Control Strategy for a Rigid Ball Trapped between Parallel Plates

Amadeu Carceller, Guillem Pascual, and Aleix Salvador
Tutor: Jaume Franch
(Dated: June 8, 2023)

This paper provides a control strategy for achieving precise motion control of a rigid ball trapped between two parallel plates. The model equations provide a driftless system which, after applying dynamic feedback, is transformed into Brunovsky form. This enables us to design a controller for the linear system and to find the inputs for the original system by inverting the dynamic feedback. To assess the effectiveness of our approach, we conduct simulations to test the control strategy. The results demonstrate that our proposed strategy successfully achieves the desired motion control of the rigid ball given certain initial conditions.

I. INTRODUCTION

Certain systems that can be mathematically modelled can be challenging to control. Different techniques can be used to tackle the same problem. In this paper we will consider the model proposed by Sampei, Mizuno, and Ishikawa [1]. However, while they tried to control it by using time-state control form, we will instead attempt to use the control strategy of differential flatness [2], that consists in linearizing the system through a dynamic system feedback. We use prolongations[3], which is a particular case of dynamic feedback.

II. SYSTEM

A. Model of the system

The model of the system that we will try to control can be seen in figure 1. It is parallel to the floor and consists of two arms with fixed roots connected to a board. Each of the arms is made up of two links connected by a free joint. The first link of each arm can be rotated at a certain angular velocity by motors 1 and 2 respectively. The angular velocity of the board can be controlled by motor 3. A rigid ball is placed between the board and

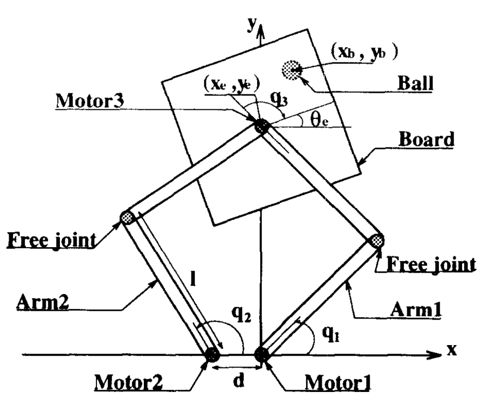

FIG. 1: Model of the system [1]

the floor, and it can only move when it is rolled by the board. It is assumed that the ball does not slide and that the board is large enough so that the ball will always remain under it.

B. State and input variables

The following variables will define the state of our system:

(x_b, y_b) : Position of the ball relative to the centre of the board
(x_e, y_e) : Coordinates of the centre of the board
(q_1, q_2) : Angle of the first link of each arm
θ_e : Angle of the board
(x_g, y_g) : Coordinates of the goal point

The origin of coordinates is defined at motor 1. The notation used to describe the state of the system x can be seen below,

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} x_b \\ y_b \\ x_e - x_g \\ y_e - y_g \\ \theta_e \end{pmatrix}$$

and the input u,

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{\theta_e} \end{pmatrix}$$

C. State equations of the system

The state equation of the system $\dot{x}$ can be found by deducing how the derivative of each of the variables behaves. Given that x_g and y_g are constant, the derivatives of x_3, x_4, and x_5 can be directly related to the input vector.

$$\begin{pmatrix} \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \end{pmatrix} = \begin{pmatrix} \dot{x_e} \\ \dot{y_e} \\ \dot{\theta_e} \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

The position of the ball relative to the board is only directly affected by the board, both by its linear and angular velocity. For the derivation, only x_b is considered, as y_b is equivalent. The total velocity is the sum of these two contributions.

$$\dot{x_b} = \dot{x_{b_{lin}}} + \dot{x_{b_{ang}}}$$

The point at the top of the ball, the one in contact with the board, moves twice as fast in relation to the floor as the centre of mass of the ball v_{cm}. This is because its velocity is the sum of the velocity of the centre of mass and the velocity of the rolling, which must also be v_{cm} in order for the point in contact with the floor to be stationary in relation to the floor.

Considering only the contribution of the linear velocity $\dot{x_e}$. Given that the point on top of the ball is always in contact with the board, it must have the same velocity relative to the floor $\dot{x_e}$. The velocity of the centre of mass is half of that $v_{cm} = \frac{1}{2}\dot{x_e}$. $\dot{x_b}$ corresponds to the difference between the velocity of the board and the ball.

The contribution of the linear velocity of the board is

$$\dot{x_{b_{lin}}} = \frac{1}{2}\dot{x_e} - \dot{x_e} = -\frac{1}{2}\dot{x_e}$$

In the case of the contribution of the angular velocity, the x-component of the velocity at the top of the ball in contact with the board must also be $2v_{cm_x}$.The velocity of the board at that point is $2v_{cm_x} = v_{ang_x} = wr_y = -\dot{\theta}_e y_b$. The negative sign is due to the direction in which θ_e is defined. The angular velocity of the board does not move its centre, so only the velocity of the centre of mass of the ball needs to be considered.

$$\dot{x_{b_{ang}}} = -\frac{1}{2}y_b\dot{\theta}_e$$

Combining both contributions, the behaviour of the derivatives is obtained.

$$\dot{x_b} = -\frac{1}{2}y_b\dot{\theta}_e - \frac{1}{2}\dot{x_e} = -\frac{1}{2}x_2u_3 - \frac{1}{2}u_1$$

$$\dot{y_b} = \frac{1}{2}x_b\dot{\theta}_e - \frac{1}{2}\dot{y_e} = \frac{1}{2}x_1u_3 - \frac{1}{2}u_2$$

The changed sign of the angular component of $\dot{y_b}$ is once again due to the direction of θ_e. Combining all of the results yields the state equation of the system.

$$\dot{x} = \begin{pmatrix}\dot{x_1}\\ \dot{x_2}\\ \dot{x_3}\\ \dot{x_4}\\ \dot{x_5}\end{pmatrix} = \begin{pmatrix}\dot{x_b}\\ \dot{y_b}\\ \dot{x_e}\\ \dot{y_e}\\ \dot{\theta_e}\end{pmatrix} = u_1\begin{pmatrix}-\frac{1}{2}\\ 0\\ 1\\ 0\\ 0\end{pmatrix} + u_2\begin{pmatrix}0\\ \frac{-1}{2}\\ 0\\ 1\\ 0\end{pmatrix} + u_3\begin{pmatrix}-\frac{1}{2}x_2\\ \frac{1}{2}x_1\\ 0\\ 0\\ 0\end{pmatrix}$$

III. MATHEMATICAL RESULTS FOR NON-LINEAR CONTROL

A. Non-Linear Control and Brunovsky Form

Definition 1 *A* ***Control System*** *is a system of differential equations. Expressed in affine form:*

$$\dot{x} = F(x) + \sum_{i=1}^{m} G_i(x)\cdot u_i$$

where $x \in \mathbb{R}^n$ are the state variables, $u_i \in \mathbb{R}$ are the control variables ($i \in \{1, ..., m\}$), $F : \mathbb{R}^n \to \mathbb{R}^n$ is called the drift vector field and $G_i : \mathbb{R}^n \to \mathbb{R}^n$ are called the input vector fields.

In order to be able to provide a control strategy for a control system, it is useful to transform it into Brunovsky form. The transformation can use non-linear changes of variables and dynamic state feedback.

Definition 2 *A system is written in* ***Brunovsky Form*** *if it has the following structure (z_i are the state variables and v_i the control variables):*

$$\dot{z_i} = z_{i+1} \text{ for } i \in \{k_1 + ... + k_{j-1} + 1, ..., k_1 + ... + k_j\}$$
$$\dot{z}_{k_1+...+k_j} = v_j \quad \forall j \in \{1, ..., m\} \tag{1}$$

In Subsection (III B 2) we will show that once the system is in Brunovsky Form, a good control strategy can be easily calculated.

Some Differential Geometry concepts and basic results (Lie derivatives, Lie brackets, Distributions, the Frobenius Theorem...) are used throughout the paper. They can be found in Appendix A.

B. Results for Non-Linear Control

1. Condition for static feedback linearization

Given a multi input control system written as in Definition 1, consider the distributions:

$$D_0 =< G_1, ..., G_m >$$
$$D_1 =< D_0, ad_F G_1, ..., ad_F G_m >$$
$$\vdots$$
$$D_{n-1} =< D_{n-2}, ad_F^{n-1}G_1, ..., ad_F^{n-1}G_m >$$

Then, the control system is static feedback linearizable iff: *i)* $dim(D_i) = ctt$ $\forall i \leq n-1$, *ii)* $dim(D_{n-1}) = n$, *iii)* D_i is involutive $\forall i \leq n-1$.

2. Algorithm to find the control strategy in Brunovsky Form

The algorithm to be implemented is the following:

1. Construct the distributions given in (III B 1) and check their involutivity

2. Define the indices

   $$r_o = d_0 \qquad r_i = d_i - d_{i-1} \qquad \forall 1 \leq i \leq k$$

   Where d_i is the dimension of D_i.

   $$k_j = \#\{r_i \mid r_i \geq j\}$$

3. Find functions $(h_1, ..., h_m)$ such that $dh_i \perp D_{k_i-2}$ and are differentially independent from $(h_1, .., h_{i-1})$

4. The change of variable is provided by,

$$z = \begin{pmatrix} h_1(x) \\ L_f h_1(x) \\ \vdots \\ L_f^{k_1-1} h_1(x) \\ \vdots \\ h_m(x) \\ L_f h_m(x) \\ \vdots \\ L_f^{k_m-1} h_m(x) \end{pmatrix}$$

And the feedback law is given by,

$$\text{v}= \begin{pmatrix} L_f^{r_1} h_1(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{pmatrix} + \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix} \text{u}$$

To design the control strategy for the system in Brunovsky form, it is important to notice that the system is formed by m independent subsystems. In this chapter we will derive the control strategy for one of the subsystems (we will define its dimension to be k) without loss of generality.

Recalling Equation 1 from Definition 2, we see that really what we have is the following differential equation: $z_i^{(k)} = v_i$.

Note that to control this system we are given an initial position $z(0)$ (initial position of the k variables) and a final position $z(T)$ (k variables). In Brunovsky form, this means that we have the initial conditions for the variable z_i and its $k-1$ derivatives, and similarly for the final conditions. The simplest function that satisfies these conditions is the interpolating polynomial $P_{2k-1}(t)$ of degree $2k-1$.

Once the interpolating polynomial $P_{2k-1}(t)$ is calculated, we can find how our control variable should behave from the aforementioned differential equation. $u_i(t) := \frac{d^k}{dt^k} P_{2k-1}(t)$, which will be a polynomial of degree $n-1$.

The final step is to reverse the transformations done to v_i to get the control strategy for u_i.

IV. APPLICATION OF THE CONTROL STRATEGY

A. Dynamical extension of the system

The model obtained in (II C) is not static feedback linearizable as the following distribution is not involutive (note that F=0):

$$D =< G_1, G_2, G_3 >$$
$$[G_1, G_3] \notin D$$

Therefore a dynamic extension is needed in order to obtain the control of the system. Control u_3 has been renamed as the sixth coordinate, and its derivative will correspond to the new control v. Physically, u_3 corresponds to the angular velocity of the table, and v to it's acceleration, which will be our new control variable. The equations obtained after the dynamic extension are:

$$\dot{x} = \begin{pmatrix} \frac{-1}{2} x_1 x_6 \\ \frac{1}{2} x_1 x_6 \\ 0 \\ 0 \\ x_6 \\ 0 \end{pmatrix} + u_1 \begin{pmatrix} \frac{-1}{2} \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ \frac{-1}{2} \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \bar{u_3} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Notice that the system is no longer driftless. With this prolongation the conditions to be static feedback linearizable (III B 1, $dim(D) = n$) are fullfilled and the algorithm of linearization proposed in (III B 2) can be applied.

First, the dimensions of the following distributions are computed:

$$D_0 =< \bar{G_1}, \bar{G_2}, \frac{\partial}{\partial x_6} > \qquad dim(D_0) = 3$$
$$D_1 =< \bar{G_1}, \bar{G_2}, \frac{\partial}{\partial x_6}, [F, \bar{G_1}], [F, \bar{G_2}], [F, \frac{\partial}{\partial x_6}] >$$
$$dim(D_1) = 6 \quad \text{for } x_6 \neq 0$$

Where the fields $\bar{G_1}, \bar{G_2}$ are the ones multiplying u_1, u_2. The distributions are involutive, and therefore the system can be linearized through a static feedback and a linearization [4]. Which means that the order to which the time derivatives of $h_i(x)$ are independent with the inputs is 1. We find these functions searching for a set of three functions such that their differential is contained in the anhilator of D_0 and are differentially independent. In order to do so, mathematical software such as MAPLE can be used to obtain the following ones:

$$h_1 = x_5; \quad h_2 = 2x_1 + x_3; \quad h_3 = 2x_2 + x_4$$

From that, the change of variable can be obtained:

$$z = \begin{pmatrix} h_1 \\ \dot{h_1} \\ h_2 \\ \dot{h_2} \\ h_3 \\ \dot{h_3} \end{pmatrix} = \begin{pmatrix} x_5 \\ x_6 \\ 2x_1 + x_3 \\ -x_2 x_6 \\ 2x_2 + x_4 \\ x_1 x_6 \end{pmatrix}$$

On the other side, the new controls are related with the former ones via regular feedback and are represented in the following expression. This feedback is regular if and only if $x_6 \neq 0$:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \ddot{h_1} \\ \ddot{h_2} \\ \ddot{h_3} \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{2} x_1 x_6^2 \\ -\frac{1}{2} x_2 x_6^2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{2} x_6 & -x_2 \\ -\frac{1}{2} x_6 & 0 & x_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \bar{u_3} \end{pmatrix}$$

B. Obtaining controls

With the new variables and controls the system can be written in Brunowsky Form, which can be divided into three different blocks, one for each control. Writing down the first block (the other ones are similar):

$$\begin{pmatrix} \dot{z_1} \\ \dot{z_2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ v_1 \end{pmatrix}$$

Recalling from Section (III B 2), the interpolating polynomial must be calculated. Since we have restrictions up to the first derivative, we will use Hermite interpolation (programmed at the end of the code). The second derivative of that polynomial will be the corresponding control.

Going back to the original coordinates, the desired controls are obtained. As we have seen, it is important to notice that when the original controls are obtained by inverting the expression (1), x_6 will be dividing some terms, this means that this coordinate can never be zero or the system will explode (this was also seen when defining D_1, where $dim(D_1) = 6$ if $x_6 \neq 0$). In consequence, our system cannot rotate in both directions, as it would have to go from positive to negative angular velocities, thus passing through zero.

The simulation of the system has been carried out with Matlab. Its code can be found in Appendix B. Given the coordinates of the initial position of the ball and table, the program follows the algorithm presented in the previous sections to find the expression of the controls. With that expressions numerical integration of the system is performed to obtain the trajectories of each variable, the graphical solution is plotted:

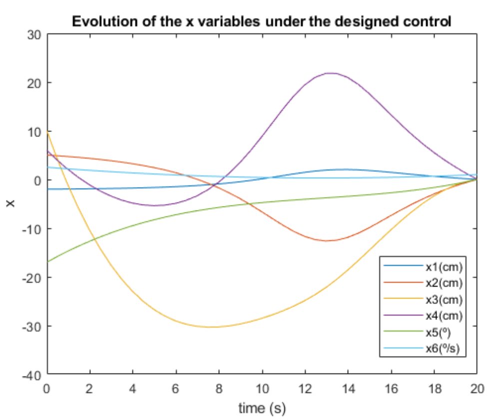

FIG. 2: Simulation of the trajectories of x with initial conditions: $x_1 = -2cm$, $x_2 = 5cm$, $x_3 = 10cm$, $x_4 = 6cm$, $x_5 = -17$, $x_6 = 2.5/s$, and $T = 20s$.

In the figure 2, all variables converge to the desired position (0, except for x_6, but this variable was artificially generated, and set to the value $x_6(T) = 1$). Additionally, the angular velocity is always positive, which prevents the system from collapsing.

1. Necessary condition for initial values

However this doesn't happen for all initial and final positions. To obtain the set of initial conditions for which this control strategy works properly, one has to examine the polynomials obtained from the interpolation.

Recalling that $\dot{z_1} = z_2$, our goal is to prevent this variable passing through zero, that is, preventing the polynomial corresponding to $(z_1(t))$ from having a change of sign in the derivative on the interval $t_0 <= t <= tf$. With our change of variables, we have $z_2(t) = x_6(t)$ and $z_1(t) = x_5(t)$.

Let's check the case $x_6 > 0$. Having a positive x_6 means that $x_5(t)$ is increasing ($\dot{x}_5(t) = \dot{z}_1(t) = z_2(t) = x_6(t) > 0$). We also know from our model definition that the final value of $x_5(T) = 0$. This will only happen if $x_5(0) < 0$.

After making a similar argument for the case $x_6 < 0$, we can express the necessary condition our initial conditions must satisfy in order for our control strategy to work: $x_5(0)$ *must be of opposite sign to* $x_6(0)$.

V. CONCLUSION

In conclusion, we have presented an effective control strategy for the system presented, modified by a dynamical extension. The mathematical machinery developed to reach a solution has been proven to be a good resource to tackle the problem. Although it contains some restrictions that have to be taken into account, mainly that the new variable can never be zero, by choosing appropriate initial and final conditions the controls obtained lead our system to the desired position.

New methods for interpolation can broaden the set of initial and final conditions for our problem. For example, we can use polynomials of superior degrees that give us more freedom to find solutions for conditions that fail when using only degree three, non-polynomic interpolations can also be explored.
The method we have applied in this system can be applied to other problems with similar dynamic equations. If the system is driftless with three controls, and the following conditions are fullfilled:

$$[G_1, G_2] \in < G_1, G_2 >$$
$$dim < G_1, G_2, G_3, [G_1, G_3], G_2, G_3 >= 5$$

Then it can be proven that one can always proceed as shown in our system and obtain effective controls.

[1] M. Sampei, S. Mizuno, and M. Ishikawa, "Motion control of nonholonomic system with rolling constraint," , vol. 1, no. 1, pp. 534–537, 1995.

[2] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International journal of control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[3] J. Franch and E. Fossas, "Linearization by prolongations: New bounds on the number of integrators," *European journal of control*, vol. 11, no. 2, pp. 171–179, 2005.

[4] A. Isidori and A. Isidori, "Local decompositions of control systems," *Nonlinear control systems*, pp. 1–76, 1995.

A. Differential Geometry concepts and results

Definition 3 *The **Lie derivative** of a function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to a vector field $F : \mathbb{R}^n \to \mathbb{R}^n$ is given by:*

$$(\pounds_F f)(\cdot) = \nabla_{F(\cdot)} f(\cdot)$$

where ∇_v indicates the derivative in direction $v \in \mathbb{R}^n$

Definition 4 *The **Lie Bracket** between two vector fields $F, G : \mathbb{R}^n \to \mathbb{R}^n$ is another vector field $[F, G] : \mathbb{R}^n \to \mathbb{R}^n$ given by:*

$$[F, G](\cdot) = \nabla_{F(\cdot)} G(\cdot) - \nabla_{G(\cdot)} F(\cdot)$$

Where ∇_v is applied to each component of the vector field separately.

The Lie Bracket will also be denoted by $[F, G] = ad_F G$. With multiple iterations written as $[F, [F, G]] = ad_F^2 G$ and so on

Definition 5 *A **Distribution** D is the subspace generated by a set of vector fields $F_1, ..., F_d$:*

$$D(\cdot) =< F_1(\cdot), ..., F_d(\cdot) >$$

*A Distribution is **Involutive** if $[F_i, F_j] \in D \; \forall i, j$.*

Definition 6 *The **Annihilator** of a distribution $D =< F_1, ..., F_d >$ is $D^\perp =< \omega_1, ..., \omega_{n-d} >$ where ω_i are 1-forms such that $\omega_i(F_j) = 0 \; \forall i, j$.*

An Annihilator $D^\perp$ is generated by exact forms if $\forall i$ $\exists \lambda_i : \mathbb{R}^n \to \mathbb{R}$ such that $\omega_i = d\lambda_i$.

Theorem 1 *(**Frobenius theorem**) A Distribution D is involutive iff its Annihilator $D^\perp$ is generated by exact forms.*

Lemma 1 *Given an involutive distribution $D =< G_1, ..., G_d >$ invariant under the effect of a vector field F (i.e. $[F, G] \in D \; \forall G \in D$), with non-changing dimension $dim(D) = d \implies$*

$\implies \exists z = \phi(x)$ *diffeomorphism such that:*

$$F(x) = F(\phi^{-1}(x)) = \bar{F}(\vec{z}) = \begin{pmatrix} \bar{F}_1(\vec{z}) \\ \vdots \\ \bar{F}_d(\vec{z}) \\ \bar{F}_{d+1}(z_{d+1}, ..., z_n) \\ \vdots \\ \bar{F}_n(z_{d+1}, ..., z_n) \end{pmatrix}$$

Note that the last $n - d$ components of $\bar{F}$ only depend on the last $n - d$ components of $\vec{z}$.

B. Simulation code

```
%INITIAL CONDITIONS and final time T
T=20;
x1=-2;
x2=5;
x3=10;
x4=6;
x5=-17;
x6=2.5;
x0=[x1;x2;x3;x4;x5;x6];

%Initial conditions for z
z1=x5;
z2=x6;
z3=2*x1+x3;
z4=-x2*x6;
z5=2*x2+x4;
z6=x1*x6;

%CONTROL STRATEGY
%Finding the interpolating polynomial
%subsystem 1
if (x6>0) %since x6 cannot cross 0
  [a0,a1,a2,a3]=hermite(z1,z2,0,1,T);
end
if (x6<0) %since x6 cannot cross 0
  [a0,a1,a2,a3]=hermite(z1,z2,0,-1,T);
end

%subsystem 2
[b0,b1,b2,b3]=hermite(z3,z4,0,0,T);

%subsystem 3
[c0,c1,c2,c3]=hermite(z5,z6,0,0,T);

% v_i = p_i''(x)
v1 = @(t,x) 2*a2+6*a3*t;
v2 = @(t,x) 2*b2+6*b3*t;
v3 = @(t,x) 2*c2+6*c3*t;

%Reversing the change of variables
```

```
u1=@(t,x) 2*x(1)*(v1(t,x))/x(6)-2*(v3(t
    ,x))/x(6)-x(2)*x(6);
u2=@(t,x) 2*x(2)*v1(t,x)/x(6)+2*v2(t,x)
    /x(6)+x(1)*x(6);
u3=@(t,x) v1(t,x);

% INTEGRATION
% Defining our system
dxdt = @(t, x)
         [-0.5*u1(t,x)-0.5*x(2)*x(6);
         -0.5*u2(t,x)+0.5*x(1)*x(6);
         u1(t,x);
         u2(t,x);
         x(6);
         u3(t,x)];

% Using a numerical integrator
[t, x] = ode45 (dxdt, [0 T], x0);


%PLOT
plot(t, x)
title('Evolution of the x variables
    under the designed control')
xlabel('time (s)')
ylabel('x')
legend (["x1(cm)" "x2(cm)" "x3(cm)" "x4
    (cm)" "x5(deg)" "x6(deg/s)"])




%Function used to design the
    interpolating pol.
function[c0,c1,c2,c3] = hermite(z0,dz0,
    zf,dzf,T)
     f00 = dz0;
     f01=(zf-z0)/T;
     f11 = dzf;
     f001=(f01-f00)/T;
     f011=(f11-f01)/T;
     f0011=(f011-f001)/T;

     c0=z0;
     c1=f00;
     c2=f001-f0011*T;
     c3=f0011;
end
```