bar length</td>
<td>0.25</td>
<td>m</td>
</tr>
<tr>
<td>( a^2 )</td>
<td>Area of the bar section</td>
<td>0.044 m^2</td>
<td></td>
</tr>
<tr>
<td>( L_{a1} )</td>
<td>Arm length</td>
<td>0.3</td>
<td>m</td>
</tr>
<tr>
<td>( L_{a2} )</td>
<td>Final effector length</td>
<td>0.05</td>
<td>m</td>
</tr>
<tr>
<td>( L_{1} )</td>
<td>MC (Mass Center) of the base</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>( L_{2} )</td>
<td>Arm equivalent length</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>( J_m )</td>
<td>Moment de inertia (MI) of the effector</td>
<td>0.0003</td>
<td>kg-m^2</td>
</tr>
<tr>
<td>( J_{eq} )</td>
<td>Equivalent MI</td>
<td></td>
<td>kg-m^2</td>
</tr>
<tr>
<td>( J_{g1} )</td>
<td>MI of the gear train</td>
<td>0.053</td>
<td>kg-m^2</td>
</tr>
<tr>
<td>( J_{g2} )</td>
<td>MI of arm and base</td>
<td></td>
<td>kg-m^2</td>
</tr>
<tr>
<td>( d_{b1} )</td>
<td>Friction constant of ( M_1 )</td>
<td>0.0001</td>
<td>N-m-s/rad</td>
</tr>
<tr>
<td>( d_{b2} )</td>
<td>Equivalent friction constant</td>
<td></td>
<td>N-m-s/rad</td>
</tr>
<tr>
<td>( d_{b3} )</td>
<td>Friction constant of a gear train</td>
<td>0.01</td>
<td>N-m-s/rad</td>
</tr>
<tr>
<td>( n )</td>
<td>Gear train ratio of ( M_1 )</td>
<td>12.5</td>
<td></td>
</tr>
<tr>
<td>( R_a )</td>
<td>Armature resistance of ( M_1 )</td>
<td>3.5</td>
<td>\Omega</td>
</tr>
<tr>
<td>( L_{a1} )</td>
<td>Armature inductance of ( M_1 )</td>
<td>0.00015</td>
<td>H</td>
</tr>
<tr>
<td>( V_{B0} )</td>
<td>EMF voltages</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>( V_{B1} )</td>
<td>Armature voltages</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>( v_{a1} )</td>
<td>Armature currents</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>( R_s )</td>
<td>Amplifier gain</td>
<td>8.3</td>
<td></td>
</tr>
<tr>
<td>( K_a )</td>
<td>Servomotor constant</td>
<td>0.0421</td>
<td>N-m/A</td>
</tr>
<tr>
<td>( K_v )</td>
<td>EMF constant</td>
<td>0.0565</td>
<td>V-s/rad</td>
</tr>
<tr>
<td>( u_{11} )</td>
<td>Control voltages</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>( q )</td>
<td>Gravitational constant</td>
<td>9.81</td>
<td>m/s^2</td>
</tr>
</tbody>
</table>

The variables and parameters of the manipulator are given in Table I. The table lists some of the key parameters used in the model, such as the moment of inertia, mass, and friction constants.

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

\[
\mathbf{D}(\dot{q})\ddot{q} + \mathbf{C}(\dot{q}, \dot{q})\dot{q} + \mathbf{G}(q) = \mathbf{u} \quad \text{or} \quad \mathbf{D} \ddot{q} + \mathbf{C} \dot{q} + \mathbf{G} = \mathbf{u}
\]

where \( \mathbf{D} \) is a positive definite matrix, \( \mathbf{C} \) is a positive semi-definite matrix, \( \mathbf{G} \) is a constant matrix, and \( \mathbf{u} \) is the control input vector.

In (1), \( \mathbf{D} = \mathbf{D}^T > 0 \) of dimension 2 \times 2 (the notation \( \mathbf{D}^T \) means that \( \mathbf{D} \) is a definite positive matrix) is the inertia matrix of the manipulator. \( \mathbf{C} \) of order 2 \times 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: \( x_1 = q_1, x_2 = q_2, x_3 = q_1, \) and \( x_4 = q_2, \) the dynamic model given in (1) can be transformed into the following state equation

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4
\end{bmatrix} =
\begin{bmatrix}
f_1(x, u) \\
f_2(x, u) \\
f_3(x, u) \\
f_4(x, u)
\end{bmatrix}
\]

### 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

\[
\mathbf{D}(\dot{q})\ddot{q} + \mathbf{C}(\dot{q}, \dot{q})\dot{q} + \mathbf{G}(q) = \mathbf{u}
\]

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

\[
J = \frac{1}{2} \int_{T_0}^{T_2} e_\tau(t + \tau)^T e_\tau(t + \tau) d\tau
\]

where \( e_\tau(t + \tau) \) is the tracking vector at the time \( t + \tau \), \( \tau \) is the prediction time, \( q(t + \tau) \) is the \( \tau \)-step ahead prediction vector of angular positions, and \( q(t + \tau) \) 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

\[
\mathbf{q}(t + \tau) = \mathbf{q}(t) + \tau \dot{\mathbf{q}}(t + \tau) + \frac{\tau^2}{2} \ddot{\mathbf{q}}(t)
\]

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

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

Using the manipulator model given in (3), successive differentiation of the output vector \( \mathbf{u}(t) \) produces the following \( \mathbf{d}(t) \) vector

\[
\mathbf{d}(t) =
\begin{bmatrix}
0_{n \times 1} \\
0_{n \times 1} \\
\mathbf{D}(\dot{q})^{-1}\mathbf{u}(t)
\end{bmatrix}
\]
Note that $\ddot{q}(t)$ was obtained from (3). Therefore, the prediction model can be formulated as

$$q(t+\tau) = T(t)Q(t)$$

where $I_{n \times n}$ is the identity matrix of dimension $n \times n$. The $q_r(t+\tau)$ vector can be calculated as before

$$q_r(t+\tau) = T(t)Q_r(t)$$

The predictive error is defined as

$$e(t) = q(t) - q_r(t)$$

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

$$J = \int_{T_1}^{T_2} \tau \Pi Q(t) - Q_r(t) dt$$

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

$$\Pi = \int_{T_1}^{T_2} T(t)^T T(t) d\tau = \begin{bmatrix} \Pi_1 & \Pi_2 \\ \Pi_2^T & \Pi_3 \end{bmatrix}$$

where $T = T_2 - T_1$. The control law $u(t)$ can be obtained from

$$u(t) = -D(q)\{\Pi_3^{-1}\Pi_2^T \ I_{n \times n}\}[M(t) - Q_r(t)]$$

Then, the dynamic model of the manipulator results

$$D(q)\ddot{q} + C(q, \dot{q}) + \Delta C \dot{q} + G(q) + \Delta G + F_r = u + b$$

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

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

where $\eta$ represents the system uncertainty

$$\eta = -\Delta D \ddot{q} + \Delta C \dot{q} + \Delta G + F_r - b$$

Observe in (16) that the $\eta$ 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 $\eta$ is unknown. Then, its estimate $\eta_{est}$ is required to compute the robust control law. Taken $\eta_{est}$ into consideration, the control law (12) results

$$u(t) = -D(q)\{K_1(q - q_r) + K_2(\dot{q} - \dot{q}_r)$$

$$-D(q)^{-1}C(q, \dot{q})\dot{q} + G(q)\} - \dot{q}_r$$

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 $\Delta(\cdot)$, which will be added to nominal matrices and vectors of the manipulator model described in (3); that is

$$D \Rightarrow D + \Delta D$$

$$C \Rightarrow C + \Delta C$$

$$G \Rightarrow G + \Delta G$$

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

$$\dot{V} = -\frac{1}{2} e^T \Gamma \dot{e} + e_\eta^T \{D^{-1}\}^T B^T \dot{P}e + \Gamma \dot{\eta}$$

Let us define

$$\dot{\eta} = -\Gamma^{-1}(D^{-1})^T B^T \dot{P}e$$

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

$$\eta_{est} = \Gamma^{-1}(D^{-1})^T B^T \dot{P}e$$
Substitution of (23) in (22) produces

\[ \dot{V} = -\frac{1}{2} e^T Q e \]  

(25)

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

\[ \eta_{est} = \int \Gamma^{-1}(D^{-1})^T B^T \Phi dt \]  

(26)

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

\[ \dot{\eta}_{est} = L(D(q))^{-1}(\eta - \eta_{est}) = -L(D(q))^{-1} \dot{\eta} + L(\dot{q} + (D(q))^{-1} C(q, \dot{q}) \dot{q} + (D(q))^{-1} G(q) - (D(q))^{-1} u(t)) \]  

(27)

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

\[ \dot{e}_q + L(D(q))^{-1} e_q = 0 \]  

(28)

Using the fact that \( L > 0 \) and \( D > 0 \), 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

\[ \dot{\eta}_{est}(t) = L\dot{e}_q(t) + K_2 \dot{e}_q(t) + K_1 e_q(t) \]  

(29)

Integration of (29) produces

\[ \eta_{est}(t) = L\dot{e}_q(t) + K_2 e_q(t) + K_1 \int e_q(t) dt \]  

(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 \( q \) is available, such observer to estimate \( \dot{q} \) has the form

\[ \dot{\hat{q}} = \dot{q}_r + L_d (q - \hat{q}) \]  

(31)

In (31), \( L_d = \ell_d I_{n \times n} > 0 \) is the observer’s gain diagonal matrix, where \( \ell_d \) 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)

\[ D(q) \dot{\hat{q}} + C(q, \dot{q}) \dot{q} + G(q) = u \]  

(32)

2) Set initial conditions of vectors \( q, \dot{q}, \hat{q}, \) and \( \dot{\hat{q}}. \)

3) Set initial values to the tuning parameters \( \ell_d \) (speed observer gain), \( \ell \) (uncertainties estimator gain), \( T_1, T_2, T = T_1 - T_2, k_1 = 10/(3T^2), k_2 = 5/(2T), K_1 = k_1 I_{n \times n}, \) and \( K_2 = k_2 I_{n \times n}, \) where \( n = 2 \) is the system order.

4) Design of the control loop for a sampling time \( T_s \) and for a reference vector \( \dot{q}_r(t). \) This work employs step wise reference signals for simulation purposes. However, \( q_s(t) \) could be any trajectory.

5) Formulate the speed observer given by (31) to determine the estimate vector \( \dot{\hat{q}} \)

\[ \dot{\hat{q}} = \dot{q}_r + L_d (q - \hat{q}) \]  

(33)

The \( \dot{\hat{q}} \) vector is obtained by integration of \( \dot{\hat{q}}. \)

6) Formulate equations \( e_q = q - q_s, \) \( \overline{e_q} = \hat{q} - \dot{q}_r, \) and \( \int e_q(t) dt, \) to determine the uncertainties estimator (equation (30))

\[ \eta_{est}(t) = L\dot{e}_q(t) + K_2 e_q(t) + K_1 \int e_q(t) dt \]  

(34)

7) Calculate the control law given in (17)

\[ u(t) = -D(q) [K_1 (q - q_s) + K_2 (\dot{q} - \dot{q}_r)] - D(q)^{-1} [C(q, \dot{q}) \dot{q} + G(q)] - \dot{\hat{q}} - \eta_{est} \]  

(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 \( q_1(t) \) and \( q_2(t) \). 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 q1 and q2 of the manipulator using nonlinear robust predictive control signals u1 and u2.](image)
\[ J_2 = m_2 L_2 / 3; \quad J_{eq} = n^2 * J_m + J_g; \quad B_m = 0.0001; \quad B_g = 0.01; \]
\[ B_{eq} = n^2 * B_m + B_g; \quad n = 2; \quad \% \text{SYSTEM ORDER} \]

\% DATA: ELECTRICAL SUBSYSTEM

\[ K_m = 0.0421; \quad K_b = 0.0564; \quad L_{arm} = 0; \quad R_a = 5.3; \]
\[ K_a = 8.5; \quad c_o = R_a / (n * K_m * K_a); \]

\% CONSTANT ELEMENTS OF D, C AND G MATRICES

\[ D_{22} = R_a * (J_2 + J_{eq} + m_2 L_2 / 4) / (n * K_m * K_a); \]
\[ D_{22e} = D_{22}; \]
\[ C_{11} = R_a * (B_{eq} + n^2 * K_m * K_b / R_a) / (n * K_m * K_a); \]
\[ C_{11e} = C_{11}; \]
\[ C_{22} = R_a * (B_{eq} + n^2 * K_m * K_b / R_a) / (n * K_m * K_a); \]
\[ C_{22e} = C_{22}; \]

\% UNCERTAINTIES

\[ \Delta D = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.02 \end{bmatrix}; \quad \Delta G = \begin{bmatrix} 0 \\ -0.1 \end{bmatrix}; \]
\[ \Delta C = \begin{bmatrix} 0.01 & -0.02 \\ 0.001 & -0.03 \end{bmatrix}; \quad \Delta u = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}; \]

\% INITIAL CONDITIONS

\[ q_1 = 0; \quad q_2 = 0; \quad dq_1 = 0; \quad dq_2 = 0; \quad q_{e1} = 0; \quad q_{e2} = 0; \]
\[ dq_{e1} = 0; \quad dq_{e2} = 0; \quad i_{eq} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \]
\[ q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}; \quad dq = \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix}; \quad q_{e} = \begin{bmatrix} q_{e1} \\ q_{e2} \end{bmatrix}; \quad dq_{e} = \begin{bmatrix} dq_{e1} \\ dq_{e2} \end{bmatrix}; \]

\% TUNING PARAMETERS

\[ L_d = 35; \quad T_1 = 0; \quad T_2 = 0.1; \quad T = T_2 - T_1; \]
\[ \ell = 0.2; \quad L = \ell * \text{eye}(n); \]
\[ K_1 = 10 / (3 * T^2); \quad K_2 = 5 / (2 * T); \]
\[ K_1 = \text{eye}(n); \quad K_2 = \text{eye}(n); \]

\% CONTROL LOOP

\[ \text{nn} = 8000; \quad T_d = 10; \quad A = 1; \quad T_s = 0.01; \quad \% \text{SAMPLING} \]

\% DESIRED REFERENCES

\[ \text{if}(k == 0 \&\& k <= \text{nn}/4) \]
\[ \text{if}(k >= \text{nn}/4 \&\& k <= \text{nn}/2) \]
\[ \text{if}(k >= \text{nn}/2 \&\& k <= 3\text{nn}/4) \]
\[ \text{if}(k >= 3\text{nn}/4 \&\& k <= \text{nn}) \]

\[ q_{r1} = 0.8 * A; \quad q_{r2} = 0.5 * A; \]
\[ q_{r1} = 0.4 * A; \quad q_{r2} = 0.8 * A; \]
\[ q_{r1} = 0.9 * A; \quad q_{r2} = 0.4 * A; \]
\[ q_{r1} = 0.5 * A; \quad q_{r2} = 0.7 * A; \]
\[ \text{end} \]
\[ q_{r1}(k) = q_{r1}; \quad dq_{r1} = 0; \quad ddq_{r1} = 0; \]
\[ q_{r2}(k) = q_{r2}; \quad dq_{r2} = 0; \quad ddq_{r2} = 0; \]
\[ q_{r} = \begin{bmatrix} q_{r1} \\ q_{r2} \end{bmatrix}; \quad dq_{r} = \begin{bmatrix} dq_{r1} \\ dq_{r2} \end{bmatrix}; \quad dq_{r} = \begin{bmatrix} ddq_{r1} \\ ddq_{r2} \end{bmatrix} \]

\% SPEED OBSERVER

\[ dq_{e} = dq_{r} + L_d * (q - q_{e}); \]
\[ q_{e} = q_{e} + T_s * dq_{e}; \]
\[ eq_{r} = eq_{r} - dq_{e} \]

\% CONTROL LAW

\[ D_{11} = R_a * (J_1 + J_{eq}) / (n * K_m * K_a) + \ldots \]
\[ C_{12} = R_a * m_2 L_2^2 / (2 * n * K_m * K_a) * dq_{e}(1) \]
\[ C_{21} = -(R_a * m_2 L_2^2 / (2 * n * K_m * K_a) + dq_{e}(1)) \]
\[ D_{e} = [D_{11} 0; 0 D_{22}]; \quad C_{e} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}; \]
\[ G_{21} = -(R_a * m_2 L_2 g / (2 * n * K_m * K_a)) * \text{sin}(q_2); \]
\[ G = \begin{bmatrix} 0 \\ G_{21} \end{bmatrix}; \quad dq_{e} = dq_{e} + T_s * (C_{e} * dq_{e} + G_{21} - \ell \text{eq} - (G + \Delta G)); \]
\[ q = q + T_s * dq_{e}; \quad q_{e1} = q_{e1}; \quad q_{e2} = q_{e2}; \]
\[ \% \text{IT FOLLOWS PLOTS SENTENCES} \]

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 (Compact Reconfigurable Input/Output) device was used to embed the control algorithm. 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.

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 \( T_s = 10 \text{ ms} \). 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 manipulator. Tuning parameters were set to \( L_d = 50, T_1 = 0, T_2 = 0.15 \) and \( \ell \).
Such parameters permit to calculate the other tuning parameters: $T = T_1 - T_2$, $k_3 = 10/(3T^2)$, $k_2 = 5/(2T)$, $K_1 = k_1 I$, and $K_2 = k_2 I$, where $I$ 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 system. 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.

REFERENCES


