Abstract: This paper presents control approaches for a flexible manipulator system. The non-minimum phase problem is treated and solved through two different approaches. A motion planning technique is used in a first approach. It searches for proper output trajectories with polynomial form, in such a manner that the effects of the unstable zeros are completely canceled. The second approach is a recent technique, called path-following. The main objective of path-following with internal model control is to steer a physical object to converge to a geometric path. The secondary objective is to ensure that the object’s motion along the path satisfies a given dynamic specification.

Keywords: Flexible robot manipulators, motion planing, path-following, internal model control, normal form representation, vibration control.

1. INTRODUCTION

There has been a long standing interest in the modeling and control of flexible manipulators robots, due to their high speed operation better accuracy and many other improved characteristics. Unfortunately, the dynamic system of this kind of robots is a highly non-linear system due to the links flexibility. Flexibility also yields a robot model with infinite degrees of freedom due to the flexible links vibration modes.

The control of the motion of flexible manipulators is one of the areas under great investigation in robotics. There are three main objectives in the control of the flexible robot arm:

- O1 - Point to point motion of the end-effector.
- O2 - Trajectory tracking in the joint space (tracking of a desired angular trajectory).
- O3 - Trajectory tracking in the operational space (tracking of a desired end-effector trajectory).

In rigid manipulator robots, aside having non-linearities, the control task is less complicated, since one achieves tip position by controlling the joint angles. This is known as dead-reckoning, as this kind of control approach does not contemplate a feedback position of the end-effector. For flexible manipulators there is the need to use a controller designed to damp out the oscillations at the tip that appear during motion. Dačić [2005] presented three distinct approaches for tracking in the presence of unstable zeros dynamics. He referred to them as: the Internal Model approach, the Flatness approach and the Inversion approach. In this paper we will mainly focus on the Internal Model Approach and the Inversion approach.

This paper is organized as follows: Section 2 recalls the dynamic equations of a one flexible link. In section 3 the trajectory planning method is presented. Section 4 presents the path-following controller. The simulation results are shown in section 5, followed by the experimental results in section 6. Finally, the conclusions are presented in section 7.

2. DYNAMIC EQUATIONS

In Martins [2007] two experimental robotic setups are presented. One for planar experiments, and another for general three dimensional experiments. The work presented here refers to the IST planar flexible manipulator, namely its flexible link, fig. 1. This robot has been studied in [Martins et al., 2002], [Martins et al., 2003] and [Martins et al., 2005] where it is widely described. It will be considered that the flexible link is clamped to a rigid hub with a moment of inertia ($I_H$), radius ($r$) and an input torque ($\tau$). Martins et al. [2002], Martins et al. [2003] and Martins et al. [2005] describes how to obtain the ordinary differential equations of the current system as

$$M\ddot{q} + Kq = T$$

where $M$ is the system inertia matrix, $K$ is the system stiffness matrix, $T$ is the vector of external forces and $q$ is the vector of generalized coordinates:
\[ q = [\theta \ \eta_1 \ \eta_2 \ \ldots \ \eta_k]^T \]
\[ T = [r \ 0 \ 0 \ \ldots \ 0]^T \]  
(2)

Here, \( \eta_i \) is the modal amplitude of the \( i \)th mode considering the assumed modes discretization procedure, representing \( k \) the total number of assumed modes. In this paper we will consider only the first two clamped-free modes.

Considering linear displacements, the inertial displacement at distance \( x \) from the frame origin in the \( OX \) direction is given by

\[ y(x, t) = x \theta(t) + \nu(x, t) \]  
(3)

Observing (3) can be concluded that the total displacement is function of the rigid body motion \( \theta(t) \) and elastic deflection (displacement) \( \nu(x, t) \). The elastic deflection is given by

\[ \nu(x, t) = \sum_{i=1}^{\eta} \phi_i(x) \eta_i(t) \]  
(4)

where \( \phi_i(x) \) and \( \eta_i(t) \) represent the modal functions and modal amplitudes of the \( i \)th Hermite-cubic modes respectively. The Hermite-cubic mode considered are:

\[ \phi_1(x) = 3 \left( \frac{x}{l} \right)^2 - 2 \left( \frac{x}{l} \right)^3 \]
\[ \phi_2(x) = \frac{x}{l} - \frac{x^3}{l^2} \]
\[ \phi(x) = [\phi_1(x) \ \phi_2(x)] \]

where the modes shape are presented in fig. 2.

![First mode and Second mode](image)

Fig. 2. The first two mode shapes

2.1 Model Simulation

For model simulation, a Matlab toolbox called MECANISMO, has been used. This toolbox was developed in Martins [2007], it works seamlessly with other Matlab tools, and has been coded with special attention to its usage in a real time framework. The modules of MECA NISMO, along with other tools available in Simulink, especially the numerical integrators, allow for simulation in free, constrained and impact motion of flexible manipulators. A quadratic model has been built with MECA NISMO for model simulation.

3. MOTION PLANNING - STABLE INVERSION METHOD

Consider \( u \) and \( y \) related by a given input-output equation.

\[ P(\frac{d}{dt})u(t) = Q(\frac{d}{dt})y(t) \]  
(6)

In (6), \( P \) and \( Q \) are polynomials in a differential operator \( d/dt \), with degrees \( m \) and \( n \) respectively (\( m < n \)). In a first analysis, due to the linear nature of (6), its solution is composed by two terms: the transient and the steady-state.

Since the system has a non-minimum phase characteristic, the transient of the system’s inverse contains divergent terms, resulting from the unstable zeros. Thus, one solution to the problem is to plan the output trajectory in a way that the undesirable response of the system input is cancelled. Considering a polynomial in time for the output trajectory,

\[ y_d = \sum_{i=1}^{p} a_i t^{i-1} \]  
(7)

where the degree of the polynomial form \( p \) depends on the number of output initial and final constraints, as well as the number of unstable zeros associated with (6), the input in equation (6) may be written as

\[ u(t) = u_1(t) + u_p(t) \]  
(8)

where \( u_1 \) is the transient solution and \( u_p \) is the particular solution of the inverse system. The transient solution can be represented by

\[ u_1(t) = \sum_{i=1}^{m} A_i(a_i, t_0, u_0, u_0(1), \ldots, u_0(n-1)) e^{r_i t} \]  
(9)

where the \( r_i \) are all the roots of the characteristic equation of the inverse problem. The \( A_i \) are linear functions of \( a_i \) coefficients and all initial conditions. It can be shown that their general expression is

\[ A_i = u_0 + \sum_{j=1}^{p} \frac{a_j}{2 \pi r_0 j} \]  
(10)

Furthermore, the particular solution can be represented by

\[ u_p(t) = \sum_{i=1}^{p} B_i(a_i)t^{i-1} \]  
(11)

To cancel the effect of the unstable zeros on the transient solution (9) (all the zeros in the right half plane or pure imaginary zeros), the \( A_i \) associated to the unstable zeros must be equal to zero

\[ A_i(a_i, t_0, u_0, u_0(1), \ldots, u_0(n-1)) = 0 \]  
(12)

With this constraint in the linear system, we can obtain the output coefficients \( (a_i) \). Adding the final and initial constraints, this leads to the linear system:

\[ \begin{cases} A_i(a_i, t_0, u_0, u_0(1), \ldots, u_0(n-1)) = 0 \\ u_0'(t_0) = \text{initial conditions} \\ u_0''(t_0) = \text{final conditions} \end{cases} \]  
(13)

where \( i \) is the highest order for the specified output derivatives. From equation (13) we have the coefficients \( a_i \) and the necessary output to cancel the unstable zeros. So, the result of the remaining \( A_i \) is known. To complete the desired input \( u(t) \), it is only necessary to obtain the particular input solution (11). The \( B_i \) elements are obtained as linear functions of the output coefficients \( a_i \), through substitution of equation (9) into the differential equation (6).
At this point, all necessary elements to obtain the desired input in an open loop form are:

\[ u_{ol} = \sum_{i=1}^{p} B_i(a_i) t^{i-1} + \sum_{i=1}^{m} A_{ist}(a_i, t_0, u_0, u_0^{(1)}, \ldots, u_0^{(n-1)}) e^{r_i t_i} \]

where \( r_{ist} \) and \( A_{ist} \) are the stable zeros and the corresponding \( A_i \) terms respectively. In a final approach to the problem, it is recommended to close the loop around the joint angle in order to gain some robustness. The final control law in a closed-loop form is

\[ u_{cl}(t) = u_{ol}(t) + k \left( e_{ij}, e_{ij}^{(1)}, \ldots, e_{ij}^{v-1} \right)^T \]

where the error is \( e_{ij}(t) = \theta_d(t) - \theta(t) \)

3.1 Example of implementation

The dynamic model of the one link flexible arm with two bodies will be considered:

- The first one is a rigid hub with a \( R = 0.075m \) radius.
- The second body is a flexible link with length of \( L = 0.5m \).

The joint torque to tip inertial displacement transfer function of the system is

\[
\frac{y}{\tau} = \frac{92.5 s^4 + 8.545 \times 10^{-12} s^3 - 4.924 \times 10^6 s^2}{s^2(s^4 + 1.066 \times 10^{-14} s^3 + 56780 s^2 + 3.015 \times 10^{-7} s + 1.163 \times 10^{10} + 5.795 \times 10^{-10} s + 1.307 \times 10^8)}
\]

For simplification we will call the coefficients of the numerator as a vector \( N \), and the coefficients of the denominator as a vector \( D \).

The transfer function zeros are:
\[ Zo_{1,2} = \pm 225.2893 \quad Zo_{3,4} = \pm 49.7712 \]
where two are unstable.

The pole zero map is shown in fig. 3.

![Pole Zero Map](image)

Fig. 3. Location of the zeros and poles for transfer function (16)

The differential equation representing equation (16) is

\[
N(1)\tau^{(4)}(t) + N(2)\tau^{(3)}(t) + N(3)\tau^{(2)}(t) + N(4)\tau(t) + N(5) = D(1)y^{(6)}(t) + D(2)y^{(5)}(t) + D(3)y^{(3)}(t) + D(4)y^{(2)}(t) + D(5)y(t) + D(6)\theta(t)
\]

associated to the initial conditions:

\[
\left\{ \begin{array}{l}
\tau(0) = \tau^{(1)}(0) = \tau^{(2)}(0) = \tau^{(3)}(0) = 0 \\
y(0) = y^{(1)}(0) = y^{(2)}(0) = y^{(3)}(0) = 0 \\
y^{(4)}(0) = y^{(5)}(0) = 0 \\
\end{array} \right.
\]

As shown previously, the obtained input-output equation can be associated to any initial conditions. For this particular case they have been assumed to be equal to zero.

The output that will cancel the unstable zero is now found. From \( (7) \), with \( p = 12 \) (2 unstable zeros plus 4 initial conditions on \( \tau \) plus 6 initial conditions on \( y \)) we have

\[
y_{d} = \sum_{i=1}^{p} a_i t^{i-1}
\]

\( y_{d} \) must satisfy 12 constraints:

- The indices \( A_i \) corresponding to the unstable zeros
- Initial and final conditions.

\[
\begin{align*}
A_1 &= \tau_0 + \sum_{j=1}^{p} \frac{a_j}{Z_{o1,j}} = 0 \\
A_3 &= \tau_0 + \sum_{j=1}^{p} \frac{a_j}{Z_{o3,j}} = 0 \\
y^{(i)}(0) &= 0 \quad i \in \{0, 1, 2, 3\} \\
y^{(i)}(f) &= y_f \quad i \in \{1, 2, 3, 4, 5\}
\end{align*}
\]

The variable \( Z_{o1,j} \) represents the \( j^{th} \) unstable zero powered to \( j \). Note that these conditions were chosen to force the desired torque to be symmetric. The desired \( a_i \) coefficients are then directly obtained, solving the linear system above. In this way it is possible to calculate the transient solution of system (9).

The next step is to obtain the particular solution of system (11). The coefficients \( B_i \) are obtained as linear functions of the output coefficients \( a_i \) substituting (11) and (7) into (17). As introduced in section 3, equation (15), in order to bring some robustness to this control, a closed-loop form should be used. Two types were chosen:

- (1) A partial state feedback, based on the joint position and velocity variables.
  \[
  T_{cl} = T_{ol} + K_p(\theta_d(t) - \theta(t)) + K_v(\dot{\theta}_d(t) - \dot{\theta}(t))
  \]
  \( K_p > 0, \quad K_v > 0 \)

- (2) A open loop form where the input to the system is the angle of the joint.

4. PATH-FOLLOWING FOR ONE LINK NON-MINIMUM PHASE ROBOT

The objective of the path-following method is to force the non-minimum phase system output to follow a geometric path without a timing law assigned to it. Systems with unstable zero dynamics have limited tracking capabilities. The only way to solve this performance limitation is to change the input-output structure of the system. This structure can be changed by reformulating the problem as path-following, rather than reference tracking. With this reformulation, it is possible to add a new timing law \( \gamma(t) \) that becomes an additional control input. In this section, we define the path-following problem, with Internal Model Control, and it has the goal to achieve asymptotic tracking of reference signals as is demonstrated by Aguiar et al.
The controller that incorporates an internal model of the exosystem is capable of ensuring an asymptotic convergence of the tracking error to zero for every possible reference signal generated by the exosystem. The following linear time-invariant system is assumed:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \quad x(t_0) = x_0 \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]  

(21)

where \(x(t)\) is the state, \(u(t)\) is the input, and \(y(t)\) is the output. The main objective of this method is for \(t \to \infty\) to reach and follow a desired geometric path \(y_d(\gamma)\). The geometric path \(y_d(\gamma)\) can be generated by an exosystem of the form:

\[
\begin{align*}
\frac{d}{dt}w(\gamma) &= S \times w(\gamma) \quad w(\gamma_0) = w_0 \\
y_d(\gamma) &= Q \times w(\gamma)
\end{align*}
\]  

(22)

where \(w \in \mathbb{R}^{2n}\) is the exogenous state and \(S + S^T = 0\). For any timing law \(\gamma(t)\), the path-following error can be defined as:

\[
e(t) = y(t) - y_d(\gamma(t))
\]  

(23)

The following problems can be associated to the Path-following methodology:

**Geometric path-following:** For the desired path \(y_d(\gamma)\), it is necessary to design a controller that achieves:

- **Boundness:** the state \(x(t)\) is uniformly bounded for all \(t \geq t_0\), and every initial condition \((x(t_0), w(\gamma_0)), \gamma_0 = \gamma(t_0)\).
- **Error convergence:** the path-following error \(e(t)\) converges to zero as \(t \to \infty\).
- **Forward motion:** \(\dot{\gamma}(t) > c\) for all \(t \geq t_0\), where \(c\) is a positive constant.

**Speed-assigned path-following:** Given a desired speed \(v_d > 0\), it is required that \(\gamma \to v_d\) as \(t \to \infty\).

As demonstrated by Aguiar et al. [2005] and Aguiar [2005], we can always assume a small \(L_2\)-norm of the path following error,

\[
J = \int_0^\infty \|y(t) - y_d(t)\|^2 \, dt = \int_0^\infty \|e(t)\|^2 \, dt < \delta
\]  

(24)

that verifies a \(\delta\) arbitrarily small in order to consider a perfect tracking problem.

### 4.1 Controller Design - Internal model control

One way to control the non-minimum phase system is represented in fig. 4.

Aguier et al. [2005] presented one solution to achieve a path controller for (21), such that the closed loop system state is bounded. If \((A, B, C, D)\) is a non-minimum phase system, the pair \((A, B)\) is stabilizable, the pair \((C, A)\) is detectable, the number of inputs is as large as the number of outputs \((m \geq q)\) and the zeros of \((A, B, C, D)\) do not coincide with the eigenvalues of \(S\) (22), then for the geometric path-following problem there exist constant matrices \(K\) and \(L\), and a timing law \(\gamma(t)\) such that the feedback law is:

\[
u(t) = Kx(t) + L(\dot{\gamma}_d)w(\gamma(t))
\]  

(25)

To calculate the matrices \(K\) and \(L\), the following internal model approach is considered:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \quad x(t_0) = x_0 \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]  

As shown before, the Sylvester equations (26), are solvable if the system \((A, B, C, D)\) is right-invertible and its zeros do not coincide with the eigenvalues of \(v_d \times S\). The methodology to solve this equation is described as follows:

- **Transform system (26) into the following system:**

\[
\begin{align*}
NewA &= \begin{bmatrix} Kron(I_{n_a}, A) - Kron(S^T, I_{n_a}) \\
Kron(I_{n_a}, C) \\
Kron(I_{n_a}, B) \\
Kron(I_{n_a}, D) \end{bmatrix} \\
NewB &= \begin{bmatrix} 0 \\
\Pi \\
Q \end{bmatrix}
\end{align*}
\]  

(27)

where \(n_a\) is the size of the square matrix \(S\) and \(n_a\) is the size of the square matrix \(A\).

- **From equation (27) and (26) the following formula is obtained:**

\[
\begin{align*}
\Pi = [NewA]^{-1} [NewB]
\end{align*}
\]  

(28)

- **Since we now have \(\Pi\) and \(\Gamma\) it is possible to obtain the controller gains \(K\) and \(L\). \(K\) is calculated by a minimum quadratic regulator, that minimizes the quadratic cost function,**

\[
J(u) = \int_0^\infty \left( x^T U x + u^T R u + 2x^T N u \right) dt
\]  

(29)

and \(L\) is equal to:

\[
L = \Gamma - K \times \Pi
\]  

(30)

Now that the path controller design is complete, an evolution rule to \(\gamma\) has to be created, in a way that:

\[
\begin{align*}
\lim_{t \to \infty} \gamma &= \gamma_d \\
\lim_{t \to \infty} \dot{\gamma} &= v_d
\end{align*}
\]  

5. SIMULATION RESULTS

In this section we report some simulations results on the IST planar flexible manipulator shown in fig.1.

5.1 Motion Planning - Control where the input is the torque

The method was applied for \(t_f = 2.7s\) and \(y_f = -0.35m\). The closed loop control was obtained using \(K_p = 4\) and
Fig. 5. Simulation tracking error

Fig. 6. Simulated deformation of the End-Effector

Fig. 7. Simulated close-loop torque

\( K_v = 0.03 \). In figs. 5 and 6 we present the simulation tracking error and the corresponding displacement of the end-effector. After a brief analysis, we obtain the stationary error equal to \( 2.5 \times 10^{-7} \) m, and the vibration on the end-effector equal to about \( 2 \times 10^{-7} \) m. Those are very small values compared to those obtained during the path evolution. The minor residual oscillations (\( t > 2.5s \)) verified in fig. 5 is due to the fact that the simulated model is quadratic in deformation [Martins et al., 2002], as explained in section 2.1.

Fig. 7 presents the output torque of the close-loop form. Indiscernible in this figure, are the small differences from the desired due to the addition of the tracking error.

\[
\begin{align*}
\frac{y}{\theta} &= \frac{1330s^4 + 1.398 \times 10^{-9}s^3 - 7.079 \times 10^7s^2 + 7.925 \times 10^8s^2 + 6.393 \times 10^9s + 2.907 \times 10^{11}}{s^6 + 334.3s^5 + 7.198 \times 10^8s^4 + 1.456 \times 10^9s^3 + 4.93 \times 10^7s^2 + 1.672 \times 10^{11}}
\end{align*}
\]

(31)

5.2 Control where the input of the system is the angle of the joint.

In this type of controller, the main idea is to make the system robust to external factors such as joint friction. Instead of calculating the transfer function between the torque and the position of the end-effector, the transfer function between the angle of the joint and the position of the end-effector has been calculated.

Calculating the desired angle in a way that the undesired transient response is eliminated, figure 8 is obtained. Fig. 9 presents the representation on a block diagram of the PD joint controller used, where variable \( P = -50 \).

Fig. 8. Desired \( \theta \)

Fig. 9. PD controller

By observing figs. 10 to 12, we notice that the stationary error has been reduced to an insignificant value. Using this controller the system has become more robust to external factors, such as joint friction, and reduces the vibration in the steady state.
5.3 Path-following results

The variable \( y_d \) was used as a desired path output. Two kind of paths were considered:

The first path is: (32):
\[
y_d = y_f \sin\left(\frac{\gamma \pi}{2}\right); \quad \gamma \in [0, 1]
\]
\[
v_a = \begin{cases} 
\pm \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{2 - a_1}{a_2}\right) + \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{1 - \gamma - a_1}{a_2}\right), & \gamma \in [0, 0.5[ \\
\pm \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{2 - a_1}{a_2}\right) - \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{1 - \gamma - a_1}{a_2}\right), & \gamma \in [0.5, 1]\end{cases}
\]
\[
\dot{v}_a = \begin{cases} 
- \frac{m_s}{\pi \sqrt{a_2}} \left(\frac{a_2}{a_2 + (1 - \gamma - a_1)^2}\right), & \gamma \in [0, 0.5[ \\
- \frac{m_s}{\pi \sqrt{a_2}} \left(\frac{a_2}{a_2 + (1 - \gamma - a_1)^2}\right), & \gamma \in [0.5, 1]\end{cases}
\]

where the parameter \( a_1 \) sets the width of the low speed regions at the beginning and the end of the path, while \( a_2 \) smoothen the square wave, and the \( m_s \) variable constrains the maximum speed. As expected, the obtained path in \( x, y \) coordinates is the arc represented in fig. 13.

Using the same path example as in Skjetne [2005], the second path is (33):
\[
y_d = y_f \sin\left(\frac{\gamma \pi}{2}\right); \quad \gamma \in [0, 1]
\]
\[
v_a = \begin{cases} 
\pm \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{2 - a_1}{a_2}\right) + \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{1 - \gamma - a_1}{a_2}\right), & \gamma \in [0, 0.5[ \\
\pm \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{2 - a_1}{a_2}\right) - \frac{m_s}{\sqrt{a_2}} \arctan\left(\frac{1 - \gamma - a_1}{a_2}\right), & \gamma \in [0.5, 1]\end{cases}
\]
\[
\dot{v}_a = \begin{cases} 
- \frac{m_s}{\pi \sqrt{a_2}} \left(\frac{a_2}{a_2 + (1 - \gamma - a_1)^2}\right), & \gamma \in [0, 0.5[ \\
- \frac{m_s}{\pi \sqrt{a_2}} \left(\frac{a_2}{a_2 + (1 - \gamma - a_1)^2}\right), & \gamma \in [0.5, 1]\end{cases}
\]

\[\text{Fig. 12. Simulated error}\]

\[\text{Fig. 13. The path in } x, y \text{ coordinates}\]

For the results presented in the next section, the following data was used. The system state space representation in continuous mode is
\[
A = \begin{bmatrix}
0 & 0 & 0 & 0 & 16040.41 & 9038.62 \\
0 & 0 & 0 & 0 & 11436.68 & 2422.93 \\
0 & 0 & 0 & -81331.5 & -57923.19 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]
\[B = [607.9 -296.4 -364.7 0 0 0]^T\]
\[C = [0 0 0 0.575 1 0]^T\]
\[D = 0\]

while in this representation, the states are represented in the following way
\[
q = \begin{bmatrix}
\dot{\theta} \\
\dot{\eta}_1 \\
\dot{\eta}_2 \\
\theta \\
\eta_1 \\
\eta_2
\end{bmatrix}
\]

where the variable \( \theta \) represents the joint angle. For the controller calculus we used the Matlab LQR command, with the matrix \( A \) and \( B \) (34) and the matrix \( Q \) and \( R \) with the following values.
\[
Q = 100 \times C^T \times C \\
R = 0.001 \times I
\]
which results in the following gain state space matrix:

$$K = \begin{bmatrix} -10.2518 & -16.8531 & -2.3238 & -181.8310 \\ -126.2391 & 8.1224 \end{bmatrix}$$  \hspace{1cm} (37)

The value of the controller $L$ is calculated for different values of speed assignments between 0 m/s and 5 m/s.

**First path simulation results:** The first path simulation has the following properties:
- A final $y$ value $y_f = \pi/8$ in a five seconds simulation
- $\omega_d$ equal to 20.

![Fig. 14. First Path - Evolution of the desired output versus the path variable](image)

In fig. 14 we present the desired $y_d$ versus the $\gamma$ variable. In figs. 15 and 16 we present the simulation tracking error and the corresponding deformation of the end-effector. The tracking error and the end-effector deformation converge to zero value, as expected.

![Fig. 15. First Path - Simulation tracking error](image)

**Second path simulation results:** The second path shows better results in the end-effector deformation domain, because the path is smoother and there was no need to introduce initial conditions in the integrators. This was due to the characteristics of the $v_s$ variable which already has initial values, so the path could evolve naturally. Also, the mean velocity is larger in the first path than in the second path, which could affect the end-effector deformation results.

By changing the variables of equation (33), two paths will be produced. First, $a_1 = 0.1$, $a_2 = 0.1$, $m_s = 0.1$ and $y_f = 0.575 \times \pi/2$ will be considered. This means that the end-effector will produce a 90 degree rotation. As it is seen in fig. 18, the desired velocity profile has non-zero initial and final conditions. So there is no need to introduce integrator’s initial conditions.

In fig. 19 the simulation tracking error is presented. It is visible that the maximum error appears at the maximum speed, and stabilizes to a very small value near zero. One thing that can be highlighted is the fact that the final value of the error is negative. This means that the system has
passed by a small value, and its cause can be due to the acceleration profile, since it has non zero initial and final values.

Fig. 19. Second Path - Simulation tracking error

Fig. 20 presents the evolution of the path variable during the simulation period. Comparing to fig. 17 it is observable that this path is smoother, and the control variable has better results. As in fig. 19 the final values of the path variable (fig. 20) can be due to the acceleration profile. This kind of property appeared to be no problem, since this controller is very robust to all kind of signals. Analyzing

Fig. 20. Second Path - Evolution of the desired path variable versus the real path variable

fig. 21, it is observed that the end-effector deformation is very small, and this method, with this path is capable of completely removing it all. Other tests were made demonstrating that the deformation does not depend on how long the path is, opposing the results in chapter 3. For the second path example, the robustness of the controller was studied for different changes of the velocity profile. Considering the path in equation (33), with \( a_1 = 0.1 \), \( a_2 = 1 \), \( m_s = 0.2 \) and \( y_f = 0.575 \times \pi/2 \), the end-effector will produce a 90 degree rotation.

With these variables, the velocity profile presented in fig. 22 has higher maximum velocity in a shorter period of time. With a normal controller, a larger end-effector deformation should be expected. Nevertheless, this controller defines the system velocity in order to reduce the error and the end-effector deformations, so, in contrary to what should be expected, the error and end-effector deformation values didn’t change. Comparing fig. 23 with 21 and fig. 24 with 19 we observe the end-effector deformation remains

Fig. 21. Second Path - Deformation of the end-effector

Fig. 22. Second Path - Simulation velocity profile similar, excepting the undershoot at the beginning of the path. As in the previous example (fig. 21), the controller could remove all system deformation. Analyzing the tracking error, all the characteristics remain (stationary error and values).

Fig. 23. Second Path - Deformation of the end-effector

6. EXPERIMENTAL RESULTS

In this section we present experimental results for the motion planing experiments

6.1 End-effector motion planning

Due to the importance of friction on the joints, the controller where the control variable is the angle of the joint has been considered. This control presents the best
results when an external torque perturbation is applied. In a first analysis, the transfer function from equation (16) was used to calculate the trajectory control. In real experimentation, the way to calculate the gains $K_p$ and $K_v$ were obtained in a different way than in simulation.

1. Set $K_v$ value equal to zero.
2. Introduce a $K_p$ value high enough to make the system very fast on a response to a step and with a slight oscillation.
3. After the previous two steps, change the $K_v$ value in order to eliminate the oscillation.

The values of $K_p$ and $K_v$ were 600 and 8 respectively. In order to achieve better results, the physical system had to be identified, since the location of poles and zeros aren’t the same as in simulation.

Identification: The identification of the system was obtained using an input step with amplitude of 0.02 rad and a period of 64 s (figs. 25 and 26).

Control: Now that the system is identified, with recalculation of the trajectory planning and new transfer function, the system appears to be well behaved since there is almost no vibration on the end-effector (fig. 28). It’s important to notice that if the average velocity of the trajectory becomes too high, the effect of the non-linearities increases, causing some vibration on the end-effector (fig. 29). If the trajectory time is reduced for the same amplitude, the torque must increase (figs. 30 and 31).

\[ y = \frac{0.9422s^5 - 1.481s^4 - 4.955 \times 10^6 s^3 + 7.761 \times 10^5 s^2 - 0.703 \times 10^6 s + 9.18 \times 10^2}{s^8 + 1625s^7 + 7.281 \times 10^5 s^6 + 2.544 \times 10^5 s^5 + 3.918 \times 10^3 s^4 + 7.168 \times 10^2 s^3 + 1.603 \times 10^1 s^2 + 1.168 \times 10^1 s + 1.603 \times 10^0} \]
Friction has an important role in the real system. Comparing fig. 7 with figs. 30 and 31 we observe that the undershoot presented on fig. 7 no longer exists. This behavior is due to existing friction on joints, acts as a resistive torque. Therefore, there is no need to add a negative torque to stop the robot arm. Regarding the control where the input of the system is the angle of the joint, it is evident that this kind of control is more robust and efficient than the control where the input is the torque. Thus, this was the controller chosen for the experimental results, due to the enormous joint friction.

Path following is still under intensive study. The results prove that the path-following controller removes dynamical limitations which reference tracking controller cannot.

Since the reference-tracking controller is an open-loop controller, when the reference achieves the final value, the controller becomes passive and it doesn’t observe the end-effector deflection, resulting in a permanent vibration. The path-following ability to separate the dynamic follower controller from the states boundedness controller, improves the control actuation. Even when the system has reached the final value, it still removes the external perturbation. It becomes evident that the path-following controller is a much more developed and robust controller than the reference tracking controller.

REFERENCES


