



# UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales  
Departamento de Física

Física Aplicada a Manipuladores Robóticos

Por

Facundo Pessacg

**Director del trabajo:** Dr. Damian Oliva

**Co-director del trabajo:** Dr. Pablo de Cristóforis

**Lugar de trabajo:** Departamento de Ingeniería en Automatización y Control Industrial - Universidad de Quilmes, y Laboratorio de Robótica y Sistemas embebidos, Departamento de Computación, FCEyN, UBA

**Tesis de Licenciatura en Ciencias Físicas**

Marzo de 2015



# Física Aplicada a Manipuladores Robóticos

Facundo Pessacg

Marzo de 2015



# Abstract

In this work, we study a number of problems related to the control of robotic manipulators. This electromechanical devices can be modeled as a set of  $N$  rigid bodies linked by  $N$  joints that are forced by electric motors. The objective of this thesis is to control the pose of the end effector of the manipulator (described by its position and orientation) to follow a desired trajectory with bounded error, even though partially known disturbances exist. To achieve this objective the following issues were addressed: (1) We solve the direct and inverse kinematics of the manipulator and study what angular value should be given to each joint for the effector to reach a desired pose. (2) We simulate two methods of trajectory generation for moving the robot between two poses of interest. (3) We study the electro-mechanical model describing the motors of the manipulator. (4) We develop a dynamic model of the joints, using the Lagrange formalism. Joining this one with the electro-mechanical model, we obtain a complete dynamic model of a manipulator with two degrees of freedom. (5) With the developed model, we study different control strategies that enable us to force the system as to asymptotically approach the end effector to the desired pose. (6) Finally, we propose a simplified dynamic model to implement and execute real-time control strategies on a microcontroller with limited computational capacity.

**Keywords:** Robotic manipulator, automatic control, kinematic and dynamic model, trajectory generator.

# Resumen

En este trabajo estudiamos una serie de problemas relacionados con el control de manipuladores robóticos. Este dispositivo electromecánico puede modelarse como un conjunto de  $N$  cuerpos rígidos vinculados por  $N$  articulaciones forzadas por motores eléctricos. El objetivo de esta Tesis es controlar la pose del efector final del manipulador, descrita por su posición y orientación, para que siga una trayectoria deseada con error acotado, aún cuando existen perturbaciones parcialmente conocidas.

Para cumplir este objetivo se abordaron los siguientes problemas: (1) Resolvimos la cinemática directa e inversa del manipulador y estudiamos qué valor angular debe darse a cada articulación para que el efector tenga una pose deseada. (2) Simulamos dos métodos de generación de trayectoria que permiten mover al robot entre dos poses de interés. (3) Estudiamos el modelo electro-mecánico que describe a los motores de un manipulador. (4) Desarrollamos el modelo dinámico de las articulaciones, utilizando el formalismo de Lagrange. Juntando este modelo con el modelo electro-mecánico, obtuvimos un modelo dinámico completo de un manipulador de dos grados de libertad. (5) Con el modelo desarrollado, estudiamos diferentes estrategias de control que permiten forzar al sistema de modo que el efector final tienda asintóticamente a la pose deseada. (6) Por último, propusimos una simplificación del modelo dinámico que permite implementar y ejecutar en tiempo real las estrategias de control en un microcontrolador, con capacidad de computo limitada.

**Palabras Clave:** Manipulador robótico, control automático, modelo cinemático y dinámico, generador de trayectorias.

# Índice general

<b>1. Modelo Cinemático</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Modelo Cinemático Propuesto . . . . .	3
1.2.1. Transformaciones Homogéneas . . . . .	3
1.2.2. Parámetros de Denavit-Hartenberg . . . . .	6
1.2.3. Modelo Cinemático Directo . . . . .	8
1.2.4. Modelo Cinemático Inverso . . . . .	11
1.3. Implementación . . . . .	13
1.3.1. Simulaciones . . . . .	13
1.3.2. Manipulador Real . . . . .	15
1.4. Experimentos y Resultados . . . . .	18
1.4.1. Manipulador Real . . . . .	18
<b>2. Generación de Trayectorias</b>	<b>23</b>
2.1. Esquemas en el Espacio de las Articulaciones . . . . .	24
2.1.1. Polinomios Cúbicos . . . . .	25
2.2. Esquemas en el Espacio Cartesiano . . . . .	28
2.2.1. Control para el Seguimiento de una Trayectoria en el Espacio Cartesiano . . . . .	31
<b>3. Modelo Electro-Mecánico, Identificación y Simulación de una   Articulación</b>	<b>35</b>
3.1. Modelo Electro-Mecánico de un Motor de Corriente continua (DC) . . . . .	36
3.1.1. Incorporación de Carga . . . . .	38

3.2. Identificación del Motor . . . . .	39
<b>4. Modelo Dinámico y simulación de control</b>	<b>49</b>
4.1. Modelo Dinámico de un Manipulador RR . . . . .	52
4.1.1. Dinámica Lagrangiana de Manipuladores . . . . .	52
4.1.2. Cálculo Automático de las Ecuaciones de Movimiento para un manipulador RR . . . . .	56
4.1.3. Modelo Dinámico Completo . . . . .	57
4.2. Simulación del Control de un Manipulador Simple por el Méto- do de Computación de Torque . . . . .	58
4.2.1. Control por el Método de Computación de Torques . .	59
4.2.2. Análisis de los Términos del Modelo Completo . . . . .	61
4.2.3. Control del RR . . . . .	62
<b>5. Control de una articulación</b>	<b>67</b>
5.1. Implementación . . . . .	67
5.2. Control de un Motor DC . . . . .	69
<b>6. Conclusiones</b>	<b>73</b>
<b>A. Software</b>	<b>77</b>



# Capítulo 1

## Modelo Cinemático

### 1.1. Introducción

Actualmente, los manipuladores robóticos son utilizados en numerosas actividades humanas, como puede ser: la industria (soldadura, paletizado, pintura, ensamble, etc.), medicina (operaciones quirúrgicas), agricultura de precisión (cosecha de frutos, extracción de maleza, relevamiento de cultivos), actividad espacial (recolección de muestras), minería, manipulación de residuos o elementos peligrosos para la vida humana, entre otras aplicaciones [1], [2], [3].

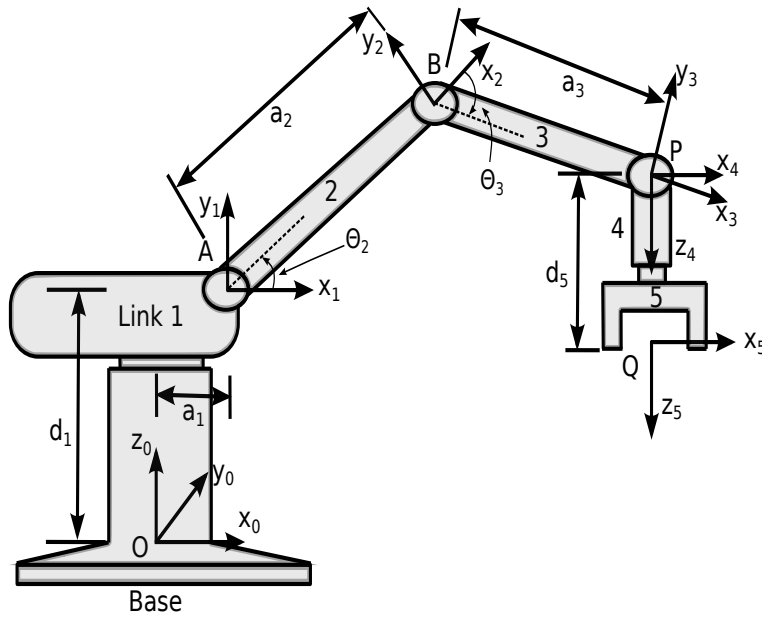
Un manipulador robótico está compuesto por una serie de eslabones unidos mediante articulaciones o juntas. Cada articulación tiene asociado sistema de referencia. El sistema de referencia inicial corresponde a la base del manipulador, mientras que el último corresponde al efector final. El efector final es el dispositivo o herramienta que se ubica en el extremo del brazo robótico y permite la interacción con el entorno, su diseño depende de la aplicación particular. Las posiciones en el entorno a las cuales puede llegar el efector final determinan el espacio de trabajo del manipulador, mientras que la cantidad de juntas determinan los grados de libertad del mismo. Para realizar una tarea con un manipulador es necesario poder dirigir el efector final de un lugar a otro en coordenadas del mundo. Para ello, se requiere controlar cada una de sus juntas.

La cinemática de un manipulador es la descripción del movimiento de las articulaciones del mismo, sin tener en cuenta las fuerzas o momentos que intervienen. En un análisis cinemático, la posición, la velocidad y la aceleración de todas las articulaciones se calculan con respecto a un sistema de referencia fijo al mundo (trama base). *Los métodos cinemáticos se utilizan para poder describir cualquier trayectoria deseada.*

Para manipular objetos la herramienta o efector final del robot debe seguir una trayectoria planeada. El modelo cinemático relaciona la posición y orientación del efector final y las posiciones espaciales de cada articulación. Podemos pensar el problema de dos formas: el modelo cinemático directo, que establece la posición y orientación del efector final en función de las coordenadas articulares; y el modelo cinemático inverso, que permite calcular todas las soluciones posibles para las coordenadas articulares partiendo de una posición y orientación conocidas del efector final. Existen diversos métodos para resolver la cinemática directa e inversa. En este trabajo seguimos el enfoque propuesto en [4] y [5].

En la Figura 1.1 se muestra el modelo cinemático utilizado. Como puede verse, el manipulador está compuesto por una serie de eslabones (1, 2, 3, 4, 5) unidos mediante articulaciones o juntas (O, A, B, P, R). Cada articulación tiene asociado un sistema de referencia o trama. El sistema de referencia inicial o trama base (que denominaremos trama  $\{0\}$ ) corresponde a la base del manipulador, mientras que el último corresponde al efector final (que denominaremos trama  $\{N\}$ ). Las posiciones en el entorno que pueden ser alcanzadas por el efector final determinan el espacio de trabajo del manipulador, mientras que la cantidad de juntas determinan los grados de libertad del mismo. Para realizar una tarea con un manipulador es necesario poder dirigir el efector final de un lugar a otro en coordenadas del mundo. Para ello es necesario poder controlar cada una de sus articulaciones.

En este capítulo se describe el modelo cinemático utilizado en esta tesis y su implementación para el manipulador robótico Scorbot-ER V Plus (ver figura 1.1).



**Figura 1.1:** Modelo cinemático propuesto para el Scrobot-ER V Plus.

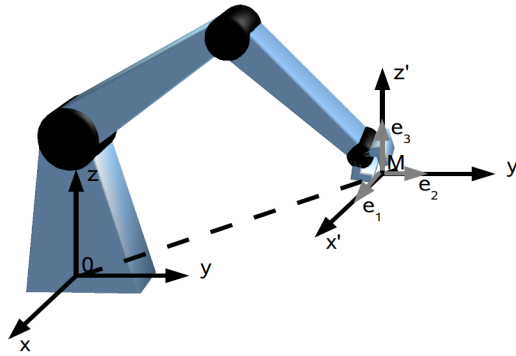
## 1.2. Modelo Cinemático Propuesto

En esta sección se describe el modelo cinemático propuesto y su implementación para el manipulador robótico Scrobot-ER V Plus. La primera parte consiste en definir el modelo del robot proporcionando sus parámetros de vínculo, la segunda en calcular la cinemática directa utilizando el modelo anterior y la tercera parte consiste en obtener la cinemática inversa. Para eso necesitamos encontrar una matriz de transformación homogénea que describa la posición y orientación cartesiana del efector final del manipulador (trama  $\{N\}$ ) a partir de las variables y parámetros de junta. Es decir, obtener la matriz  ${}^N_0T$ , que relaciona la trama  $\{0\}$  con la trama  $\{N\}$ .

### 1.2.1. Transformaciones Homogéneas

Como dijimos inicialmente, un manipulador robótico está compuesto por cuerpos rígidos (eslabones) unidos en serie por medio de articulaciones (ver fig. 1.1).

Para describir la relación entre las articulaciones del manipulador se utiliza un sistema de coordenadas homogéneas. Estas coordenadas pueden ser introducidas empleando el esquema mostrado en la figura 1.2.



**Figura 1.2:** Relación entre la trama base y la trama del efector final.

En la figura 1.2 se ha definido un sistema (absoluto) de coordenadas ortogonales  $(x, y, z)$  en la base fija  $\{0\}$  y otro sistema ortogonal  $(x', y', z')$  en la mano  $\{M\}$ . A lo largo de los ejes  $(x', y', z')$  hay respectivos vectores unitarios  $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$  ortogonales entre sí. La orientación del sistema  $(x', y', z')$  respecto del sistema  $(x, y, z)$  puede ser descrita por la matriz de rotación  $R$ .

$$R = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \quad (1.1)$$

donde  $e_{kx}$  denota coordenada  $k$  del vector  $\mathbf{x}$  ( $k = 1, 2, 3$ ).

Nótese que  $R$  es una matriz de  $3 \times 3$  inversible (no singular). Cabe destacar que no es una descripción óptima. Estamos utilizando una matriz de  $3 \times 3$  y por lo tanto 9 números para describir una orientación, pero sabemos que hacen falta solo 3 ángulos para ello. Esto se debe a que existen 6 relaciones en los componentes de la matriz  $R$  debido a que como las 3 columnas son versores, tienen norma 1 y son normales.

Sea ahora  $P$  un punto cuya posición respecto al sistema centrado  $M$  es dada por el vector  $\mathbf{q}'$ . La posición de  $P$  respecto al sistema centrado en  $0$  es dada por

$$\mathbf{p} = R\mathbf{q}' + r \quad (1.2)$$

donde  $\mathbf{r}$  es el vector  $\overline{OM}$  desde 0 hasta  $M$ .

La ecuación 1.2 puede escribirse como una sola multiplicación matricial como se muestra en 1.3.

$$\mathbf{p} = \begin{bmatrix} R & \mathbf{r} \end{bmatrix} \mathbf{q} \quad (1.3)$$

donde

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}' \\ 1 \end{bmatrix} = \begin{bmatrix} q'_x \\ q'_y \\ q'_z \\ 1 \end{bmatrix} \quad (1.4)$$

La ecuación 1.3 no se altera si se agrega una cuarta fila  $[0 \ 0 \ 0 \ 1]$  a la matriz, escribiéndola entonces como 1.5.

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = T \begin{bmatrix} \mathbf{q}' \\ 1 \end{bmatrix} \quad (1.5)$$

donde  $T$  es la matriz de transformación homogénea y viene dada por 1.6.

$$T = \begin{bmatrix} & R & \mathbf{r} & \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & r_x \\ e_{1y} & e_{2y} & e_{3y} & r_y \\ e_{1z} & e_{2z} & e_{3z} & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

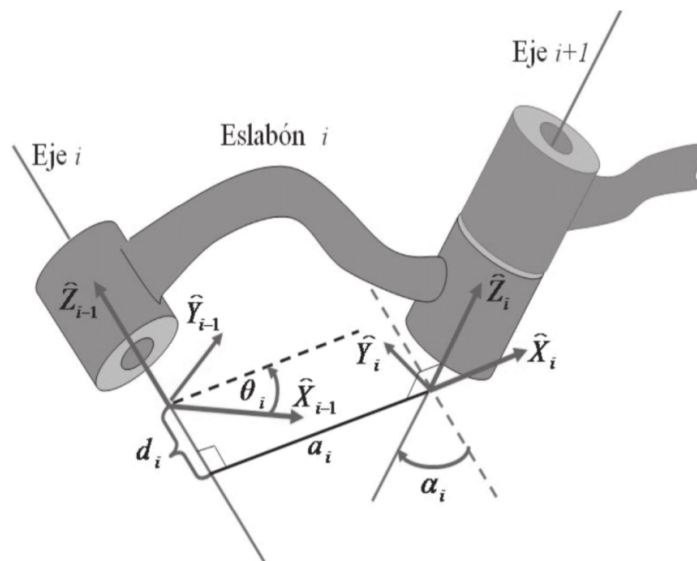
Nótese que  $P$  puede ser un punto variable que describa vértices, lados, formas de algún cuerpo o pieza manipulada por la mano.

Entonces una transformación homogénea es una matriz que relaciona dos sistemas de referencia que pueden estar rotados y/o trasladados uno respecto del otro. Una propiedad muy importante de estas transformaciones es que se pueden componer. Supongamos que se quiere hallar la transformación  $({}^1_3T)$  que relaciona el sistema de referencia 3 con el sistema de referencia 1, y se conocen las transformaciones entre sistemas de coordenadas consecutivos, entonces se puede hallar  ${}^1_3T = {}^1_2T {}^2_3T$ .

Otra propiedad importante es que las transformaciones homogéneas son inversibles, es decir,  ${}^2_1T = ({}^1_2T)^{-1}$ .

### 1.2.2. Parámetros de Denavit-Hartenberg

Para describir la cinemática de un manipulador tenemos que definir en forma metódica y apropiada los sistemas de coordenadas de cada articulación. Existen diversos métodos, siendo la notación de Denavit-Hartenberg la más usada. En esta convención se le asigna a cada articulación un sistema de coordenadas de forma tal que para hacer coincidir los sistemas  $i$  y  $i + 1$  se requiera una sucesión definida de a lo sumo dos rotaciones y dos traslaciones. Estas a su vez, se especifican en base a dos parámetros de rotación y dos de traslación, que son definidos como se muestra en la figura 1.3.



**Figura 1.3:** Relación entre dos tramas consecutivas dada por los parámetros de D-H. Figura extraída de [4].

El sistema de coordenadas de la articulación  $i$  se elige de la siguiente manera:  $z_i$  en la dirección del eje de rotación;  $x_i$  se elige paralelo a la normal común, ( $x_i = z_{i-1} \times z_i$ ), si no hay una única normal común (ejes  $z$  paralelos) entonces la dirección de  $x_i$  es de  $z_{i-1}$  a  $z_i$ ; y por último  $y_i$  se elige a partir de

$x_i$  y  $z_i$  para que formen una terna de mano derecha.

Los parámetros que describen a cada articulación son:  $d$ ,  $a$ ,  $\alpha$  y  $\theta$ .

1.  $a_i$  distancia desde  $z_{i-1}$  a  $z_i$
2.  $\alpha_i$  ángulo entre  $z_{i-1}$  y  $z_i$  medido alrededor de  $x_i$
3.  $d_i$  distancia desde  $x_{i-1}$  a  $x_i$  a lo largo de  $z_i$
4.  $\theta_i$  ángulo entre  $x_{i-1}$  y  $x_i$  medido alrededor de  $z_{i-1}$

Utilizando estos parámetros podemos calcular las transformaciones homogéneas  ${}^i{}_{i-1}T$  que describen la trama  $i$  respecto de la trama  $i - 1$ . La forma general de esta transformación viene dada por:

$${}^i{}_{i-1}T = \begin{bmatrix} c_{\theta_i} & -c_{\alpha_i}s_{\theta_i} & s_{\alpha_i}s_{\theta_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\alpha_i}c_{\theta_i} & -s_{\alpha_i}c_{\theta_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

donde  $s_{\theta_i} = \sin(\theta_i)$ ,  $c_{\theta_i} = \cos(\theta_i)$ ,  $s_{\alpha_i} = \sin(\alpha_i)$  y  $c_{\alpha_i} = \cos(\alpha_i)$ .

Para determinar estos parámetros en el manipulador Scorbot se utilizó la convención descripta en [4]. En la Tabla 1.1 se muestran los valores de dichos parámetros para el manipulador Scorbot-ER V Plus.

Art. $i$ -ésima	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	$d_1$	$a_1$	$\pi/2$	$\theta_1$
2	0	$a_2$	0	$\theta_2$
3	0	$a_3$	0	$\theta_3$
4	0	0	$\pi/2$	$\theta_4 + \pi/2$
5	$d_5$	0	0	$\theta_5$

**Tabla 1.1:** Parámetros de D-H del Scorbot-ER V Plus.  
 $d_1 = 349cm$ ,  $a_1 = 16cm$ ,  $a_2 = 221cm$ ,  $a_3 = 221cm$ ,  $d_5 = 145cm$ .

### 1.2.3. Modelo Cinemático Directo

La transformación que describe la trama del efector final respecto de la trama base viene dada por 1.8.

$${}^0_5T = {}^0_1T \quad {}^1_2T \quad {}^2_3T \quad {}^3_4T \quad {}^4_5T \quad (1.8)$$

Entonces, utilizando (1.7) y (1.8) obtenemos:

$${}^0_5T = \begin{bmatrix} s_1s_5 - s_{234}c_1c_5 & s_1c_5 + s_{234}c_1s_5 & c_{234}c_1 & x \\ -c_1s_5 - s_{234}s_1c_5 & s_{234}s_1s_5 - c_1c_5 & c_{234}s_1 & y \\ c_{234}c_5 & -c_{234}s_5 & s_{234} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.9)$$

donde:

$$\begin{aligned} x &= c_1(a_1 + a_2c_2 + a_3c_{23} + d_5c_{234}) \\ y &= s_1(a_1 + a_3c_{23} + a_2c_2 + d_5c_{234}) \\ z &= d_1 + a_3s_{23} + a_2s_2 + d_5s_{234} \end{aligned} \quad (1.10)$$

donde  $s_i = \sin(\theta_i)$ ,  $c_i = \cos(\theta_i)$ ,  $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$  y  $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$ . Sabiendo los ángulos  $\theta_i$  en los que se encuentra cada articulación y utilizando la transformación  ${}^0_5T$  vamos a obtener la posición y orientación del efector final. La transformación  ${}^0_5T$  puede pensarse como una matriz rotación  $R$  más un vector de traslación  $\mathbf{t}$  que relaciona la trama base del manipulador con la trama del efector final.

$${}^0_5T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1.11)$$

El vector de traslación  $\mathbf{t}$  corresponde a la posición del efector final y se define directamente a partir de (1.10). Para definir la matriz de rotación  $R$  vamos a utilizar (1.9) e igualarla a una matriz de rotación genérica en el sistema de coordenadas que utilicemos. Esta matriz  $R$  sirve para obtener los tres ángulos que especifican la orientación de la trama del efector final respecto de la trama base.

Cuando se tiene un sistema de coordenadas móvil (trama del efector final) respecto de uno fijo (trama base) en tres dimensiones, y se desea dar la



orientación del sistema móvil en un momento dado, existen varias posibilidades. Para este trabajo se consideraron los ángulos de dirección *pitch* ( $\beta$ ), *roll* ( $\gamma$ ) y *yaw* ( $\alpha$ ) debido a que, para el modelo cinemático utilizado, (Fig. 1.1) estos ángulos están desacoplados y por tanto, son fácilmente identificables, como se explica a continuación.

Siguiendo la convención utilizada en [5], encontramos que: *yaw* se define como una rotación de ángulo  $\alpha$  alrededor del eje  $z_0$ , *pitch* como una rotación de ángulo  $\beta$  alrededor del eje  $y_0$  y *roll* como una rotación de ángulo  $\gamma$  alrededor de  $x_0$ . Si se cumple que el sistema de coordenadas móvil y el sistema de coordenadas fijo coinciden en el instante inicial, entonces vale que:

$$R_{\alpha,\beta,\gamma} = R_{z_0}(\alpha)R_{y_0}(\beta)R_{x_0}(\gamma) \quad (1.12)$$

donde  $R_{z_0}$ ,  $R_{y_0}$  y  $R_{x_0}$  son las matrices de rotación alrededor de los ejes  $x_0$ ,  $y_0$  y  $z_0$  respectivamente. Luego, podemos expresar una forma genérica para la matriz  $R_{\alpha,\beta,\gamma}$  de la siguiente forma:

$$R_{\alpha,\beta,\gamma} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha c_\gamma s_\beta + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha c_\gamma s_\beta - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \quad (1.13)$$

donde  $s_i = \sin(i)$ ,  $c_i = \cos(i)$ .

Sin embargo, en el caso del manipulador Scorbot-ER V Plus, el sistema de coordenadas fijo (trama base) y el sistema de coordenadas móvil (trama del efector final inicialmente) no coinciden. Por lo tanto, es necesario definir una nueva trama (a la que llamamos trama herramienta) que representa la posición del efector final pero con la orientación de la trama base. La transformación que va de la trama 5 a la trama herramienta ( ${}^5_h T$ ) viene dada por la matriz:

$${}^5_h T = \begin{bmatrix} R_h & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1.14)$$

donde

$$R_h = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (1.15)$$

Tomando como sistema de coordenadas fijo la trama base y como sistema de coordenadas móvil la trama herramienta, la transformación que va de la primera a la segunda viene dada por:

$${}^0_hT = {}^0_5T {}^5_hT \quad (1.16)$$

Entonces, ahora podemos igualar la parte rotacional de  ${}^0_hT$  con la matriz de rotación genérica  $R_{\alpha,\beta,\gamma}$

$$R_{\alpha,\beta,\gamma} = R R_h \quad (1.17)$$

donde  $R$  se puede hallar de (1.9), (1.11) y  $R_h$  de (1.15). Reemplazando  $R$  y  $R_h$  se obtiene (1.18).

$$R_{\alpha,\beta,\gamma} = \begin{bmatrix} c_{234}c_1 & -s_1c_5 - s_{234}c_1s_5 & s_1s_5 - s_{234}c_1c_5 \\ c_{234}s_1 & c_1c_5 - s_{234}s_1s_5 & -c_1s_5 - s_{234}s_1c_5 \\ s_{234} & c_{234}s_5 & c_{234}c_5 \end{bmatrix} \quad (1.18)$$

Comparando (1.13) y (1.18) observamos que la relación entre los ángulos  $\alpha$ ,  $\beta$  y  $\gamma$  y los ángulos  $\theta_i$  es muy simple. En particular, los manipuladores Scorbot-ER V Plus poseen 5 grados de libertad, 3 de traslación y 2 correspondientes a *pitch* ( $\beta$ ) y *roll* ( $\gamma$ ), es decir, no podemos controlar el *yaw* ( $\alpha$ ). Teniendo en cuenta esto, los ángulos  $\alpha$ ,  $\beta$  y  $\gamma$  queda definidos según (1.19).

$$\begin{aligned} \alpha &= \theta_1 \\ \beta &= -(\theta_2 + \theta_3 + \theta_4) = -\theta_{234} \\ \gamma &= \theta_5 \end{aligned} \quad (1.19)$$

Es importante destacar que el modelo descrito en este trabajo presenta similitudes con el modelo desarrollado en [6], [7],[8],[9]. Una diferencia importante radica en las tramas asignadas a cada vinculo del robot (ver Tabla

1.1 y Figura 1.1). Esto hace que, a diferencia del modelo [6] y [7], en nuestro modelo la punta del robot suba cuando aumentamos los valores de  $(\theta_2, \theta_3, \theta_4)$  en forma similar a la convención de fábrica. Cabe destacar que decidimos rehacer los cálculos debido a inconsistencias en las ecuaciones presentadas en [9] y [8]. Los cálculos analíticos se realizaron con el *toolbox* simbólico de Matlab.

### 1.2.4. Modelo Cinemático Inverso

La cinemática inversa consiste en calcular los ángulos de cada link o junta  $(\theta_i, i \in [1, 5])$  partiendo de la posición  $(\mathbf{t} = (x, y, z)^t)$  y orientación del efector final (*pitch* ( $\beta$ ) y *roll* ( $\gamma$ )). La forma general de (1.16) viene dada por (1.20).

$$\begin{bmatrix} R_{\alpha, \beta, \gamma} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = {}^0T_5 {}^5T_h \quad (1.20)$$

Utilizando (1.20) y conociendo  $\mathbf{t}$ ,  $\beta$  y  $\gamma$  tenemos que hallar cada uno de los  $\theta_i$ . Igualando los elementos de matriz (1,3) y (2,3) de (1.20), se obtiene el valor para  $\theta_1$ . El resultado se muestra en (1.21).

$$\theta_1 = \text{atan2}(y, x) \quad (1.21)$$

Para encontrar los ángulos  $\theta_2$  y  $\theta_3$  se utiliza (1.20) y multiplicamos a ambos lados por  $({}^0T_1)^{-1}$ , de esta manera se obtiene (1.22).

$$({}^0T_1)^{-1} \begin{bmatrix} R_{\alpha, \beta, \gamma} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = {}^1T_5 {}^5T_H \quad (1.22)$$

Igualando los elementos de matriz (1,4) y (2,4) de (1.22) se obtiene (1.23) para  $\theta_3$ .

$$\theta_3 = \text{acos}((k_1^2 + k_2^2 - a_2^2 - a_3^2)/(2a_2^2 a_3^2)) \quad (1.23)$$

donde

$$\begin{aligned} k_1 &= xc_1 + ys_1 - a_1 - d_5c_{234} \\ k_2 &= z - d_1 - d_5s_{234} \end{aligned} \quad (1.24)$$

donde  $a_i$  y  $d_i$  corresponde a los parámetros D-H correspondientes a la junta  $i$ , ver tabla 1.1.

Para calcular  $\theta_2$  se vuelve a los elementos de matriz (1,4) y (2,4) de (1.22) y se escriben como se muestra en (1.25).

$$\begin{aligned} k_1 &= a_3c_{23} + a_2c_2 = c_2B + s_2(-A) \\ k_2 &= a_3s_{23} + a_2s_2 = c_2B + s_2A \end{aligned} \quad (1.25)$$

donde  $A = a_2 + a_3c_3$  y  $B = a_3s_3$ . De estas ecuaciones se deduce que:

$$\begin{aligned} c_2 &= (Ak_1 + Bk_2)/(A^2 + B^2) \\ s_2 &= (Ak_2 - Bk_1)/(A^2 + B^2) \end{aligned} \quad (1.26)$$

Por lo que se puede hallar  $\theta_2$ :

$$\theta_2 = \text{atan2}(s_2, c_2) \quad (1.27)$$

El ángulo  $\theta_4$  se calcula utilizando (1.19), (1.23) y (1.27).

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (1.28)$$

Por último, a partir de (1.19) se tiene que  $\theta_5$  es directamente  $\gamma$ .

En general, los manipuladores pueden tener más de una forma de lograr una posición y orientación del efector final, es decir, las soluciones que se encuentran para la resolución de la cinemática inversa pueden no ser únicas [5], [8], [7]. En particular, el Scorbot-ER V Plus puede poseer una, dos o ninguna solución para la cinemática inversa. El caso donde no existe solución es cuando, la posición del efector final cae fuera del espacio de trabajo alcanzable por el manipulador. Entonces, para posiciones del efector final dentro del espacio de trabajo podemos tener una o dos soluciones, estas se denominan codo arriba ( $\theta_3 < 0$ ) y codo abajo ( $\theta_3 > 0$ ). El caso donde la solución es única se da cuando las configuraciones codo abajo y codo arriba coinciden.

La solución hallada para la cinemática inversa corresponde a la configura-

ción *codo arriba*. Para obtener la solución *codo abajo* es necesario cambiar el signo de  $\theta_3$  en (1.23), y repetir los cálculos. Esta configuración está descrita por (1.29). El resto de los ángulos quedan igual.

$$\begin{aligned}\theta_3 &= -\text{acos}((k_1^2 + k_2^2 - a_2^2 - a_3^2)/(2a_2^2a_3^2)) \\ \theta_2 &= -\text{atan2}(s_2, c_2)\end{aligned}\tag{1.29}$$

## 1.3. Implementación

Para implementar el modelo cinemático directo e inverso descrito en la sección anterior se utilizó Matlab. Para corroborar la validez del método se utilizaron primero simulaciones computacionales y luego se realizaron experimentos con un manipulador real Scorbot-ER V Plus.

### 1.3.1. Simulaciones

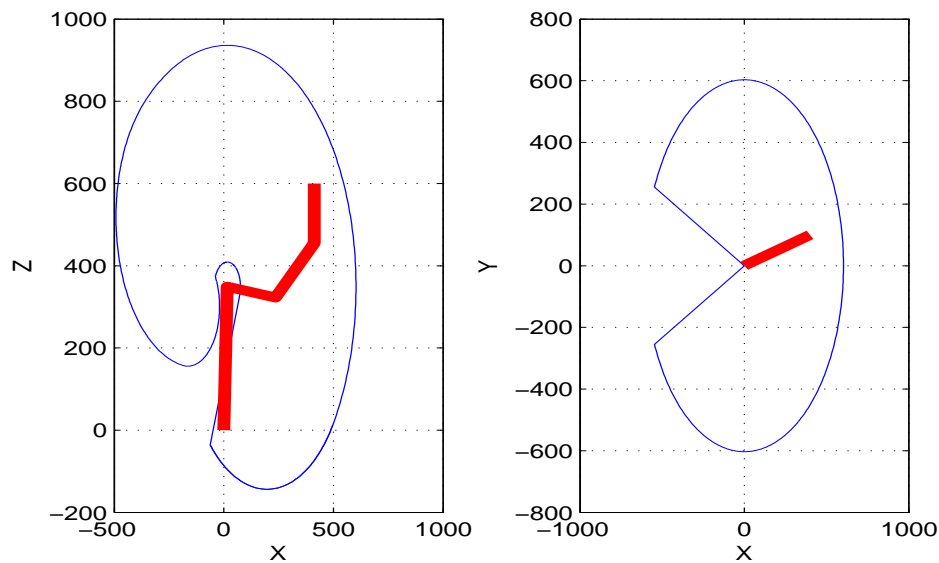
Se simuló la cinemática inversa y directa en Matlab y se estudiaron los casos de error de las mismas. Se utilizaron dos tipos de visualizadores diferentes para realizar los experimentos, el primero de ellos corresponde al visualizador que viene incluido en el *toolbox* de robótica de Peter Corke [10]. El segundo visualizador se desarrolló para este trabajo con mayor versatilidad y permite, entre otras funcionalidades, observar al manipulador dentro del espacio de trabajo del mismo.

La forma de validar el modelo cinemático directo consistió en darle al programa desarrollado las coordenadas de cada junta y verificar que la posición y orientación del efector final en los graficadores coincidan con las calculadas para dicha coordenadas de las juntas. De esta manera se probó que el modelo y las simulaciones eran consistentes entre sí.

Para verificar el modelo cinemático inverso, se utilizó la cinemática directa. Dadas una posición y orientación del efector final deseadas, se utilizó la cinemática inversa para obtener las coordenadas angulares de cada junta, y a continuación, se usó la cinemática directa para calcular la coordenadas finales del efector. De esta manera si las coordenadas deseadas coinciden con

las coordenadas finales, el modelo cinemático inverso es consistente con el modelo cinemático directo.

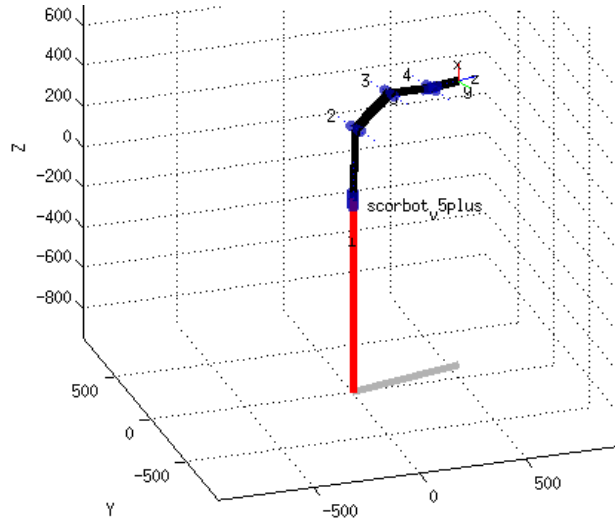
Para probar las simulaciones se utilizaron dos visualizadores. En la Fig. 1.4 se muestran los gráficos obtenidos, cuando se utiliza el visualizador desarrollado en Matlab. Esta función recibe como parámetros los ángulos de cada junta y devuelve dos gráficos que corresponden a la configuración del robot para esos parámetros. En particular muestra el manipulador desde dos perspectivas diferentes, la primera figura muestra el robot desde la vista lateral y la segunda presenta una vista superior del mismo. Las coordenadas del efector final en la Fig. 1.4 son  $x = 400\text{cm}$ ,  $y = 100\text{cm}$ ,  $z = 600\text{cm}$ ,  $\beta = 90^\circ$  y  $\gamma = 0^\circ$  y se grafica la solución *codo abajo*. Para calcular el espacio de trabajo se utilizaron los rangos máximos de cada articulación.



**Figura 1.4:** Espacio de trabajo del robot calculado a partir de la CD usando los rangos máximos de las articulaciones. Las unidades están en mm, el espacio de trabajo se puede calcular con los valores que se muestran en la tabla 1.1.

En la Fig. 1.5 se muestra el gráfico obtenido cuando se utiliza el visualizador que viene incluido en el *toolbox* de Peter Corke. Las coordenadas del efector final en la Fig. 1.4 son  $x = 600\text{cm}$ ,  $y = 100\text{cm}$ ,  $z = 600\text{cm}$ ,  $\beta = 0^\circ$  y

$\gamma = 0^\circ$  y se grafica la solución *codo arriba*.



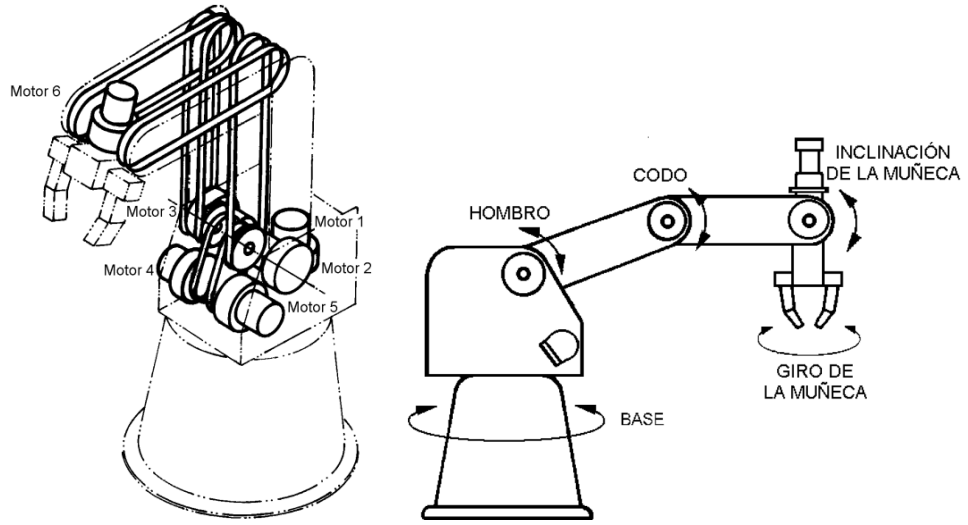
**Figura 1.5:** Simulación del manipulador Scorbot-ER V Plus utilizando el visualizador de Peter Corke.

### 1.3.2. Manipulador Real

La controladora del Scorbot-ER V Plus se conecta a una computadora utilizando un puerto serie RS232. Como las computadoras actuales no suelen poseer este puerto de fábrica, se utilizó un adaptador serial-USB que facilita la comunicación con el robot. Una ventaja importante de este sistema es que permite controlar el robot utilizando el software Matlab, desligándose del antiguo programa de fábrica ATS (Advance Terminal Software) y de este modo, se aumenta considerablemente la versatilidad del sistema robótico.

El Scorbot-ER V Plus es un robot desacoplado. En este tipo de robots, un movimiento en el hombro (eje 2, asociado a  $\theta_2$ ) no cambia el ángulo del codo (eje 3, asociado a  $\theta_3$ ) y de la muñeca (eje 4, asociado a  $\theta_4$ ) respecto de la trama base. En la figura 1.6 se observa el sistema de transmisiones del Scorbot-ER V Plus.

Nuestro modelo supone un robot acoplado, es decir, un movimiento en el eje 2 cambia el ángulo del eje 3 respecto de la trama base, pero deja constante



**Figura 1.6:** Transmisiones (a la izquierda) y juntas (a la derecha) del manipulador Scorbot-ER V Plus. Figuras extraída de [11].

el ángulo entre el eje 2 y el eje 3.

Por lo tanto usamos el sistema de ecuaciones (1.30) para poder trabajar con el robot Scorbot-ER V Plus como si fuera un robot acoplado.

$$\begin{aligned}
 ce_1 &= cg_1\theta_1 \\
 ce_2 &= cg_2\theta_2 \\
 ce_3 &= cg_3(-\theta_2 - \theta_3) \\
 ce_4 &= cg_4(-\theta_2 - \theta_3 - \theta_4 + \theta_5) \\
 ce_5 &= cg_5(\theta_2 + \theta_3 + \theta_4 + \theta_5)
 \end{aligned} \tag{1.30}$$

donde  $ce_i$  son las cuentas de *encoder* del motor  $i$ ,  $cg_i$  son constantes que dictan a cuantas cuentas de *encoder* equivale un grado, estas constantes son diferentes para cada motor, ya que dependen de la granularidad del *encoder*.

Otra particularidad del Scorbot-ER V Plus es que los motores 4 y 5 no están desacoplados, es decir, para mover la muñeca del brazo tenemos que mover el motor 4 y el motor 5 en diferentes sentidos. Ocurre lo mismo cuando queremos realizar un movimiento en *roll* ( $\theta_5$ ), tenemos que mover los motores 4 y 5 en el mismo sentido. Esta particularidad se puede observar en la ecuación (1.30), donde  $ce_4$  también depende de  $\theta_5$ , sin embargo  $\theta_4$  y  $\theta_5$



tienen signo contrario. De la misma forma  $ce_5$  depende de  $\theta_4$  y  $\theta_5$  pero con el mismo signo.

Para obtener las constantes  $cg_i$  se realizó una calibración de cada motor. Para ello se desplazó por una serie de ángulos conocidos midiendo en cada uno de estos como variaban las cuentas de *encoder*. Los valores obtenidos se compararon con los valores almacenados en la memoria de la controladora del robot. En la Tabla (1.2) se observa una comparación entre los resultados obtenidos para la calibración realizada y los valores de fábrica. La incerteza en la medición viene dada por el error que se comete al medir el ángulo relativo entre dos posiciones, que es menor a 2 grados. De esta manera a partir de la incerteza en la medición angular y suponiendo que no se pierden cuentas de *encoder*, se propagan errores para calcular la incertidumbre en las constantes de calibración.

motor	$cg_i$ Fábrica	$cg_i$ Calibración
1	141.88	141±3
2	113.51	111±3
3	113.51	111±3
4	27.9	29.0±0.5
5	27.9	29.0±0.5

**Tabla 1.2:** Constantes que relacionan a cuantas cuentas de encoder equivale un grado, obtenidas luego de la calibración.

La posición *home* de un manipulador es la posición donde las cuentas de encoder de todos los motores se anulan. Los manipuladores Scorbot poseen 3 *switches* en cada junta, 2 de ellos le permiten conocer los  $\theta_{i_{max}}$   $\theta_{i_{min}}$ , mientras que el tercer *switch* marca la posición *home* de cada articulación. Es decir, en la posición *home* del robot todas las articulaciones se encuentran en la posición del tercer *switch*. Esta posición *home* del robot no coincide con la posición *home* de nuestro modelo (ver Figura 1.1). Entonces se midió la diferencia entre ambas posiciones para contemplarla en el modelo. Los resultados obtenidos se comparan con los valores guardados en la memoria de la controladora. Estos resultados se muestran en la Tabla (1.3). El error viene

dado por la imprecisión del operario de colocar al robot en la posición *home* del modelo.

motor	Fábrica	Medido
1	0	$0\pm 282$
2	13653	$13515\pm 222$
3	-2867	$-2630\pm 222$
4	1773	$1794\pm 58$
5	-1773	$-2011\pm 58$

**Tabla 1.3:** Diferencias entre las posiciones home del robot y las de nuestro modelo, en cuentas de encoder.

## 1.4. Experimentos y Resultados

### 1.4.1. Manipulador Real

Para validar el modelo cinemático propuesto con el manipulador Scrobot-ER V Plus se realizaron dos experimentos. En primer lugar, se estudió la precisión y repetitividad del robot utilizando la implementación del modelo cinemático propuesto en este trabajo. En segundo lugar, se comparó la precisión obtenida con la del sistema de control del Scrobot-ER V Plus provista por el fabricante.

Para medir los movimientos del robot se utilizó un sistema de localización externa basado en visión denominado Whycon [12]. Este sistema es capaz de calcular en tiempo real la posición 3D de un patrón (una circunferencia con un diámetro determinado) en una secuencia de imágenes con una precisión de un milímetro. De esta forma, colocando los patrones en el manipulador podemos ordenar al robot que siga una trayectoria dada y conocer la posición del efector final, y la de cada una de las articulaciones, en todo momento. En la Fig. 1.7 se puede ver como se realizó el montaje de varios patrones y de que manera el sistema de localización visual las detecta.



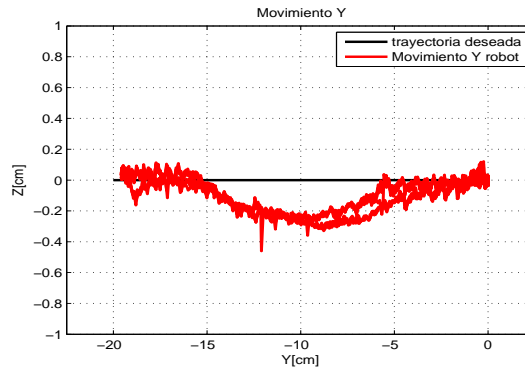
**Figura 1.7:** Sistema de localización externa calculando la posición 3D de las tres circunferencias de la imagen.

La forma de mover el robot a través de una trayectoria deseada utilizando el modelo cinemático desarrollado, consistió en elegir una secuencia de coordenadas en el mundo  $(x_j, y_j, z_j, \beta_j, \gamma_j)$  que definían la trayectoria. Posteriormente, se utilizó el modelo de cinemática inversa para pasar estos puntos a coordenadas para cada junta  $(\theta_{i_j})$ . Una vez conocidos los ángulos  $(\theta_{i_j})$ , se obtuvieron los valores de las cuentas de encoder  $(ce_{i_j})$  correspondiente para cada junta (utilizando las ecuaciones (1.30)). Por último se ordenó al robot que siga este conjunto de coordenadas.

Para mover el manipulador utilizando el modelo cinemático provisto de fábrica, se pasaron las coordenadas del mundo  $(x_j, y_j, z_j, \beta_j, \gamma_j)$  al controlador del robot para que recorriera esos puntos. Si la distancia entre dos puntos consecutivos de la trayectoria es grande, el desplazamiento del robot entre ellos puede no ser recto. Para evitar esto, se utilizó una separación entre puntos de 1 cm.

En primera instancia se le asignó una trayectoria deseada al manipulador y este la repitió cinco veces. De esta manera, podemos observar la precisión del control del robot cuando se le asigna una trayectoria, y al mismo tiempo podemos ver la capacidad del robot de repetir la misma trayectoria. La trayectoria deseada y la recorrida por el robot se muestran en la Fig. 1.8.

Como se observa en la Fig. 1.8, el robot fue capaz de seguir la trayec-



**Figura 1.8:** Movimiento en la dirección Y del robot. Se utilizó un sistema de localización externa para obtener la posición del efector final.

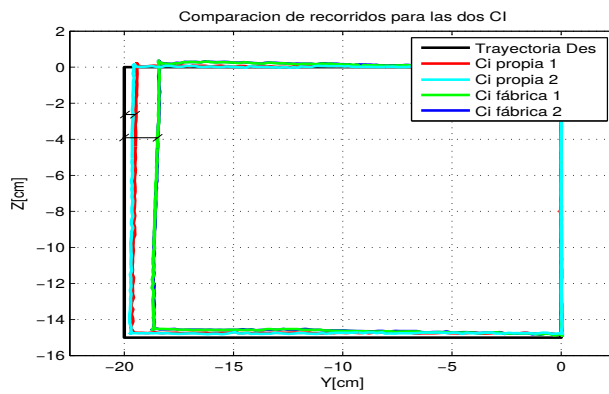
toría deseada a lo largo de todas las repeticiones, con un error sistemático máximo de 3 mm en la dirección perpendicular al movimiento. De esta forma evaluamos la repetibilidad del sistema, que se comportó de manera estable.

En segundo lugar se comparó la precisión del modelo cinemático desarrollado en este trabajo con el modelo provisto de fábrica. Para esto se recorrió una trayectoria cuadrada definida en el plano  $xz$  utilizando ambos modelos y se compararon los resultados obtenidos. La trayectoria fue elegida de manera que el manipulador tenga que usar todos los motores que posee a lo largo del movimiento, ya que se buscó que el error cometido por el robot sea el máximo posible.

En la Fig. 1.9 se compara la trayectoria deseada (negro) con las trayectorias realizadas por el robot utilizando el modelo desarrollado (en rojo y cian) y la cinemática provista de fábrica (en verde y azul), para dos repeticiones en cada caso.

Se definió el error de seguimiento como el valor medio de la distancia entre la trayectoria deseada y la real en la dirección perpendicular a la trayectoria deseada (ver Fig. 1.9). Los valores obtenidos fueron:  $0,17 \pm 0,02$  cm para nuestro método y  $0,47 \pm 0,01$  cm para el método de fábrica.

De esta forma podemos afirmar que el modelo cinemático propuesto supera al provisto por el fabricante. Por último, en los dos casos se puede



**Figura 1.9:** Comparación entre las trayectorias realizadas por el robot para los dos modelos cinemáticos en cuestión.

observar un error sistemático. Este tipo de errores puede atribuirse a la diferentes calibraciones de los modelos, es decir, diferencias en las constantes que se muestran en las Tablas 1.2 y 1.3. Para mejorar el control del robot, se plantea abordar el problema de los errores sistemáticos como trabajo futuro.



## Capítulo 2

# Generación de Trayectorias

En este Capítulo se muestran diferentes métodos que permiten generar trayectorias o rutas que puedan utilizar los manipuladores. Este problema se conoce como Generación de Trayectorias, involucra encontrar la trayectoria deseada pero también involucra evasión de obstáculos, prevenir la saturación del actuador, entre otras cosas.

En este capítulo no se va a cubrir el problema más general, si no el problema más específico de seguir una secuencia de puntos ya dados que componen la trayectoria. En otras palabras nos interesa saber como hacer para que el efector final del manipulador siga una secuencia de puntos deseados (posiciones y orientaciones del efector final). Por ejemplo, el usuario podría especificar nada más la posición y orientación de destino deseadas del efector final y dejar que el sistema decida la forma exacta de la ruta para llegar ahí, la duración, el perfil de velocidades y otros detalles.

El problema básico es mover el manipulador desde su posición actual,  $X_{inicial}$ , hasta una posición final deseada,  $X_{final}$ . Algunas veces es necesario especificar el movimiento con mucho más detalles, como puede ser incluyendo una secuencia de *puntos vías* deseados, (puntos intermedios entre la posición final e inicial), o especificando la forma de la ruta. Generalmente es conveniente que el movimiento del manipulador sea uniforme, es decir, continuo y su derivada primera continua. Los movimientos bruscos y torpes tienden a producir un mayor desgaste en el mecanismo y producen vibraciones al

excitar las resonancias del manipulador.

Para abordar el problema de la generación de trayectorias podemos generarlas en el espacio de las articulaciones o en el espacio Cartesiano. Para éste último se requiere conocer el modelo cinemático del manipulador, que estudiamos en el capítulo anterior. En la Sección 2.1 analizamos la generación de trayectorias en el espacio de las articulaciones y en la Sección 2.2 en el espacio Cartesiano. Existen una gran variedad de formas que pueden especificarse y planearse rutas. En este Capítulo se muestran algunas elecciones simples para las funciones que determinan la ruta [13],[14],[15].

### 2.1. Esquemas en el Espacio de las Articulaciones

En esta sección se consideran métodos en los que las formas de las trayectorias generadas (en espacio y tiempo) se describen en términos de funciones de ángulos de articulación, en nuestra notación en términos de  $\theta_i$ . Cada punto de la ruta se especifica generalmente en términos de una posición y una orientación deseadas del efector final en la trama base. Cada uno de estos puntos se “convierte” en un conjunto de ángulos de articulación deseados mediante la cinemática inversa. Después se encuentra una función uniforme para cada una de las  $n$  articulaciones que pasa a través de los puntos vía y que termina en el punto de destino. Estas funciones son distintas e independientes en cada articulación. El tiempo requerido por segmento es el mismo para cada articulación, de manera que todas las articulaciones llegarán al punto vía al mismo tiempo.

Los esquemas en el espacio de la articulación logran alcanzar la posición y orientación deseadas en los puntos vía. Sin embargo, el trayecto entre los puntos vía, la forma de la ruta, aunque es considerablemente simple en el espacio de articulación, es compleja si se la describe en el espacio cartesiano. Los esquemas en el espacio de articulación son generalmente más sencillos de calcular y, como no hay correspondencia continua entre el espacio de articulación y el espacio cartesiano, esencialmente no hay ningún problema



con las singularidades del mecanismo. Las singularidades son puntos donde el manipulador pierde uno o más grados de libertad, se explican con más detalle en la Sección 2.2.

### 2.1.1. Polinomios Cúbicos

El problema que queremos resolver es mover la herramienta desde su posición inicial a una posición de destino en cierta cantidad de tiempo. La cinemática inversa permite calcular el conjunto de ángulos de articulación que corresponden a la posición y orientación de destino. La posición inicial también se conoce como un conjunto de ángulos de articulación. Lo que se requiere es una función para cada articulación, cuyo valor en  $t_0$  sea la posición inicial y cuyo valor en  $t_f$  sea la posición deseada de esa articulación. Para crear un solo movimiento uniforme necesitamos por lo menos cuatro condiciones de contorno, dos sobre la posición y dos sobre la velocidad.

$$\begin{aligned}\theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \\ \dot{\theta}(0) &= \dot{\theta}_i \\ \dot{\theta}(t_f) &= \dot{\theta}_f\end{aligned}\tag{2.1}$$

El caso más simple a considerar es si la trayectoria deseada no contiene puntos vía, por lo que  $\dot{\theta}_i(0) = \dot{\theta}_f(t_f) = 0$ . Sin embargo, la solución de este problema es un caso particular del caso general, es decir, la inclusión de puntos vía.

En general deseamos permitir la especificación de rutas que contengan puntos vía intermedios. Cada punto vía se especifica en términos de una posición y una orientación deseadas de la trama herramienta relativa a la trama estación. Usando la cinemática inversa “convertimos” cada punto vía a un conjunto de ángulos de articulación. Después consideramos el problema de calcular ecuaciones cúbicas que conecten entre sí los valores de los puntos vía para cada articulación de una manera uniforme.

Si se conocen las velocidades deseadas en los puntos vía, tenemos que las condiciones de contorno para la velocidad ahora son  $\dot{\theta}(0) = \dot{\theta}_0$  y  $\dot{\theta}_i(t_f) = \dot{\theta}_f$ .

Las cuatro ecuaciones que describen este polinomio cúbico general son:

$$\begin{aligned}\theta_0 &= a_0, \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \\ \dot{\theta}_0 &= a_1, \\ \dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2.\end{aligned}\tag{2.2}$$

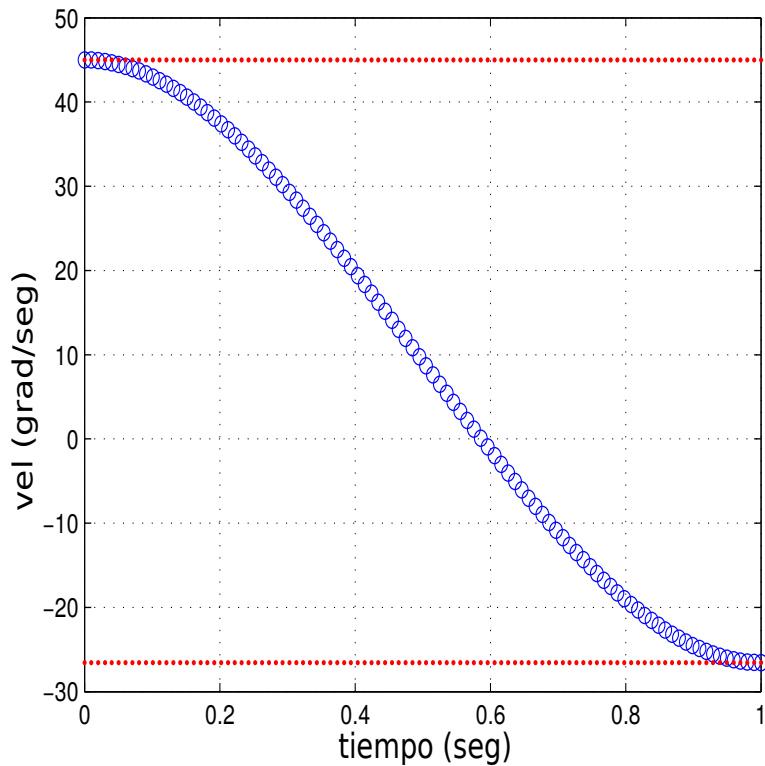
Al resolver estas ecuaciones para las  $a_i$  obtenemos

$$\begin{aligned}a_0 &= \theta_0, \\ a_1 &= \dot{\theta}_0, \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f, \\ a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f - \dot{\theta}_0).\end{aligned}\tag{2.3}$$

Utilizando las ecuaciones 2.3 podemos calcular el polinomio cúbico que conecta las posiciones iniciales y finales con velocidades iniciales y finales dadas.

Existen varias formas en las que podría especificarse la velocidad deseada en los puntos vía. Se podría pedir que el usuario la especifique, sin embargo, sería una carga requerir que el usuario siempre haga estas especificaciones. Otra opción es que el sistema seleccione automáticamente las velocidades en los puntos vía, mediante una heurística adecuada, o pidiendo por ejemplo que la aceleración sea constante en los puntos vía. Una heurística simple en términos conceptuales y computacionales consiste en imaginar los puntos vía conectados con segmentos de líneas rectas. Si la pendiente de estas líneas cambia de signo en el punto vía elegimos velocidad cero, si la pendiente de estas líneas no cambia de signo elegimos un promedio de las dos pendientes como la velocidad del punto vía.

En la figura 2.1 se muestra la trayectoria obtenida con un planificador cúbico para un grado de libertad.

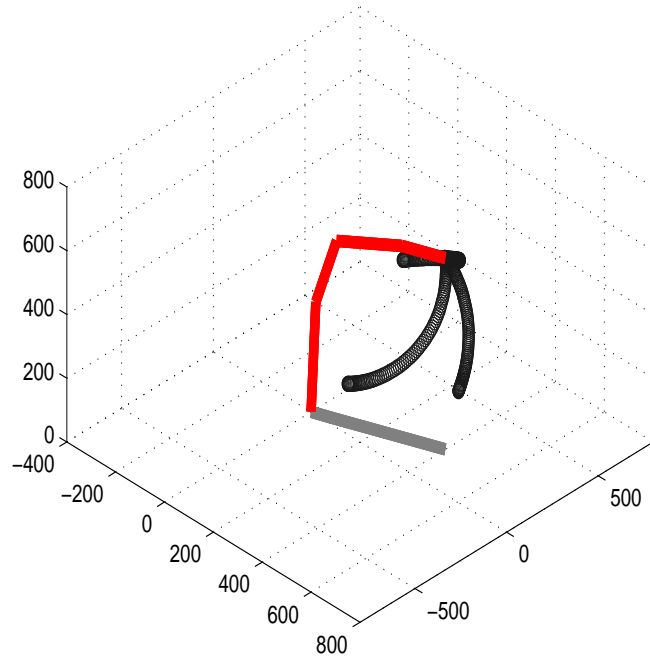


**Figura 2.1:** Trayectoria seguida por una articulación para un generador cúbico.

En la figura 2.2 se muestra trayectorias cúbicas, en el espacio cartesiano, para un manipulador de 5 grados de libertad.

Los problemas más comunes de trabajar con polinomios además de no lograr rutas deseadas en el espacio cartesiano, son el *overshoot* (superar la posición deseada), y pasar mucho tiempo a velocidades menores a la máxima. Por lo que no son métodos muy eficientes.

Existen otros métodos de generación de trayectorias en el espacio de articulación. Uno de ellos consiste en utilizar polinomios de orden mayor, por ejemplo, si deseamos especificar la posición, velocidad y aceleración al inicio y final de cada segmento se requiere un polinomio de quinto grado. Otro método consiste en interpolar con funciones lineales la posición inicial y final, pero agregando una región de “mezcla” parabólica en cada punto de la ruta de manera que la posición y velocidad sean continuas en toda la ruta. Sin



**Figura 2.2:** Generador de trayectorias cúbico para un manipulador de 5 grados de libertad, desde distintas posiciones iniciales.

embargo, aún en este esquema donde el movimiento de cada articulación es lineal el efector final no se mueve en general en una línea recta en el espacio.

## 2.2. Esquemas en el Espacio Cartesiano

Como se mencionó en la sección anterior las rutas que se calculan en el espacio de articulación pueden asegurar que se llegue a los puntos vía y los puntos de destino, incluso cuando estos puntos de ruta se hayan especificado mediante tramas cartesianas. Sin embargo, la forma espacial de la ruta tomada por el efector final no es una línea recta a través del espacio, si no una forma complicada que depende de un comportamiento cinemático específico del manipulador que se este utilizando. En esta sección se considera un método de generación de rutas en el cual las formas de la ruta se describen en

términos de funciones que calculan la posición y orientación cartesianas como funciones del tiempo; de esta manera se puede especificar también la forma espacial de la ruta entre los puntos de la ruta. En aplicaciones industriales la forma más común de una ruta es en línea recta, pero pueden usarse también rutas de forma circular, sinusoidal u otro tipo.

En particular vamos a pensar en rutas simples como pueden considerarse rectas, circunferencias o sinusoidales. Estas rutas son simples debido a que el usuario sólo tiene que especificar el tipo de ruta y un par de parámetros, por ejemplo en el caso de una recta, debería especificar que es una recta y dar dos puntos en el espacio de trabajo. Cualquier ruta más compleja generalmente se puede pensar como combinación de las anteriores. De esta forma, contamos con una trayectoria en el espacio cartesiano que es la ruta que queremos que siga el efector final del manipulador, se llamará ruta deseada.

Para describir el método se utilizaran ecuaciones halladas en el capítulo anterior. Recordando 1.10 y 1.19 se puede escribir 2.4.

$$\begin{aligned}
 x &= c_1(a_1 + a_2c_2 + a_3c_{23} + d_5c_{234}) \\
 y &= s_1(a_1 + a_3c_{23} + a_2c_2 + d_5c_{234}) \\
 z &= d_1 + a_3s_{23} + a_2s_2 + d_5s_{234} \\
 \alpha &= \theta_1 \\
 \beta &= -(\theta_2 + \theta_3 + \theta_4) = -\theta_{234} \\
 \gamma &= \theta_5
 \end{aligned} \tag{2.4}$$

Como se explicó en el capítulo anterior este sistema de ecuaciones tiene 6 ecuaciones pero solo 5 incógnitas, debido a que en nuestro manipulador  $\alpha$  no es un grado de libertad. El valor de  $\alpha$  queda determinado con el resto de las ecuaciones. Podemos reescribir 2.4 de forma compacta, para ello definimos  $X = [x, y, z, \beta, \gamma]$  como el vector de 5 coordenadas en el espacio cartesiano y  $\Theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$  al vector de 5 coordenadas en el espacio de las articulaciones. De esta forma reescribiendo 2.4 se obtiene 2.5.

$$X = h(\Theta) \tag{2.5}$$

El jacobiano ( $J(\Theta)$ ) es una matriz de 5x5 de derivadas parciales y queda

definido en 2.6.

$$\delta X = J(\Theta)\delta\Theta \quad (2.6)$$

Al dividir ambos lados de la ecuación 2.6 por el elemento de tiempo diferencial, se obtiene 2.7. De esta manera, se piensa al jacobiano como una asignación o mapeo de las velocidades angulares  $\dot{\Theta}$  a las velocidades cartesianas  $\dot{X}$ .

$$\dot{X} = J(\theta)\dot{\Theta} \quad (2.7)$$

Resulta importante verificar si el jacobiano es inversible o singular. Si la matriz es no singular, entonces se puede invertir para calcular las velocidades de articulación a partir de las velocidades cartesianas, usando 2.8.

$$\dot{\Theta} = J^{-1}(\theta)\dot{X} \quad (2.8)$$

La mayoría de los manipuladores tienen valores de  $\Theta$  para los que el jacobiano se vuelve singular. Dichas ubicaciones se llaman singularidades del mecanismo, o simplemente singularidades y según la ecuación 2.8, la velocidad angular de las articulaciones se vuelve infinita. Todos los manipuladores poseen singularidades en los límites o fronteras de su espacio de trabajo, y la mayoría tiene regiones de singularidades dentro de su espacio de trabajo. Cuando el manipulador se encuentra en una configuración singular, ha perdido uno o más grados de libertad (viéndolo en el espacio cartesiano). Esto significa que hay cierta dirección (o subespacio) en el espacio cartesiano sobre la cual es imposible mover el robot. Es obvio que esto ocurre en la frontera del espacio de trabajo. Un estudio minucioso de las singularidades está más allá del alcance de este trabajo; para más información ver [16].

Resumiendo queremos seguir una trayectoria conocida dependiente del tiempo en el espacio cartesiano. Esta ruta puede ser especificada por el usuario como una función en 3 dimensiones dependiente del tiempo.

El usuario generalmente especifica el tipo de ruta que quiere recorrer, la posición final y/o alguna posición intermedia en caso de ser necesaria, y

por último puede especificar el tiempo de ejecución total o en caso de no especificarlo dejar que el manipulador calcule cual es ese tiempo.

Los esquemas cartesianos requieren un mayor poder de cómputo para ejecutarse, ya que en tiempo de ejecución tiene que resolver el jacobiano inverso.

### 2.2.1. Control para el Seguimiento de una Trayectoria en el Espacio Cartesiano

Suponemos conocidos la posición inicial y final, el tiempo de ejecución y la trayectoria deseada  $X_d$  en el espacio cartesiano ( $X_d$  son trayectorias simples como rectas o circunferencias). Se mide la posición angular  $\Theta$  y utilizando la cinemática directa se estima la posición cartesiana  $X(t)$  del robot. La velocidad del robot se calcula de forma proporcional al error cartesiano ( $\dot{X}(t) = k_p(X - X_d)$ ) y utilizando la ecuación 2.6 se calculan las velocidades angulares de los motores. Resumimos este método en el algoritmo 1.

---

**Algorithm 1:** Control de trayectorias

---

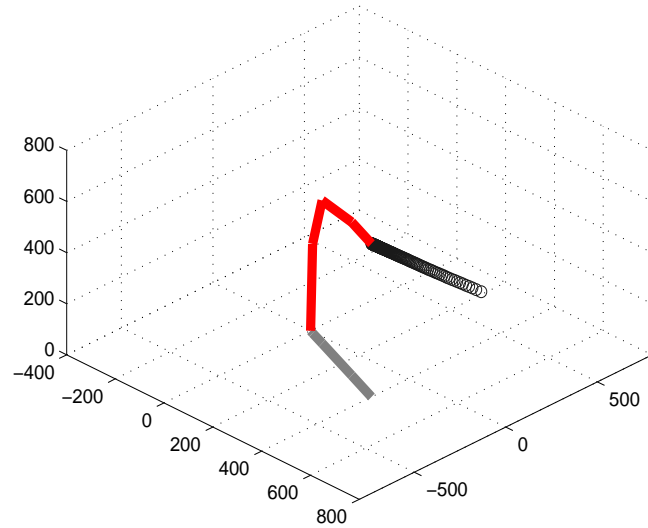
```

Input:  $X_d(t)$ : posición deseada
Output:  $\dot{\Theta}(t)$ : velocidad del efector final
 $Q \leftarrow X_{inicial}$       /* Encuentro posición inicial del robot */

for el tiempo de ejecución do
     $e(t) \leftarrow X(t) - X_d(t)$           /* error cartesiano */
     $J \leftarrow J(\Theta_i(t))$           /* calculo  $J$  con las coord actuales */
    if  $J$  no singular then
         $\dot{X}(t) \leftarrow k_p e(t)$           /* aplico el control */
         $\dot{\Theta}(t) \leftarrow J^{-1} \dot{X}(t)$       /* paso a coord angulares */
         $\Theta(t + \Delta t) = \Theta(t) + \dot{\Theta} dt$  /* uso Euler para evolucionar  $\Theta$  */
        */
    else
         $\perp$  aviso que pase por una singularidad
     $X \leftarrow \Theta$           /* uso la cinem directa para pasar a coord
    cartesianas */

```

---



**Figura 2.3:** desplazamiento del efector final del robot en línea recta utilizando el método del jacobiano. La figura está en cm.

En 2.3 se muestra en 3D la trayectoria recorrida por el efector final del manipulador, desde una posición inicial a una posición final. Se observa que usando el método del jacobiano se logran rutas rectas en el espacio cartesiano.

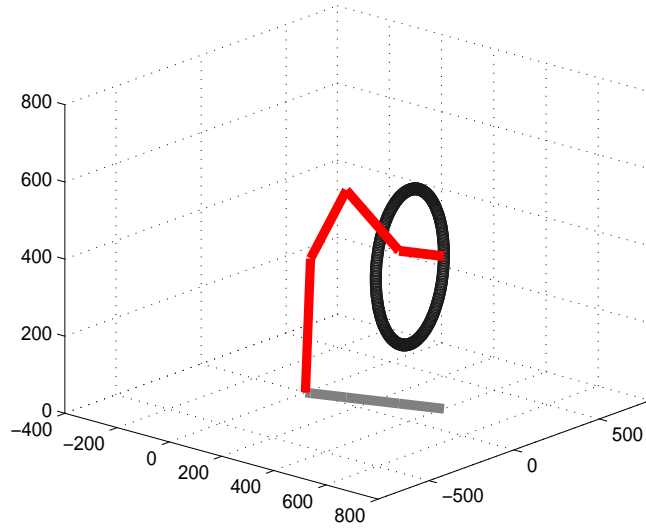
En la figura 2.4 se muestra una ruta circular recorrida por el manipulador utilizando el método del jacobiano.

En la figura 2.5 se muestran cada una de las coordenadas cartesianas, en rojo se observan las rutas deseadas de cada coordenada respectivamente. Se observa que en los ejes  $y$  y  $z$  son funciones senoidales desfasadas.

Generalmente, cuando se planean rutas utilizamos una aceleración pre-determinada o máxima en cada punto. En realidad, la aceleración que el manipulador es capaz de producir en cualquier instante es una función de la dinámica del brazo y de los límites del actuador.

Cuando planeamos una ruta suponiendo que hay una aceleración máxima en cada articulación o a lo largo de cada grado de libertad, estamos haciendo una gran simplificación. Para tener cuidado de no exceder las capacidades del dispositivo, esta aceleración máxima debe elegirse de manera conservadora.

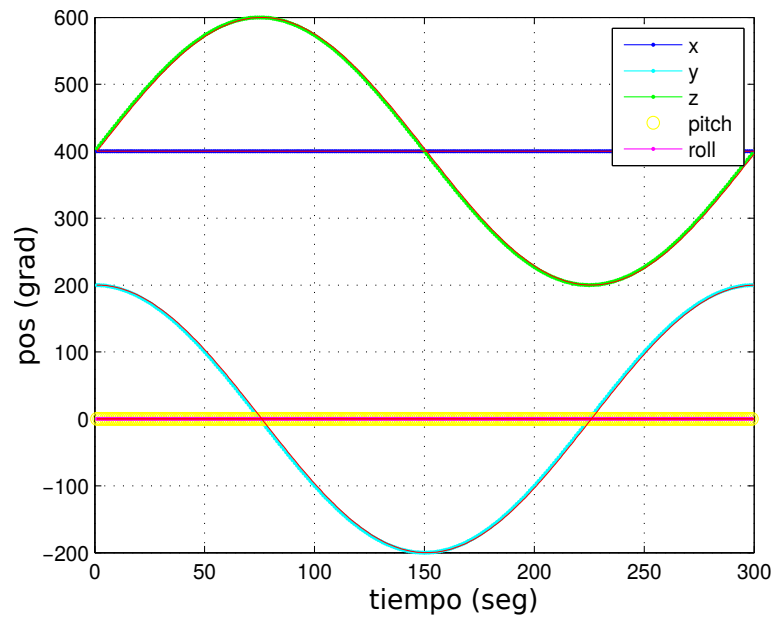




**Figura 2.4:** Desplazamiento en cada dirección cartesiana, para un movimiento circular del efector final. Los ejes están en cm.

Por lo tanto, no estamos utilizando por completo las capacidades de velocidad del manipulador en rutas planeadas por los métodos que presentamos.

Para poder usar al máximo las capacidades de velocidad del manipulador necesitamos conocer el modelo dinámico de cada uno de los motores que fuerzan cada articulación. En el capítulo siguiente abordaremos este tema.



**Figura 2.5:** Desplazamiento en cada dirección cartesiana, para el movimiento que se muestra en la figura 2.4.

## Capítulo 3

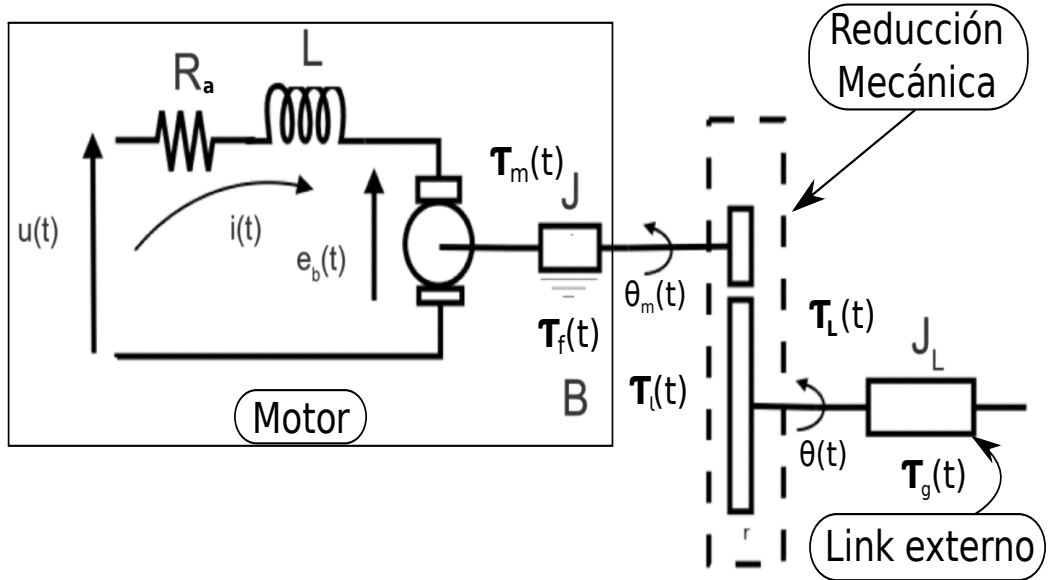
# Modelo Electro-Mecánico, Identificación y Simulación de una Articulación

La mayoría de manipuladores utilizan motores de corriente continua (DC) para mover sus articulaciones, debido a su estructura simple, un rendimiento de control muy bueno y bajo costo. Para construir un modelo dinámico completo de un manipulador se debe incluir no solo la dinámica de sus elementos (barras o eslabones) sino también la de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al modelo dinámico nuevas inercias, rozamientos, saturaciones de los circuitos electrónicos, entre otras cosas.

En la Sección 3.1 se presenta el modelo electro-mecánico de un motor de corriente continua. El mismo permite modelar la dinámica de los sistemas de transmisión, los actuadores y equipos electrónicos de mando, que junto con la dinámica de los eslabones del brazo robótico (que se presenta en el Capítulo 4) completa el modelo dinámico del manipulador. A continuación, en la Sección 3.2 se detalla un método que permite identificar las constantes del modelo electro-mecánico para un motor de corriente continua. Las simulaciones realizadas con el modelo se comparan con los resultados obtenidos mediante mediciones del motor real.

### 3.1. Modelo Electro-Mecánico de un Motor de Corriente continua (DC)

En la figura 3.1 se muestra el esquema eléctrico y mecánico de un motor de corriente continua, que puede representarse mediante el sistema de ecuaciones 3.1, [17], [18]. La primera ecuación del sistema describe los fenómenos eléctricos y nos permite obtener la tensión eléctrica ( $u(t)$ ) y la segunda ecuación describe los fenómenos mecánicos en un motor con reducción conectado a un cuerpo rígido externo (carga o *link*), relacionando el momento de inercia ( $J$ ) y la aceleración ( $\ddot{\theta}_m$ ) del motor con los torque que actúan en el sistema.



**Figura 3.1:** Esquema electro-mecánico de un motor de corriente continua.

$$\begin{aligned} u(t) &= L \frac{di(t)}{dt} + R_a i(t) + e_b(t) \\ J \ddot{\theta}_m(t) &= \tau_m(t) - \tau_l(t) - \tau_f(t) \end{aligned} \quad (3.1)$$

donde  $L, R_a$  y  $J$  representan la inductancia, la resistencia y el momento de inercia del motor que consideramos constantes. Además,  $\tau_m(t)$  es el torque del motor,  $i(t)$  es la corriente eléctrica,  $\theta_m(t)$  es la posición angular,  $e_b(t)$

### 3.1 Modelo Electro-Mecánico de un Motor de Corriente continua (DC)

fuerza electromotriz,  $\tau_l(t)$  es el torque de la carga visto desde el motor,  $\tau_f(t)$  es el torque provocado por la fricción.

Este tipo de motores satisface además ecuaciones de acople electromagnético que vienen dadas por 3.2.

$$\begin{aligned}\tau_m(t) &= k_m i(t) \\ e_b(t) &= k_b \dot{\theta}(t)\end{aligned}\quad (3.2)$$

donde  $k_m$  es la constante del torque o y  $k_b$  la constante de la fuerza contra-electromotriz.

Para estimar el toque de fricción consideramos el modelo clásico, que consta de 3 componentes: fricción de Coulomb ( $\tau_{fc}$ ), fricción viscosa ( $\tau_{fv}$ ) y fricción estática ( $\tau_{fs}$ ).

$$\tau_f = \tau_{fc}(t) + \tau_{fv}(t) + \tau_{fs}(t) \quad (3.3)$$

donde:

$$\begin{aligned}\tau_{fv}(t) &= B\dot{\theta}_m(t) \\ \tau_{fc}(t) &= \tau_c \operatorname{sgn}(\dot{\theta}_m(t)) \\ \tau_{fs}(t) &= \begin{cases} \tau_e(t), & \dot{\theta}_m(t) = \ddot{\theta}_m(t) = 0 \\ \tau_{fs}^{max} \operatorname{sig}(\tau_e(t)), & \dot{\theta}_m(t) = 0, \quad \ddot{\theta}_m(t) \neq 0 \end{cases}\end{aligned}\quad (3.4)$$

donde los parámetros  $B, \tau_c$  y  $\tau_e$  representan las constantes de fricción viscosa, de Coulomb y estática respectivamente.  $\tau_e(t)$  representa el par externo  $\tau_e(t) = \tau_m(t) - \tau_l(t) - J\ddot{\theta}_m(t)$ .  $\tau_e^{max}$  representa el máximo valor que puede tomar  $\tau_e(t)$  antes de que el motor comience a moverse.

El torque de fricción estática actúa cuando el motor está inmóvil o cuando hay un cambio de sentido del eje del motor (cuando la aceleración angular  $\ddot{\theta}_m(t) \neq 0$ ). Como vemos lo que hace el par de fricción estática  $\tau_{fs}$  en la situación de arranque del motor es impedir el movimiento del eje del motor hasta que el par externo supere un valor constante  $\tau_{fs}^{max}$ . La variación temporal se debe a la variación de la corriente  $i(t)$ , es decir que en el arranque  $i(t) \neq 0$ .

Utilizando las ecuaciones 3.2 se reescribe el sistema de ecuaciones 3.1 y se obtiene:

$$\begin{aligned} u(t) &= L \frac{di(t)}{dt} + R_a i(t) + k_b \dot{\theta}_m(t) \\ J \ddot{\theta}_m(t) &= k_m i(t) - B \dot{\theta}_m(t) - \tau_{fs}(t) - \tau_{fc}(t) - \tau_l(t) \end{aligned} \quad (3.5)$$

De esta forma quedan definidas completamente las ecuaciones que describen el modelo dinámico del motor de corriente continua.

### 3.1.1. Incorporación de Carga

Se considera que el eje del motor está unido a un cuerpo rígido (la carga del motor) por algún mecanismo reductor de reducción  $r \in (0, 1)$ . Entonces la coordenada angular  $\theta(t)$  y el torque de carga  $\tau_L(t)$  a la salida del reductor estarán relacionados con  $\theta_m(t)$  y  $\tau_l(t)$  en el eje del motor según 3.6 (ver figura 3.1).

$$\begin{aligned} \theta &= r \theta_m \\ \tau_L &= \frac{1}{r} \tau_l \end{aligned} \quad (3.6)$$

Además, consideramos que la carga es un cuerpo rígido cuya ecuación dinámica se puede escribir según 3.7.

$$\tau_L(t) = J_L \ddot{\theta}(t) + \tau_g(\theta) \quad (3.7)$$

donde  $J_L$  es el momento de inercia de la carga,  $\tau_g$  es el torque de gravedad. Se quieren escribir todas las fuerzas en el sistema correspondiente al eje del motor, entonces se escribe 3.7 como función de  $\theta_m(t)$  y  $\tau_l(t)$ .

$$\tau_l(t) = r^2 J_L \ddot{\theta}_m(t) + r \tau_g(\theta) \quad (3.8)$$

Introduciendo la expresión de  $\tau_l$  en la ecuación mecánica del motor dada por 3.5, y agrupando los términos según las derivadas de  $\theta_m$  se obtiene:

$$\begin{aligned} u(t) &= L \frac{di(t)}{dt} + R_a i(t) + k_b \dot{\theta}_m(t) \\ J_{eff} \ddot{\theta}_m(t) &= k_m i(t) - B \dot{\theta}_m(t) - \tau_{fs}(t) - \tau_{fc}(t) - r \tau_g(\theta, t) \end{aligned} \quad (3.9)$$

donde  $J_{eff} = J + r^2 J_L$  y se denomina el momento de inercia efectivo.

Además, suponemos que  $L \simeq 0$  [4]. De esta manera, despejando la corriente de la primera ecuación del sistema 3.9 y reemplazándola en la segunda ecuación se obtiene:

$$J_{eff}\ddot{\theta}_m(t) = \frac{k_m}{R_a}u(t) - B_{eff}\dot{\theta}_m(t) - \tau_{fs}(t) - \tau_{fc}(t) - r\tau_g(\theta, t) \quad (3.10)$$

donde  $B_{eff} = B + \frac{k_m k_b}{R_a}$ .

Nosotros trabajamos con  $\theta$  debido a que se mide el movimiento después de la reducción. Teniendo en cuenta esto, se reescribe la ecuación 3.10, utilizando la ecuación 3.6, de manera de cambiar de variables a  $\theta$ .

$$J_{eff}\ddot{\theta}(t) = \frac{rk_m}{R_a}u(t) - B_{eff}\dot{\theta}(t) - r\tau_{fs} - r\tau_{fc}(t) - r^2\tau_g(\theta, t) \quad (3.11)$$

## 3.2. Identificación del Motor

Los motores del robot Scorbob-ER V Plus se controlan por medio de un sistema de tipo caja cerrada que utiliza un *software* de control propietario por la empresa Eashed Robotec que fabrica el manipulador, que se encuentra obsoleto. Por este motivo se decidió estudiar el modelo dinámico de los motores utilizando el manipulador RB4 de la empresa nacional KitRobot que nos permitió tener completa libertad en el uso de los motores. Este manipulador utiliza servomotores (o servos) de bajo costo que fueron modificados para emular los motores de corriente continua que se utilizan en los robots Scorbob.

Los servomotores son un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango y mantenerse estable en esa posición. Básicamente un servo es un motor de corriente continua, al que se le ha agregado un sistema de control embebido, un potenciómetro y un conjunto de engranajes.

Se puede modificar el servo para obtener un motor de corriente continua que, si bien no tiene la capacidad de controlar del servo, conserva la fuerza,

### Cap.3 Modelo Electro-Mecánico, Identificación y Simulación de una Articulación

---

velocidad y baja inercia que caracteriza a estos dispositivos. Esta modificación consiste básicamente en extraer la placa de control del servomotor. De esta forma podemos controlar el motor de corriente continua, y utilizar el potenciómetro adherido al eje del motor para tener una medida absoluta de la posición del eje del motor.

En los motores de corriente continua generalmente se utilizan encoders para medir el movimiento del eje del motor, sumando de manera incremental los movimientos del eje, calculamos la posición del motor. Pero esta medida es relativa a la posición inicial (cuando se prende el encoder este se encuentra en la posición 0) del eje. Por lo que tener una medida absoluta de la posición del eje representa una gran ventaja, ya que podemos saber la posición del motor de manera muy simple.

De esta forma la posición absoluta del motor se obtiene midiendo el voltaje en el potenciómetro. Con el fin de reducir el ruido de medición para tomar un valor real de posición, se toman 10 mediciones y se promedian. A partir de la medición de la posición es necesario estimar la velocidad. La forma más simple de hacerlo es medir la posición en dos instantes consecutivos y de tiempo, y derivar según la siguiente fórmula  $\dot{\theta}_t = (\theta_t - \theta_{t-1})/\Delta t$ . Sin embargo, a pesar de promediar las mediciones de la posición este cálculo provoca que la velocidad obtenida sea muy ruidosa. Para eliminar el ruido de la señal se utiliza un filtro pasa-bajo que viene dado por la siguiente ecuación:

$$\dot{\theta}_t = (1 - \alpha)\dot{\theta}_{t-1} + \alpha \frac{\theta_t - \theta_{t-1}}{\Delta t} \quad \text{con} \quad \alpha = 0,2. \quad (3.12)$$

Para controlar el servo modificado es necesario un driver, que es capaz de trabajar con la potencia que necesita el motor, y una placa con un micro controlador, que permita controlar el driver y a su vez el motor. En nuestro caso, utilizamos como driver un L298 y un Arduino Uno. El motor se controla usando modulación de pulso (o *pwm* por sus siglas en inglés). El Arduino proporciona un *pwm* que varía desde 0 a 255, donde 0 significa 0V y 255 significa 5V. La frecuencia del *pwm* es de 8 KHz. Como la frecuencia del *pwm* es mucho más grande que los tiempos característicos mecánicos, es una buena aproximación asumir que el *pwm* es proporcional al voltaje aplicado



a los motores. A partir de ahora cuando se hable de *pwm* aplicado al motor, significa un voltaje proporcional al valor del *pwm* en cuestión aplicado al motor.

Para adquirir los datos del motor, como la posición, velocidad, *pwm* y tiempo, se conectó la tarjeta Arduino a una computadora y por comunicación serial se guardan los datos.

Debido a que los servomotores generalmente tienen otros usos, como puede ser modelismo de aviones, barcos, helicópteros o trenes, las características del motor no se especifican. Sin embargo, el uso masificado de estos motores hace que su precio sea muy bajo. Debido a esto, es necesario encontrar todas las constantes del modelo 3.11,  $(k_m, B_{eff}, J$  y  $\tau_c)$ . Se asume que la resistencia  $R$  y la reducción  $r$  se conocen o se pueden estimar. En particular, para este motor se tiene que  $R = 5\Omega$  y  $r = 0,0033$ . Al eje del motor se fijó una barra de aluminio rectangular rígida de masa  $m = 0,2Kg$  y largo  $l = 0,2m$ .

Para identificar los parámetro propios del motor se renombraron de manera conveniente las constantes del modelo, utilizando 4 constantes nuevas  $\alpha_i$  con  $i = 1, 2, 3, 4$ . Donde:

$$\alpha_1 = \frac{rk_m}{R_a B_{eff}} \quad \alpha_2 = \frac{r\tau_c}{B_{eff}} \quad \alpha_3 = \frac{r^2}{J_{eff}} \quad \alpha_4 = \frac{J_{eff}}{B_{eff}} \quad (3.13)$$

Reemplazando estas constantes en la ecuación del modelo 3.11 se obtiene:

$$\ddot{\theta}(t) = \frac{\alpha_1}{\alpha_4} u(t) - \frac{1}{\alpha_4} \dot{\theta}(t) + \frac{\alpha_2}{\alpha_4} \text{sign}(\dot{\theta}(t)) + r\alpha_3 \tau_{fs}(t) + \alpha_3 \tau_g(\theta, t) \quad (3.14)$$

El método para identificar todas las constantes del modelo consiste en estudiar condiciones particulares en la ecuación del modelo, de manera de simplificar la estimación de las mismas. En primer lugar, se estimaron los valores de  $\alpha_1$  y  $\alpha_2$  estudiando la relación entre las velocidades estacionarias y el voltaje aplicado al motor. A continuación, se estudió el rozamiento estático del motor y bajo que condiciones actúa. Por último, se estimaron los valores de  $\alpha_3$  y  $\alpha_4$  utilizando cuadrados mínimos. En particular, primero se obtiene  $\alpha_4$

para algún caso particular donde se pueda despreciar el torque de gravedad, y por último, se estima  $\alpha_3$  para el caso general.

**Estimación de  $\alpha_1$  y  $\alpha_2$ :** Para estimar  $\alpha_1$  y  $\alpha_2$  vamos a estudiar la relación entre la velocidad estacionaria y el voltaje aplicado al motor. De esta manera se reemplazan en la ecuación del modelo 3.14, las condiciones de que la velocidad sea estacionaria ( $\ddot{\theta} = 0$ ) y distinta de cero ( $\tau_{fs} = 0$ ). Además, si el movimiento del motor es en el plano perpendicular a la gravedad podemos asumir  $\tau_g = 0$ . Para cumplir con la última condición se colocó de forma vertical logrando que la barra se mueva en el plano perpendicular a la gravedad. Reemplazando estas 3 condiciones en la ecuación del modelo (3.14) se obtiene:

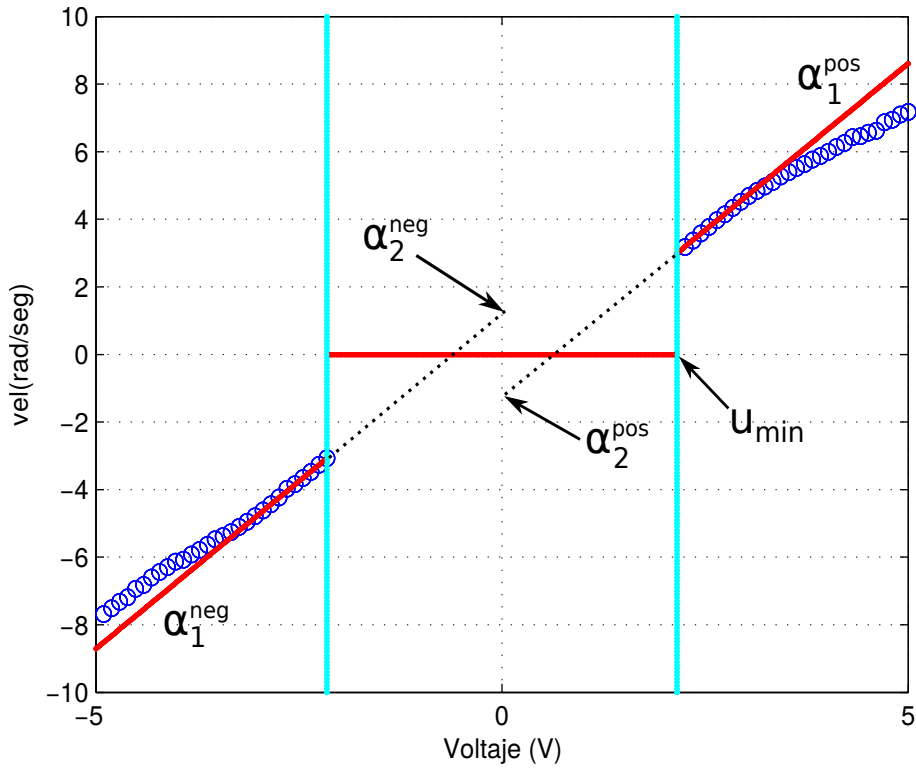
$$\dot{\theta}_{est} = \alpha_1 u + \alpha_2 \text{sgn}(\dot{\theta}) \quad (3.15)$$

En la ecuación 3.15 se observa que la velocidad angular estacionaria se relaciona linealmente con el voltaje aplicado al motor, es decir, si graficamos la velocidad estacionaria en función del voltaje observamos que  $\alpha_1$  representa la pendiente y  $\alpha_2$  la ordenada al origen.

Para replicar estas condiciones de forma experimental, en primer lugar se aplica al motor un voltaje fijo (comenzando por el voltaje máximo) y se espera que el motor alcance una velocidad estacionaria. A continuación, se apaga el motor ( $u(t) = 0$ ) y se espera que el motor se detenga. Se repite esta medición en el sentido contrario para estudiar velocidades positivas y negativas. Por último, se aplica un voltaje menor y se repite la medición. En la figura 3.2 se observan los resultados obtenidos.

En la figura 3.2 se muestra en azul los resultados obtenidos y en rojo el ajuste realizado utilizando la ecuación 3.15. Se observa que la relación lineal predicha por la ecuación (entre la velocidad y el voltaje), no se cumple en todo el rango de voltajes. Entonces se ajusto de forma lineal las mediciones en el rango  $[2, 15; 3, 30]V$  y  $[-2, 15; -3, 30]V$ .

Como se mencionó antes, en la figura 3.2  $\alpha_1$  representa la pendiente y  $\alpha_2$  la ordenada al origen. Se observa que las constantes  $\alpha_1$  y  $\alpha_2$  dependen de la dirección del motor. Entonces se estimaron dos constantes para velocidades



**Figura 3.2:** Relación lineal entre el voltaje aplicado al motor y la velocidad del mismo.

positivas y dos para velocidades negativas. Los valores obtenidos son:

$$\alpha_1^{pos} = 1,973V^{-1}s^{-1}, \alpha_2^{pos} = -1,256s^{-1}, \alpha_1^{neg} = 1,982V^{-1}s^{-1} \text{ y } \alpha_2^{neg} = -1,198s^{-1}.$$

**Estimación de  $\tau_{fs}$ :** Según la definición (ecuación 3.4) el rozamiento estático actúa cuando la velocidad es nula. En la figura 3.2 se observa que existe una rango de voltajes para los cuales la velocidad es nula, esta región se llama zona muerta y el máximo valor de voltaje dentro de este rango se llama  $u_{min} = 2,16V$ . La estimación de este valor resulta importante ya que permite estimar el rozamiento estático máximo ( $\tau_{fs}^{max}$ ) y, como se explica en la siguiente sección, sirve para corregir el efecto no lineal que produce el rozamiento estático.

Para tener rozamiento estático la velocidad y la aceleración tienen que ser necesariamente nulas, sin embargo, existe un instante de tiempo donde la

### Cap.3 Modelo Electro-Mecánico, Identificación y Simulación de una Articulación

---

velocidad es nula y la aceleración es distinta de 0, este instante es el pasaje de rozamiento estático a rozamiento dinámico y en ese punto el rozamiento estático es máximo. Se reemplaza en la ecuación del modelo 3.11 las condiciones: velocidad, aceleración y torque de gravedad nulos, y se obtiene:

$$\tau_{f_s}(t) = \frac{k_m}{R_a} u(t) \quad (3.16)$$

Sin embargo, según la figura 3.2 cuando el voltaje  $u(t)$  es mayor  $u_{min}$  la velocidad es no nula, entonces el máximo valor que toma  $\tau_{f_s}$  es:

$$\tau_{f_s}^{max} = \frac{k_m}{R_a} u_{min} \quad (3.17)$$

Este valor máximo es constante y también vale para el caso con gravedad. Sin embargo, cuando se tiene torque de gravedad la expresión del rozamiento estático es:

$$\tau_{f_s}(t) = \frac{k_m}{R_a} u(t) - \alpha_3 \tau_g(t) \quad (3.18)$$

**Estimación de  $\alpha_3$  y  $\alpha_4$ :** Para estimar las constantes  $\alpha_3$  y  $\alpha_4$  se integran la ecuación del modelo 3.14 mediante el método de Euler, el resultado se compara con la medición real y utilizando cuadrados mínimos se estiman  $\alpha_3$  y  $\alpha_4$ . Antes de integrar la ecuación es necesario tener en cuenta algunos detalles de implementación.

Para integrar la ecuación del modelo (3.14) es necesario tener en cuenta que dependiendo de las condiciones en las que se encuentre el motor cambian los términos de la ecuación, por ejemplo, los rozamientos dependen del signo de la velocidad y del valor del  $u(t)$  aplicado. Si la velocidad no es nula, se tiene fricción de Coulomb y fricción viscosa, pero por el contrario si la velocidad es nula vamos a tener rozamiento estático que a su vez depende del  $u(t)$ . Para

simplificar la implementación estos detalles se resumen pseudocódigo 2.

**Algorithm 2:** Modelo del motor con gravedad

```

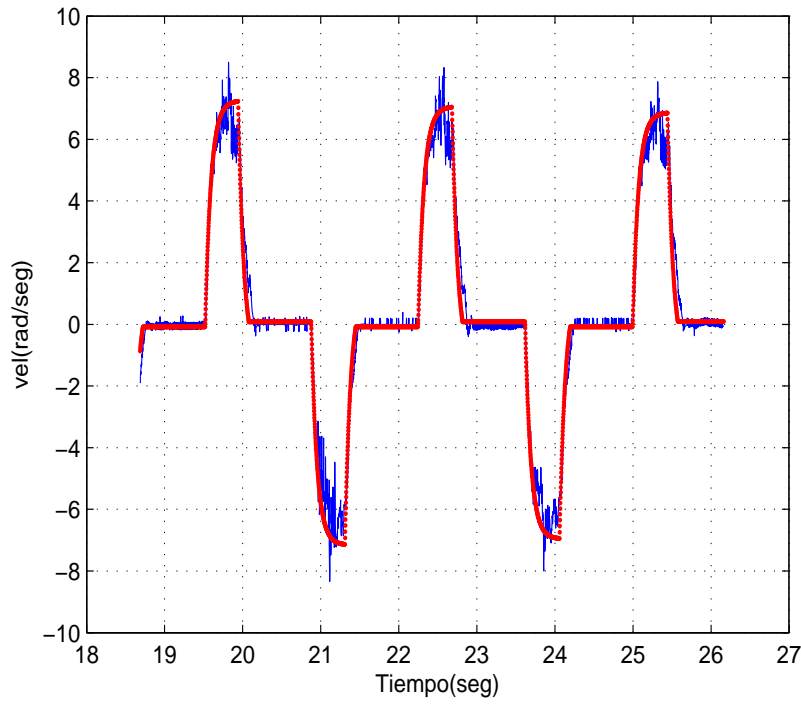
Input:  $u(t)$ 
Output:  $\theta, \dot{\theta}$ 
for el tiempo de ejecución do
    if  $\dot{\theta} \neq 0$  then
         $\ddot{\theta}(t) \leftarrow \frac{\alpha_1}{\alpha_4} u(t) - \frac{1}{\alpha_4} \dot{\theta}(t) - \frac{\alpha_2}{\alpha_4} \text{sign}(\dot{\theta}(t)) - \alpha_3 \tau_g(t)$ 
    else
        if  $\tau_{f_s}^{max} > \tau_m + \tau_g$  then
             $\ddot{\theta}(t) \leftarrow 0$  /* el roz. est. anula todo */
        else
             $\ddot{\theta}(t) \leftarrow \frac{\alpha_1}{\alpha_4} u(t) - r\alpha_3 \tau_{f_s}^{max} - \alpha_3 \tau_g(t)$ 
         $[\theta, \dot{\theta}] \leftarrow \text{Euler}(\ddot{\theta})$  /* con el método de Euler se integra  $\ddot{\theta}$  */
    */

```

Además, el motor es controlado por un microcontrolador y por ende los tiempos de actuado del mismo son discretos. Para estimar el tiempo de actuado ( $dt$ ), se promedió el tiempo de actuado de una medición real del motor obteniendo  $dt = 0,022ms$ . Se utilizó este tiempo como tiempo de sampleo del algoritmo, es decir, el ciclo principal corre cada  $0,022ms$ .

Teniendo en cuenta estas consideraciones se simula el modelo del motor para el caso donde se puede despreciar el torque de gravedad, es decir, el movimiento del motor era en el plano perpendicular a la gravedad, entonces  $\tau_g(t) = 0$ . En las simulaciones se utilizan como voltaje del motor  $u(t)$  los datos de voltaje obtenidos de la medición del motor. Se ajustó por cuadrados mínimos el valor de  $\alpha_4$  comparando la simulación con la medición real utilizada para estimar  $\alpha_1$  y  $\alpha_2$ . Se obtuvo que  $\alpha_4 = 0,04s$ . En la figura 3.3 se muestran en color azul la medición de la velocidad real del motor y en rojo el ajuste obtenido. En este caso los perfiles de velocidades son simétricos ya que no hay torque de gravedad.

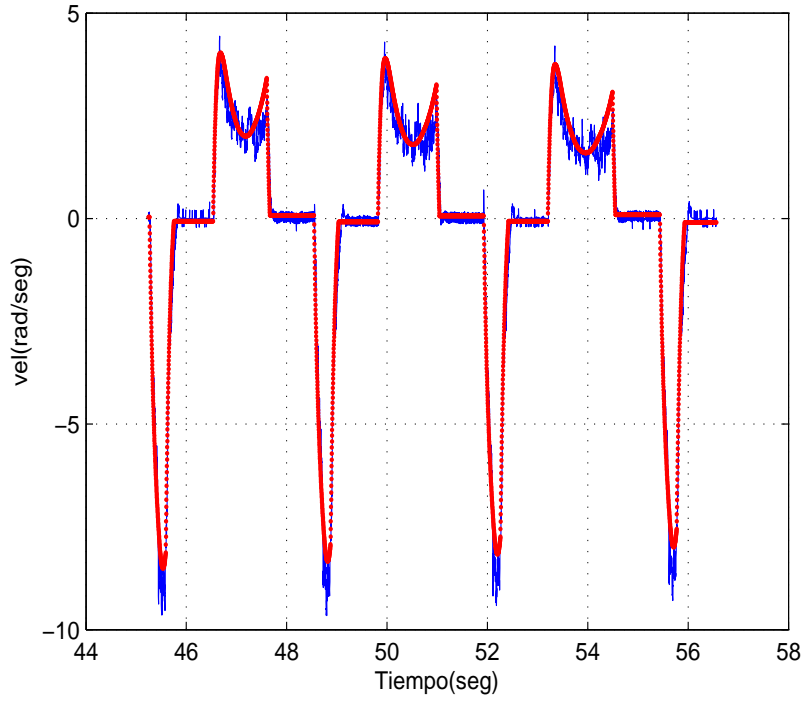
Para estimar  $\alpha_3$  se repitió la medición de velocidades, pero con la barra de forma vertical. De esta manera, el torque de gravedad somete a la barra a una fuerza no lineal que depende de la posición de la misma. En particular el



**Figura 3.3:** Simulación de la velocidad del motor para el caso sin torque de gravedad, en azul se observa la medición real y en rojo el ajuste realizado.

torque de gravedad viene dado por  $\tau_g(t) = mgl_{cm} \cos(\theta)$ , donde  $m$  es la masa de la barra,  $l_{cm}$  la distancia al centro de masa de la barra y  $g$  es la constante de gravedad.

Conociendo  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_4$  se utilizó el método de cuadrados mínimos para estimar  $\alpha_3$ , comparando la simulación con la medición real, obteniendo que  $\alpha_3 = 6900Kg^{-1}m^{-2}$ . El ajuste realizado se muestra en la figura 3.4. Se observa que a diferencia de la figura 3.3 donde no teníamos gravedad, el gráfico que se obtiene no es simétrico. Esto se debe a que en un sentido, en este caso para las velocidades negativas, la gravedad esta en contra del movimiento, es por eso que las velocidades son más baja; y en cambio, para las velocidades positivas la gravedad actúa en el sentido de giro haciendo que las velocidades sean más altas.



**Figura 3.4:** Simulación de la velocidad cuando el modelo considera torque de gravedad, en azul se observa la medición real y en rojo el ajuste realizado.

Una vez que se estimaron las 4 constantes  $\alpha_i$ , se desean obtener las constantes reales del modelo para ello, despejando del sistema de ecuaciones 3.13 se obtiene:

$$J_{eff} = \frac{r^2}{\alpha_3} \quad B_{eff} = \frac{J_{eff}}{\alpha_4} \quad k_m = \frac{\alpha_1 B_{eff} R_a}{r} \quad \tau_c = \frac{\alpha_2 B_{eff}}{r} \quad (3.19)$$

Entonces, reemplazando los valores estimados se obtienen las constantes del modelo. Los valores obtenidos se resumen en la tabla 3.1.

Los valores de la tabla 3.1 definen los parámetros característicos del modelo de motor utilizado.

### Cap.3 Modelo Electro-Mecánico, Identificación y Simulación de una Articulación

---

$J_{eff}$	=	$9,23e^{-08}kgm^2$
$B_{eff}$	=	$1,22e^{-06}Nms$
$k_m$	=	$3,60e^{-03}NmA^{-1}$
$\tau_c$	=	$4,50e^{-04}Nm$

**Tabla 3.1:** Parámetros característicos del modelo de motor, correspondientes al motor utilizado.



# Capítulo 4

## Modelo Dinámico y simulación de control

Para obtener un modelo completo de un manipulador es necesario describir las ecuaciones dinámicas que relacionan los torques aplicados en las articulaciones por los motores y las variables cinemáticas del robot. La dinámica se ocupa de la relación entre fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina [27]. Por lo tanto, el modelo dinámico de un robot tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas aplicadas en el mismo [10].

Esta relación se obtiene mediante el denominado **modelo dinámico**, que relaciona tres elementos:

1. La localización del robot definida por sus variables articulares y sus derivadas: velocidad y aceleración.
2. Las fuerzas y pares aplicados en las articulaciones (o en el extremo del robot).
3. Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

La obtención de este modelo para mecanismo de uno o dos grados de libertad no es excesivamente compleja, pero a medida que el número de grados de libertad aumentan, el planteamiento y obtención del modelo dinámico

se complica enormemente. Por este motivo no siempre es posible obtener un modelo dinámico expresado de una forma cerrada, esto es, mediante una serie de ecuaciones, normalmente de tipo diferencial de 2<sup>do</sup> orden, cuya integración permita conocer que movimiento surge al aplicar unas fuerzas o que fuerzas hay que aplicar para obtener un movimiento determinado. El modelo dinámico debe ser resuelto entonces de manera iterativa mediante la utilización de un procedimiento numérico.

El problema de la obtención del modelo dinámico de un robot es, por lo tanto, uno de los aspectos más engorrosos del modelado de los manipuladores robóticos, lo que ha llevado a ser obviado en numerosas ocasiones. Por razones didácticas en esta Tesis trabajamos con una simplificación del manipulador Scrobot-ER V Plus reduciendo la cantidad de grados de libertad a dos articulaciones, que corresponden a las articulaciones 2 y 3 del Scrobot.

El modelo dinámico es imprescindible para conseguir los siguientes fines:

1. Simulación del movimiento de un robot teniendo en cuenta las fuerzas que intervienen.
2. Diseño y evaluación de la estructura mecánica del robot.
3. Dimensionamiento de los actuadores.
4. Diseño y evaluación del control dinámico del robot.

Este último fin es evidentemente de gran importancia, pues de la calidad del control dinámico del robot depende la precisión y velocidad de sus movimientos. La gran complejidad ya comentada, existente en la obtención del modelo dinámico del robot, ha motivado a que se realicen ciertas simplificaciones, de manera que, así pueda ser utilizado en el diseño del controlador.

Es importante hacer notar que el modelo dinámico completo de un robot debe incluir no solo la dinámica de sus elementos (barras o eslabones) sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al modelo dinámico nuevas inercias, rozamientos, saturaciones de los circuitos electrónicos, etc. aumentando aún más su complejidad.

---

En este Capítulo veremos que las ecuaciones dinámicas de manipuladores pueden escribirse de la siguiente forma:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) + F(q, \dot{q}) \quad (4.1)$$

donde  $\tau$  es el vector torque (de dimensión  $nx1$ , donde  $n = \text{num. de links}$ ),  $q$  corresponde al vector de coordenadas articulares generalizadas (en nuestro caso  $q = [\theta_1, \theta_2, \dots, \theta_n]$ ),  $M$  es la matriz de inercia ( $\text{dim} = nxn$ ),  $V$  es el vector centrífugo y de Coriolis,  $G$  es el vector de gravedad y  $F$  es el vector de torques de fricción (todos de dimensión  $nx1$ ).

Una vez conocida la dinámica (ecuación 4.1), el objetivo es que el manipulador siga la trayectoria objetivo  $[q_d(t), \dot{q}_d(t), \ddot{q}_d(t)]$  propuesta por el generador de trayectoria desarrollado en el capítulo 2.

Una primer solución para que el robot siga la trayectoria deseada sería introducir esta información en la ecuación 4.1 y calcular los torques en las articulaciones. Sin embargo, en todo robot existen perturbaciones desconocidas que se acumulan a través del tiempo y harían fallar el método. Por este motivo, es necesario diseñar una estrategia de control que calcule los torques de los motores realimentando la información del error en el seguimiento de la trayectoria con el objetivo de anularlo.

En este capítulo desarrollamos los siguientes temas: En la sección 4.1.1 describimos un algoritmo general basado en las ecuaciones dinámicas de Lagrange que nos permite obtener las matrices  $M$  y los vectores  $V$ ,  $G$  para un manipulador cualquiera a partir de los utilizando los jacobianos asociados al movimiento de cada *link* del robot. En la sección 4.1.2 desarrollamos un programa en Matlab que calcula automáticamente las ecuaciones simbólicas asociadas a un manipulador y testamos su funcionamiento para un robot de dos articulaciones de rotación (RR).

En la Sección 4.2.1, se extiende el modelo mecánico incorporando el modelo electro-mecánico para los motores eléctricos de DC. En la sección 4.2.2 analizamos la utilidad del modelado dinámico para estudiar la importancia relativa de los distintos términos según las condiciones de trabajo impuestas al robot. Finalmente, en la Sección 4.2.3 nos adentramos en la problemática

de control donde testeamos (por medio de simulaciones) distintas estrategias llamadas control por computación de torque. Trataremos de encontrar una estrategia de control estable y simple que pueda ser implementada en un microcontrolador.

## 4.1. Modelo Dinámico de un Manipulador RR

### 4.1.1. Dinámica Lagrangiana de Manipuladores

La energía de un cuerpo rígido es la suma de dos términos: la energía de traslación del centro de masa, y la energía cinética de rotación del centro de masa. Entonces la energía cinética  $K$  viene dada por:

$$K = \frac{1}{2}mv^\top v + \frac{1}{2}w^\top Iw \quad (4.2)$$

donde  $m$  es la masa total del objeto,  $v$  y  $w$  los vectores de velocidad linear y angular, respectivamente, y  $I$  es el tensor de inercia (una matriz de 3x3).

Se considera un manipulador serial de  $n$  *links* o articulaciones, entonces es necesario encontrar los vectores de velocidad angular y linear conociendo la velocidad ( $\dot{q}$ ) de cada *link*. Para ello se utilizan los jacobianos  $J_{v_i}$  y  $J_{w_i}$  como se muestra en la ecuación 4.3,  $i$  corresponde al número de *link*. El cálculo de los jacobianos se explica más adelante.

$$v_i = J_{v_i}(q)\dot{q} \quad w_i = J_{w_i}(q)\dot{q} \quad (4.3)$$

Utilizando la ecuación 4.3 se reescribe la ecuación de  $K$  y se obtiene:

$$K = \frac{1}{2}\dot{q}^\top \sum_{i=1}^n [m_i J_{v_i}^\top(q) J_{v_i}(q) + J_{w_i}^\top(q) R_i(q) I_i R_i^\top(q) J_{w_i}(q)] \dot{q} \quad (4.4)$$

En otras palabras la energía cinética del manipulador se puede escribir como:

#### 4.1 Modelo Dinámico de un Manipulador RR

---

$$K = \frac{1}{2} \sum_{i,j=1}^n m_{ij}(q) \dot{q}_i \dot{q}_j := \frac{1}{2} \dot{q}^\top M(q) \dot{q} \quad (4.5)$$

donde  $M(q)$  es una matriz simétrica y definida positiva que se llama matriz de inercia.

Ahora se considera el término de energía potencial, en el caso de un cuerpo rígido la única fuente de energía potencial es la gravedad. La energía potencial en el *link*  $i$ -th se puede calcular asumiendo que la masa total del cuerpo está concentrada en el centro de masa, de esta forma se obtiene:

$$P_i = g^\top r_{c_i} m_i \quad (4.6)$$

donde  $g$  es el vector de la gravedad y el vector  $r_{c_i}$  son las coordenadas del centro de masa del link  $i$ . La energía potencial total de los  $n$  link es entonces:

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n g^\top r_{c_i} m_i \quad (4.7)$$

Conociendo la energía cinética y potencial del manipulador se utilizó el método de Euler Lagrange para encontrar las ecuaciones de movimiento del sistema. La ecuación de Lagrange es entonces:

$$L = K - P = \frac{1}{2} \sum_{i,j} m_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \quad (4.8)$$

Una vez escrito el Lagrangiano, aplicamos las ecuaciones de Lagrange obteniendo:

$$\sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \left\{ \frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial m_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j - \frac{\partial P}{\partial q_k} = \tau_k \quad (4.9)$$

que podemos reescribir como:

$$M(q) \ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (4.10)$$

El término  $V(q, \dot{q})$  puede ser reescrito como:

$$V(q, \dot{q}) = \sum_{i,j} c_{ijk} \dot{q}_i \dot{q}_j \quad (4.11)$$

donde

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right\} \quad (4.12)$$

se denominan los Símbolos de Cristofell.

En la ecuación 4.9 se observa como calcular la matriz  $M$  y los vectores  $V$  y  $G$ . Como la matriz  $M$  y la energía potencial  $P$  son funciones de  $q$  entonces, solo necesitamos calcular las derivadas parciales, que se pueden calcular de manera sencilla con el *toolbox* simbólico de Matlab.

**Calculo de Jacobianos:** Como se observa en la matriz 4.9, para calcular la matriz  $M$  es necesario conocer los jacobianos de traslación  $J_v$  y rotación  $J_w$  para el centro de masa de cada link.

**Jacobiano de traslación:** Para calcular el jacobiano  $J_v$ , partimos del hecho de que en el capítulo 1 se calcularon expresiones explícitas de cualquier punto de interés  $X_p$  en el robot. Por lo tanto para escribir  $J_v$  solo necesitamos calcular derivadas parciales de  $X_p$  respecto a las coordenadas generalizadas (ver ecuación 4.13).

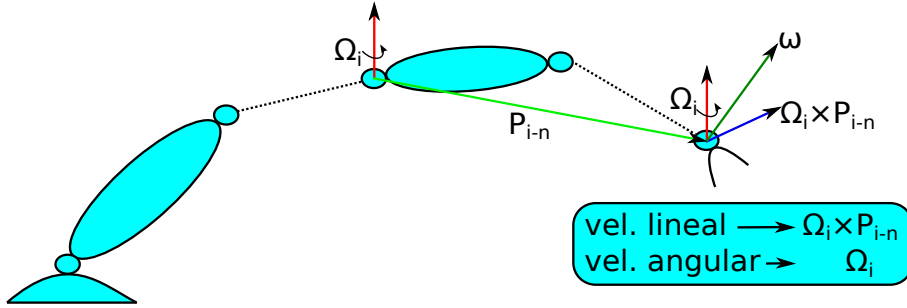
$$v = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \dot{x}_p = \frac{\partial x_p}{\partial q_1} \dot{q}_1 + \frac{\partial x_p}{\partial q_2} \dot{q}_2 + \dots + \frac{\partial x_p}{\partial q_n} \dot{q}_n \quad (4.13)$$

Entonces el jacobiano viene dado por:

$$J_v = \begin{pmatrix} \frac{\partial x_p}{\partial q_1} & \frac{\partial x_p}{\partial q_2} & \dots & \frac{\partial x_p}{\partial q_n} \end{pmatrix} \quad (4.14)$$

**Jacobiano de rotación:** Como se ve en la figura 4.1 la velocidad angular de rotación, solo sumará términos asociados a los giros de las articulaciones previas.

Entonces la velocidad angular  $\omega$  viene dada por la ecuación



**Figura 4.1:** Esquema manipulador serial, se observa que la velocidad angular de la articulación  $n$  depende únicamente de las velocidades angulares de las  $n - 1$  articulaciones anteriores.

$$\omega = \sum_{i=1}^n Z_i \dot{q}_i \quad (4.15)$$

De este modo, podemos escribir la velocidad angular de cualquier punto del robot respecto a la trama base como:

$$\omega = [Z_1 \quad Z_2 \quad \dots \quad Z_n] \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (4.16)$$

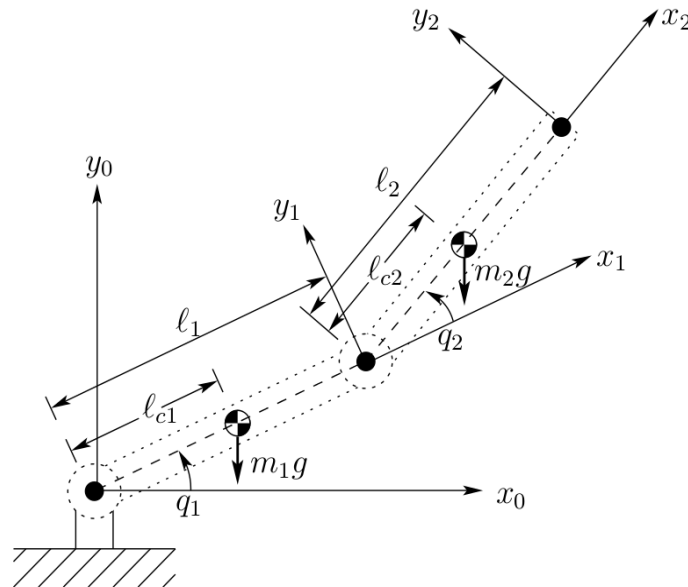
que se puede reescribir como:

$$\omega = J_w \dot{q} \quad (4.17)$$

Como se ve en la ecuación 4.16, el jacobiano de rotación tiene en la columna  $n$  el versor  $Z_n$  respecto a la trama base. Este versor también se calculó en el capítulo 1, donde vimos que  ${}^0Z_n$  corresponde a la tercer columna de la matriz  ${}^0R_n$ . En conclusión los jacobianos  $J_v$  y  $J_w$  se pueden calcular automáticamente con un software que resuelva ecuaciones simbólicas.

### 4.1.2. Cálculo Automático de las Ecuaciones de Movimiento para un manipulador RR

Aplicamos el método de calculo descrito en la sección anterior para calcular las ecuaciones de movimiento del robot RR (ver figura 4.2)



**Figura 4.2:** Esquema dinámico de un manipulador de dos grados de libertad. Imagen extraída de [4]

En el Apéndice 1 puede verse el programa desarrollado. La matriz M y los vectores V y G para este robot son:

$$M = \begin{bmatrix} m_2 l_1^2 + 2m_2 \cos(q_2) l_1 l_{c2} + m_1 l_{c1}^2 + I_{zz1} + I_{zz2} & m_2 l_{c2}^2 + l_1 m_2 \cos(q_2) l_{c2} + I_{zz2} \\ m_2 l_{c2}^2 + l_1 m_2 \cos(q_2) l_{c2} + I_{zz2} & m_2 l_{c2}^2 + I_{zz2} \end{bmatrix} \quad (4.18)$$



## 4.1 Modelo Dinámico de un Manipulador RR

---

Los elementos del vector  $V$  se los denomina  $c_{ijk}$  y para este robot son:

$$\begin{aligned}
 c_{111} &= \frac{1}{2} \frac{\partial m_{11}}{\partial q_1} = 0 \\
 c_{121} &= c_{211} = \frac{1}{2} \frac{\partial m_{11}}{\partial q_2} = -m_2 l_1 l_{c_2} \sin(q_2) \\
 c_{221} &= \frac{\partial m_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial m_{22}}{\partial q_1} \\
 c_{112} &= \frac{\partial m_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial m_{11}}{\partial q_2} \\
 c_{122} &= c_{212} = \frac{1}{2} \frac{\partial m_{22}}{\partial q_1} = 0 \\
 c_{222} &= \frac{1}{2} \frac{\partial m_{22}}{\partial q_2} = 0
 \end{aligned} \tag{4.19}$$

$$G = \begin{bmatrix} gm_2(l_{c_2} \cos(q_1 + q_2) + l_1 \cos(q_1)) + gl_{c_1} m_1 \cos(q_1) \\ gl_{c_2} m_2 \cos(q_1 + q_2) \end{bmatrix} \tag{4.20}$$

### 4.1.3. Modelo Dinámico Completo

En la sección 4.1.2 se presentó la dinámica de un manipulador robótico rígido, sin embargo, el robot necesita motores para moverse, entonces es necesario agregar el modelo dinámico de los motores (estudiado en el capítulo 3) al modelo dinámico del manipulador, para obtener una descripción completa de la dinámica del manipulador.

La ecuación para un manipulador de  $n$ -links viene dada por:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \tag{4.21}$$

donde  $q \in R^n$  es la variable de movimiento angular de cada *link*. Además, la ecuación dinámica de los motores de corriente continua que mueve cada articulación vienen dada por un sistema de  $n$  ecuaciones desacopladas:

$$J_m \ddot{q}_m + B_{eff} \dot{q}_m + \tau_{fs} + \tau_{fc} + r\tau = K_m u \tag{4.22}$$

donde  $u = \text{vec}\{u_{m_i}\} \in R^n$  con  $u_{m_i}$  el voltaje del motor  $i$ -ésimo y  $q_m = \text{vec}\{q_{m_i}\} \in R^n$ , con  $q_{m_i}$  la posición angular del motor  $i$ -ésimo. La entrada del control es el voltaje del motor  $u$ .

Las matrices de coeficientes del manipulador son todas constantes dadas por:

$$\begin{aligned}
 J_m &= \text{diag}\{J_{m_i}\} \\
 B &= \text{diag}\{B_{m_i} + \frac{k_{m_i}^2}{R_{a_i}}\} \\
 r &= \text{diag}\{r_i\} \\
 K_m &= \text{diag}\{\frac{k_{m_i}}{R_{a_i}}\}
 \end{aligned} \tag{4.23}$$

donde el motor  $i$ -esimo tiene constante de inercia  $J_{m_i}$ , constantes de fricción viscosa  $B_{m_i}$ , constante electromotriz  $k_{m_i}$  y resistencia  $R_{a_i}$ .

La reducción  $r_i$  del motor  $i$ -esimo al  $i$ -esimo *link* se define como:

$$q_i = r_i q_{m_i} \quad o \quad q = r q_m \tag{4.24}$$

La fricción viscosa viene dada en forma vectorial como  $\tau_{f_s} = \text{vec}\{\tau_{f_{s_i}}\}$  y la fricción de Coulomb como  $\tau_{f_c} = \text{vec}\{\tau_{f_{c_i}}\}$ .

Usando la ecuación 4.24 para eliminar  $q_m$  de la ecuación 4.22, y sustituyendo  $\tau$  de la ecuación 4.21 se obtiene:

$$(J_m + r^2 M)\ddot{q} + B\dot{q} + r^2 V + r(\tau_{f_s} + \tau_{f_c}) + r^2 G = K_m u \tag{4.25}$$

Renombrando se obtiene:

$$M_1(q)\ddot{q} + F_{visc} + V_1(q, \dot{q}) + F_{fric}(q) + G_1(q) = K_1 u \tag{4.26}$$

Es importante notar que la dinámica completa del manipulador 4.26 tiene la misma forma funcional que la dinámica del manipulador 4.21.

## 4.2. Simulación del Control de un Manipulador Simple por el Método de Computación de Torque

Se han propuesto muchos tipos de esquemas de control para manipuladores. La mayoría de ellos se pueden considerar casos especiales de la clase controles de torque computado o *computed-torque controllers*.

## 4.2 Simulación del Control de un Manipulador Simple por el Método de Computación de Torque

---

El control por computación de torque permite obtener convenientemente controladores de robots muy eficaces, mientras que proporciona un marco que nos permite unir el control de juntas independientes PID clásico y algunas técnicas de diseño modernas.

### 4.2.1. Control por el Método de Computación de Torques

El problema de controlar un manipulador es que estos no se adaptan bien a métodos de linealización local, dado que se mueven constantemente a lo largo de regiones de su espacio de trabajo tan ampliamente separadas que no puede encontrarse una linealización válida para todas las regiones. Básicamente, esto implica que resulta muy difícil controlar un sistema no lineal (como son los manipuladores) con controles lineales. Algunos métodos plantean linealizaciones móviles, es decir, usar controles lineales cuyas constantes dependan del tiempo. Si bien esta técnica resulta útil en ciertos casos, el problema general se resuelve trabajando directamente con las ecuaciones no lineales de movimiento, sin recurrir a linealizaciones para derivar un controlador.

A continuación se detalla el método para encontrar un control PID por computación de torque, sin embargo, la metodología de este caso particular se aplica al resto de los controles estudiados. Para obtener el control por computación de torque se parte de la ecuación del modelo dinámico completo 4.26, y se propone:

$$u = K_1^{-1}(\alpha\tau' + \beta) \quad (4.27)$$

donde  $u$  es el vector de voltajes (de  $dim = nx1$ ) de las  $n$  articulaciones, es decir, la entrada del sistema. Se eligen  $\alpha$  y  $\beta$  de la siguiente manera:

$$\begin{aligned} \alpha &= M_1(q) \\ \beta &= F_{visc} + V_1(q, \dot{q}) + F_{fric}(q) + G_1(q) := N(q, \dot{q}) \end{aligned} \quad (4.28)$$

notar que  $N(q, \dot{q})$  representa los términos no lineales del sistema.

El torque  $\tau'$  se propone como un control PID clásico, de la siguiente forma:

$$\tau' = \ddot{q}_d + K_d \dot{e} + K_p e + K_I \epsilon \quad (4.29)$$

donde  $e = q_d - q$  y  $\dot{e} = \dot{q}_d - \dot{q}$ . Utilizando las ecuaciones 4.27, 4.28 y 4.29 se obtiene:

$$\ddot{e} + K_d \dot{e} + K_p e + K_I \epsilon = 0 \quad (4.30)$$

donde las matrices  $K_d$  y  $K_p$  son diagonales y entonces se puede reescribir para cada articulación:

$$\ddot{e}_i + K_{d_i} \dot{e}_i + K_{p_i} e_i + K_{I_i} \epsilon = 0 \quad (4.31)$$

donde cada  $K_{d_i}$ ,  $K_{p_i}$  y  $K_{I_i}$  se eligen de manera que  $e_i$  tienda asintóticamente a 0.

Resumiendo el voltaje para un control con computación de torque PID es:

$$u = K_1^{-1}(M(q)(\ddot{q}_d + K_d \dot{e} + K_p e + K_I \epsilon) + N(q, \dot{q})) \quad (4.32)$$

Siguiendo el mismo procedimiento se obtiene el voltaje de control para diferentes estrategias. En la tabla 4.1 se comparan distintas estrategias de control que se utilizan para controlar manipuladores.

---

1	<b>PID Computed Torque</b>	$u = K_1^{-1}(M(q)(\ddot{q}_d + K_d \dot{e} + K_p e + K_i \epsilon) + N(q, \dot{q}))$
2	<b>PID sin <math>V_1</math></b>	$u = K_1^{-1}(M(q)(\ddot{q}_d + K_d \dot{e} + K_p e + K_i \epsilon) + F_{visc} + F_{fric}(q) + G_1(q))$
3	<b>PID Control Gravedad</b>	$u = K_1^{-1}M_d(K_d \dot{e} + K_p e + K_i \epsilon + G(q))$
4	<b>PID Clásico</b>	$u = K_1^{-1}M_d(K_d \dot{e} + K_p e + K_i \epsilon)$

---

**Tabla 4.1:** Tabla comparativa de diferentes controles utilizados en manipuladores, los primeros tres controles se obtienen utilizando el método de torques computados. La matriz  $M_d$  es una matriz diagonal que corresponde a los valores máximos de la diagonal de  $M(q)$ .

En la sección 4.2.3 se muestran las simulaciones de cada uno de estos controles aplicados a un manipulador RR.

### 4.2.2. Análisis de los Términos del Modelo Completo

El robot RR que queremos simular corresponde a los primeros dos links de un robot educativo que se está desarrollando. Los movimientos de los links de este manipulador se mueven en un plano vertical, es decir, su movimiento es afectado por la gravedad. Los valores de los parámetros propuestos para este manipulador se muestran en la tabla 4.2.

Art. $i$ -esima	$m_i[Kg]$	$l_i[m]$	$l_{c_i}[m]$
1	0,2	0,1	0,05
2	0,1	0,05	0,025

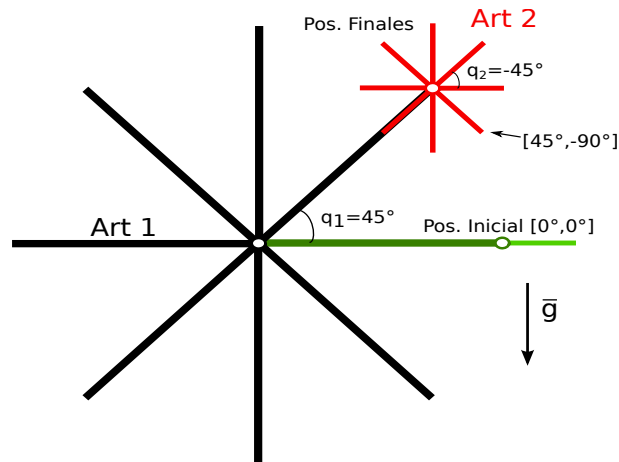
**Tabla 4.2:** Parámetros del robot RR simulado.

Se realizó un estudio sobre los ordenes de magnitud de los términos del modelo dinámico propuesto ( $G_1, F_{fric}, F_{visc}, V_1$ ) para trayectorias dentro del espacio de trabajo del manipulador. De esta forma, se puede determinar cuales son los términos relevantes y si es necesario tener en cuenta todos los términos para controlar el robot.

Se eligió la posición inicial del manipulador como la posición  $[0^\circ, 0^\circ]$  y se simularon 81 posiciones finales. Para abarcar todas las regiones, las posiciones finales, se eligieron sumando  $45^\circ$  a la posición inicial en ambos sentidos y en ambas articulaciones. En la figura 4.3 se muestra un esquema de la posición inicial (en color verde) y posiciones finales propuestas (en color negro y rojo).

Para generar la trayectoria entre la posición inicial y final que tiene que recorrer cada articulación se utilizó el generador de trayectorias polinomial presentado en el capítulo 3, donde el tiempo de llegada es de 5 segundos para todos los casos. En la figura 4.4 se muestran las trayectorias deseadas para la articulación 1 (izquierda) y para la articulación 2 (derecha). Combinando las 9 posiciones finales de cada articulación se obtienen 81 casos de estudio.

Como se mencionó anteriormente se estudió la importancia relativa entre los términos del modelo propuesto ( $G_1, F_{fric}, F_{visc}, V_1$ ). En la figura 4.5 se muestran dichos términos para un movimiento dado, a la izquierda se muestran los términos para la articulación 1, y a la derecha para la articulación



**Figura 4.3:** Posición inicial (en verde) y posiciones finales en negro (art. 1) y rojo (art. 2). En total se tienen 81 posiciones finales.

dos.

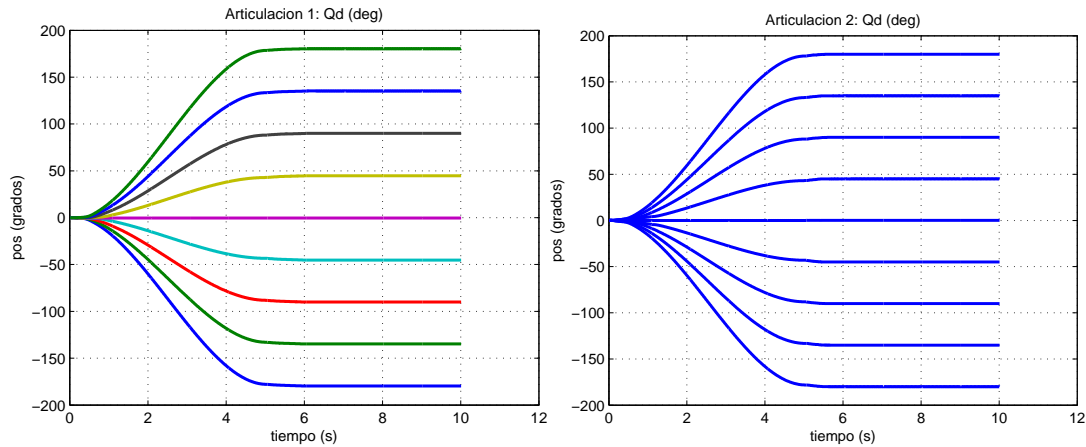
En la figura 4.5 se observa que para el primer *link* el término dominante es el torque de gravedad, y el término de Coriolis ( $V_1$ ) resulta despreciable para todo el movimiento. Para la articulación 2 se observa que al igual que en el caso anterior, el término de Coriolis ( $V_1$ ) resulta despreciable. Si bien en la figura 4.5 muestra un caso particular, las observaciones mencionadas se repiten para los 81 casos de prueba.

### 4.2.3. Control del RR

En esta sección se testearon mediante simulaciones las cuatro estrategias de control detalladas en la tabla 4.1.

Las formas de cuantificar el error de una estrategia de control son variadas, en este trabajo se utilizan dos mediciones, que llamaremos medición estacionaria y medición promedio. En el primer caso, se estudia el valor final al que llega la articulación, entonces se puede saber si el control converge al valor deseado y de no hacerlo cual es el error final que se comete, como las mediciones tienen ruido no se toma el valor final, si no el promedio del último segundo. El segundo caso, (medición promedio) corresponde a la integral del error en todo el tiempo de ejecución dividido el periodo o tiempo de ejecución, esta medición permite cuantificar la eficiencia de convergencia

## 4.2 Simulación del Control de un Manipulador Simple por el Método de Computación de Torque



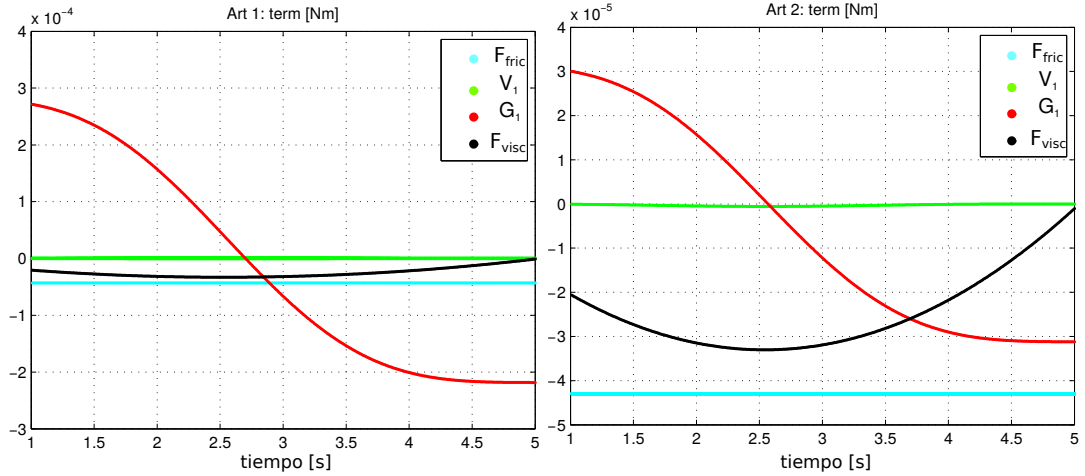
**Figura 4.4:** Trayectorias deseadas para las dos articulaciones (art. 1 a la izquierda y art. 2 a la derecha). Combinando las 9 posiciones finales de cada articulación se obtienen los 81 casos de estudio.

al valor deseado.

La convergencia de los métodos (y por ende el error de convergencia) depende fuertemente de que la elección de constantes ( $k_p, k_d, k_i$ ) sea acertada. Antes de comparar la *performance* de los controles es necesario encontrar constantes que maximicen la eficiencia del control, minimizando el error producido por la elección de constantes. El procedimiento utilizado consistió en hallar un conjunto de constantes de forma experimental, de manera que el sistema converja, y a continuación se estudió la variación del error promedio para cambios de estas constantes. Este método nos permitió encontrar las constantes que minimizan el error promedio para cada método de control propuesto.

Utilizando las mejores constantes para cada método, se midieron el error promedio y el error estacionario. Los errores promedio de cada control se normalizaron con respecto al control de máximo error, que era el control 4 (PID Clásico).

En la tabla 4.3 se observa que los controles 1 y 2 presentan los mismos resultados, esto se debe a que, como se mencionó en la sección anterior, el término  $V_1$  resulta despreciable. El error promedio se normalizó con respecto al mayor error que correspondía al control PID clásico. De esta forma se



**Figura 4.5:** Comparación del orden de magnitud de términos del modelo dinámico ( $G_1, F_{fric}, F_{visc}, V_1$ ) para un caso particular, a la izquierda los términos de la art. 1 y a la derecha los de la art. 2.

Error \ Control	1	2	3	4
promedio (%)	$6,5 \pm 1,5$	$6,5 \pm 1,5$	$35 \pm 5$	$100 \pm 5$
estacionario (°)	$0,22 \pm 0,01$	$0,22 \pm 0,01$	$0,60 \pm 0,13$	$1,25 \pm 0,37$

**Tabla 4.3:** Comparación de los errores para los distintos métodos propuestos. El método 1 corresponde al método de compensación total, el método dos correspondo al método de compensación sin Coriolis, el método 3 es el control PID más gravedad y por último el control 4 corresponde al PID clasico.

observa claramente que el control 1 posee un eficiencia significativamente mayor al control 3 y 4. El error estacionario se muestra en (°), y se observa que todos los métodos convergen al valor deseado con un error menor a 2°.

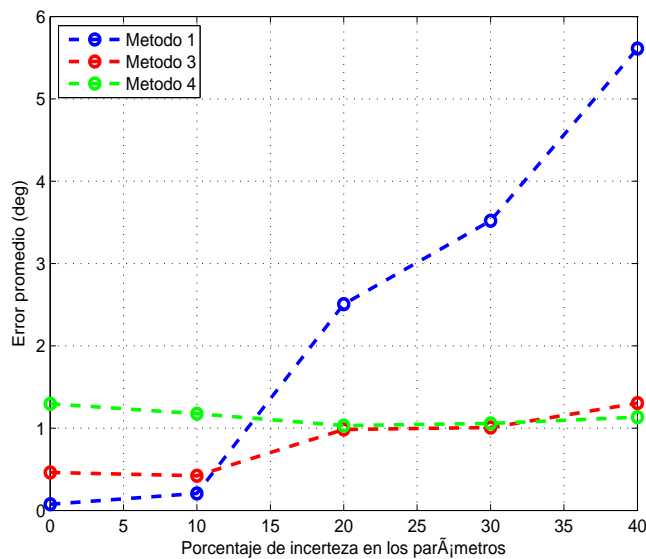
Los controles por torque computado asumen que los parámetros característicos del modelo son conocidos, por lo que se estudió la dependencia de los resultados obtenidos (error promedio para cada control) si los términos del modelo no están bien identificados.

Se simuló un error en la identificación de los siguientes parámetros:  $J_{eff}, B_{eff}, \tau_c$  y  $k_m$ . Que son los parámetros identificados en el capítulo 3. El error en la identificación simulado fue de 0 %, 10 %, 20 %, 30 %, 40 %. Además, se usó un



## 4.2 Simulación del Control de un Manipulador Simple por el Método de Computación de Torque

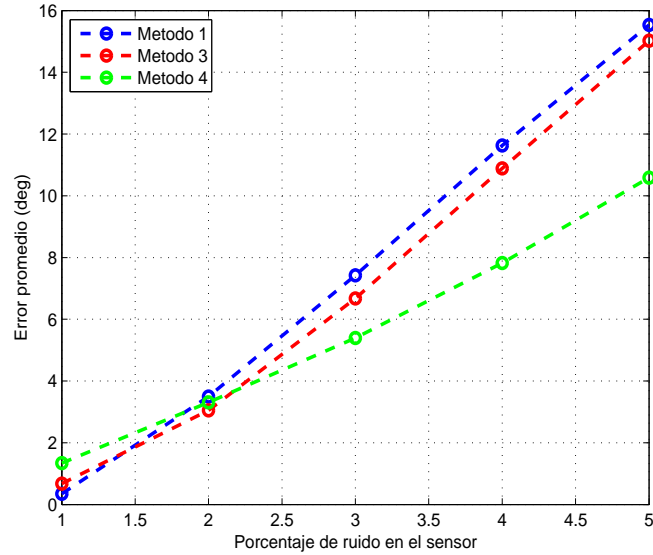
error de  $1\% = 10$  divisiones DAC para el sensor de posición. Para cada condición se promedió 5 veces y los resultados obtenidos se muestran en la figura 4.6. Se ve claramente que para los métodos 1 y 3, que dependen de la identificación del modelo, el error aumenta pero el control no se vuelve inestable.



**Figura 4.6:** Dependencia de los controles propuestos con respecto al ruido en los parámetros identificados. Se observa que los controles 1 y 3, que dependen del modelo, pierden precisión con el aumento del ruido.

También se estudió la dependencia de los métodos de control propuestos con el ruido en el sensor que mide la posición del motor. Se simuló un error de medición de  $1\%$ ,  $2\%$ ,  $3\%$ ,  $4\%$ ,  $5\%$ . Se usó un error de 10, 20, 30, 40, 50 divisiones DAC para el sensor de posición. Para cada condición se promedió 5 veces. Se ve claramente que el error aumenta pero el control no se vuelve inestable (ver figura 4.7).

En conclusión se observa que el esquema de compensación total asegura la anulación asintótica del error si no hay ruido y el sistema está perfectamente identificado. Se observó que este esquema es el que produce menos error y genera una convergencia más rápida. Además, se encontró que como



**Figura 4.7:** Dependencia de los controles propuestos con respecto al ruido en la medición de la posición del motor. Se observa que el ruido afecta a los tres controles, disminuyendo su precisión.

$V$  (Coriolis) es despreciable los esquemas de control propuestos 1 y 2 producen los mismos resultados. Por otro lado el método de compensación de gravedad arroja resultados aceptables, y la dependencia con la identificación de los parámetros es menor que para el método 1. Finalmente las simulaciones muestran que el control PID clásico se mantiene estable, a pesar de su simpleza, para los parámetros óptimos. Sin embargo, tiene mayor error estacionario y tarda más en asegurar la convergencia.

# Capítulo 5

## Control de una articulación

Gracias al modelo dinámico simulado en el capítulo anterior se estudiaron diferentes estrategias de control y se determinó que el control PID se comporta mejor cuando aumenta la incerteza de los parámetros de identificación del motor. Debido a esto y a su simpleza, se optó por este tipo de control para su implementación en esta Tesis como primera opción. En este Capítulo se detalla la implementación de un control tipo PID clásico en una placa Arduino y se estudia la *performance* del control en el motor real. Además, se simula el mismo control utilizando el modelo electro-mecánico desarrollado en el Capítulo 3. En la Sección 5.1 se describen detalles de implementación del control tanto para simularlo como para implementarlo en el motor real. Por último, en la Sección 5.2 se muestran y comparan los resultados obtenidos en la simulación del control y en el control del motor real.

### 5.1. Implementación

En primer lugar se implementó un control PID para el motor detallado en el capítulo 3, el motor se encontraba anclado a la barra rígida descrita en dicho capítulo. De esta forma, el motor está sometido al torque de gravedad, que es no lineal. Las constantes de PID utilizadas son:  $k_p = 1 \frac{V}{deg}$ ,  $k_d = 0,04 \frac{Vs}{deg}$  y  $k_i = 0,001 \frac{V}{s deg}$ . Estas constantes se utilizaron para el motor real y para las simulaciones.

Para controlar el motor se utilizó una placa Arduino Uno, esta plataforma permite utilizar una gran cantidad de librerías de código abierto que simplifican la implementación del control. En particular se utilizó la librería llamada PID\_v1, que permite implementar el control PID de manera sencilla.

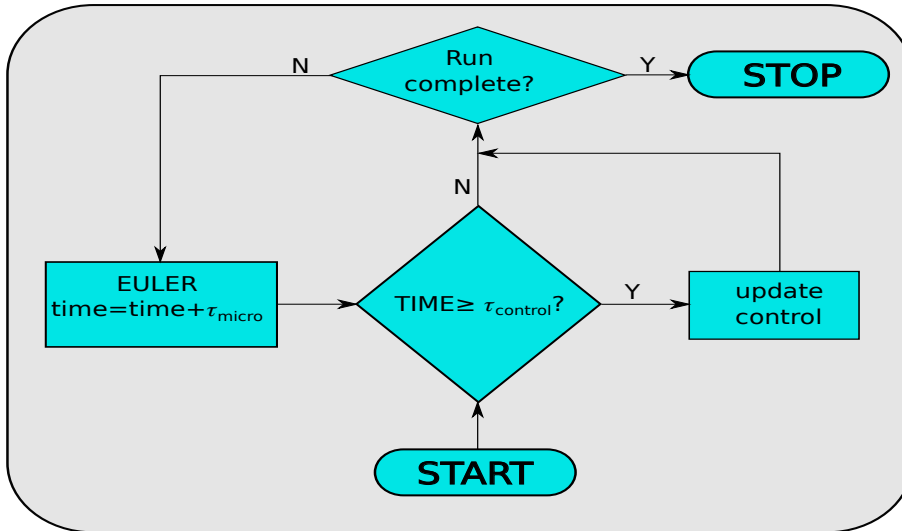
Para ajustar las constantes del PID se utilizó el método de Ziegler-Nichols [26], en el cual se propone un procedimiento para encontrar las ganancias óptimas del controlador basándose en la observación de la respuesta del sistema ante entradas tipo escalón y su respuesta en frecuencia.

Como se estudió en el capítulo 3 el sistema que se quiere controlar es no lineal, debido al torque de gravedad, al torque de fricción dinámica y al torque de fricción estática. En el capítulo 4 se verificó que un control PID clásico es capaz de compensar lo no linealidades provocadas por el torque de gravedad y la fricción dinámica. Para compensar la no linealidad producida por el torque de fricción estática existen diferentes métodos. La idea básica es incluir en el lazo de control un algoritmo que se anticipe a los efectos de la fricción estática y ayude a corregirlos. En este trabajo utilizamos la compensación propuesta en [25], la cual consiste en realizar una identificación experimental de  $u_{min}$  y utilizar la función de signo para agregarlo a la señal de control.

El primer paso para la compensación de la zona muerta, es la identificación de el voltaje  $u_{min}$ . Para esto es necesario utilizar la figura 3.2, donde se gráfico la velocidad en función de el voltaje aplicado al motor. Cuando el voltaje aplicado comienza a generar movimiento, lo denominamos  $u_{min} = 2,16V$ . Una vez identificado el voltaje de rompimiento, la señal de control con compensación de la zona muerta será  $u'(t) = u(t) + sign(u)u_{min}$ .

Otro detalle de implementación a tener en cuenta es que la mayoría de los controles son diseñados para tiempos continuos, pero son aplicados discretamente en robots reales. Es decir, las señales de control se actualizan en intervalos de tiempo discretos que dependen del microcontrolador utilizado. Es necesario simular que el tiempo de actualización de nuestro control es discreto, es decir, el tiempo en el que actualizamos la entrada del sistema. Lo llamamos tiempo de *sampleo* del control ( $\tau_{control}$ ), la elección de este valor depende del sistema en particular a controlar. Además, es necesario tener en cuenta el tiempo de ejecución ( $\tau_{micro}$ ) del bucle principal.

Para simular un control discreto se utilizará el esquema mostrado en la figura 5.1.



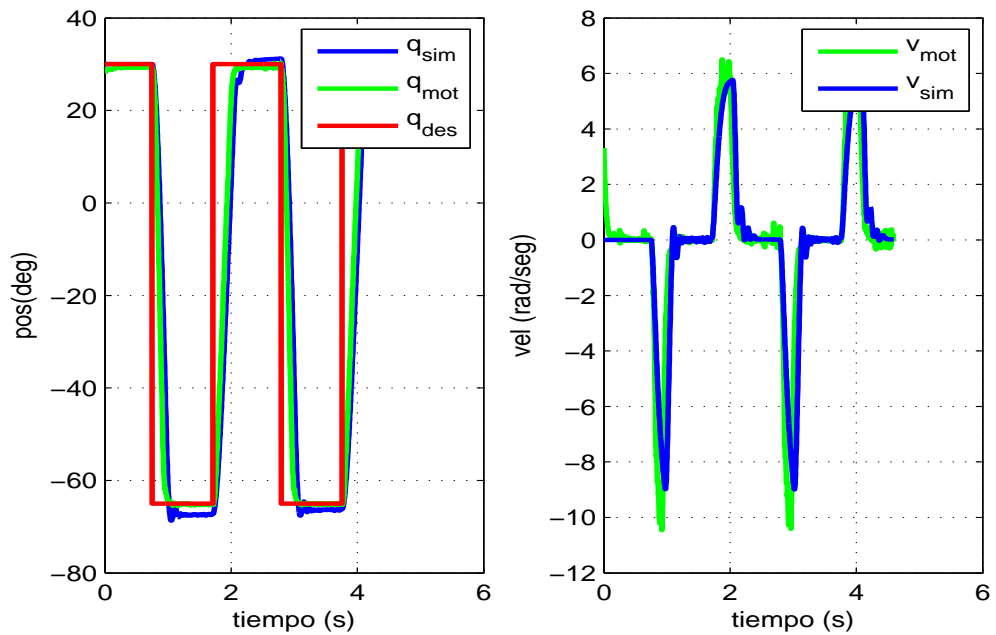
**Figura 5.1:** Esquema del ciclo de control discreto.

## 5.2. Control de un Motor DC

Para estudiar la robustez del control implementado y compararlo con las simulaciones, se realizaron dos tipos de experimentos, en primer lugar, se estudió la respuesta del motor ante un estímulo tipo escalón; y en el segundo caso, se estudió como se comportaba el controlador ante una trayectoria deseada tipo senoidal. En el primero de los caso la trayectoria deseada del manipulador es una señal cuadrada, donde la diferencia entre la parte alta y baja de la señal representa un movimiento de  $170^\circ$ . El periodo de la misma se eligió de manera de darle tiempo al control de converger. La elección de un señal senoidal, en el segundo experimento, se debe a que para seguir una trayectoria de este tipo, el motor tiene que cambiar constantemente de velocidad, ya que la consigna cambia con el tiempo.

En la figura 5.2 se muestra la *performance* del control PID simulado para una trayectoria rectangular, es decir, un estímulo tipo escalón. En el gráfico

de la izquierda se comparan la posición deseada (en rojo), la posición real del motor (en verde) y la posición simulada (en azul). En la figura de la derecha se muestran las velocidades para la simulación (en azul) y para el motor real (en verde).

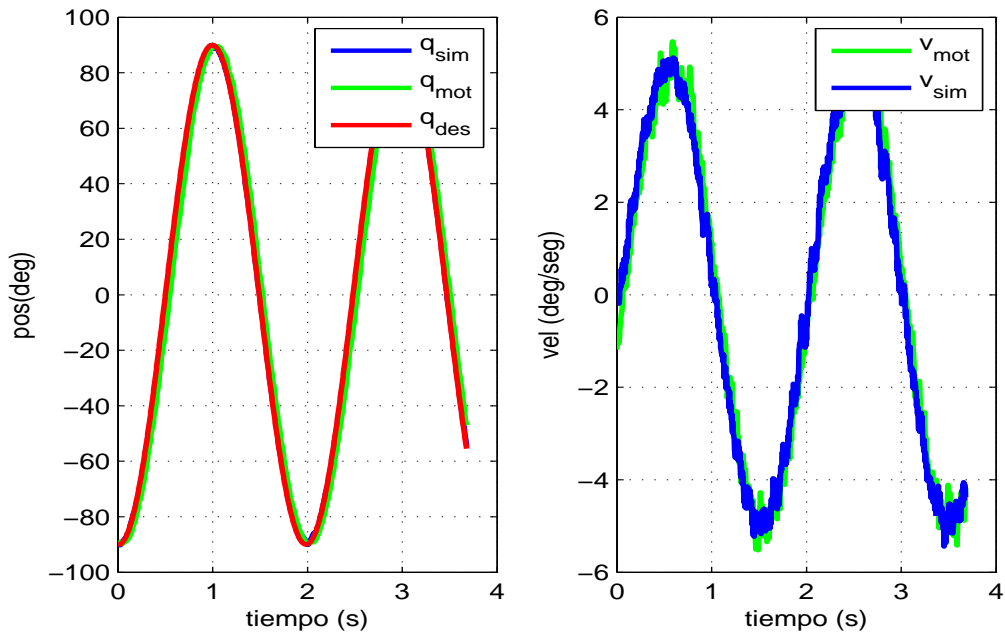


**Figura 5.2:** Comportamiento de la simulación y el motor real ante un estímulo tipo escalón para un control PID.

Como puede observarse en la figura 5.2 la simulación se comporta de manera similar al comportamiento del motor real. Se verificó que con las constantes de PID utilizadas el control convergía para las siguientes posiciones deseadas:  $90^\circ$ ,  $45^\circ$ ,  $0^\circ$ ,  $-45^\circ$  y  $-90^\circ$ . Es importante resaltar que las constantes son las mismas utilizadas en la simulación y en el control sobre el motor real. Se midió el error estacionario para los dos casos (simulación y motor real). El estacionario para el motor real es de  $0,50^\circ \pm 0,25^\circ$  y para la simulación de  $0,48^\circ \pm 1,77^\circ$ .

Además, del estímulo tipo escalón, se utilizó como trayectoria deseada una señal sinusoidal. De esta forma el motor cambia de velocidad constantemente. En la figura 5.3 se comparan las posiciones y velocidades simuladas y reales.

En el gráfico de la izquierda se comparan la posición deseada (en rojo), la posición real del motor (en verde) y la posición simulada (en azul), y en la figura de la derecha se muestran las velocidades para la simulación (en azul) y para el motor real (en verde).



**Figura 5.3:** Comportamiento de la simulación y el motor real ante un estímulo sinusoidal para un control PID.

Como se observa en la figura 5.3 se logró controlar una motor DC incluso cuando las trayectorias son senoidales, este tipo de trayectorias son las generadas por el generador de trayectorias desarrollado en el capítulo 2. Además, el comportamiento simulado es similar al comportamiento presentado por el motor real, lo que verifica la validez del modelo propuesto.

Para este motor en particular utilizando un control PID logramos los resultados esperados, sin necesidad de controles más complejos. Además, el hecho de que las constantes del control son las mismas en la simulación y en el control del motor real resulta muy importante, ya que, además de corroborar la validez del modelo, implica que el modelo simulado sirve para probar otras

## **Cap.5** Control de una articulación

---

constantes o incluso otras estrategias de control sin necesidad de utilizar el motor real y evitar así someterlo a sobre esfuerzos o la posibilidad de dañarlo.



# Capítulo 6

## Conclusiones

En este trabajo estudiamos una serie de problemas relacionados con el control de un manipulador robótico. El objetivo es lograr que el efector del robot siga una trayectoria deseada con error acotado aún cuando existen perturbaciones del sistema parcialmente conocidas.

En primer lugar se estudió el modelo cinemático del Scorbot-ER V Plus (Sección 1.2), para describir las relaciones entre tramas consecutivas se utilizó el formalismo de DH. Se resolvió el modelo Cinemático Directo y Cinemático Inverso del manipulador Scorbot-ER V Plus. Este modelo se testó por medio de simulaciones y de forma experimental. Para controlar este robot se desarrolló una interfaz de comunicación serial en Matlab (Sección 1.3). Se encontró que el error que el manipulador comete con nuestro modelo es menor al error que comete con el modelo cinemático provisto de fábrica (no calibrado). Como trabajo futuro se plantea desarrollar un método de autocalibración utilizando cámaras que nos permiten estimar de manera precisa la posición del robot.

Una vez resuelto el modelo cinemático, estudiamos el problema de generar trayectorias que permitan mover al robot entre dos poses de interés. En este caso nos propusimos estudiar métodos que no exijan introducir una gran cantidad de puntos para detallar el movimiento del robot. Se implementaron dos esquemas, por un lado el polinómico (Sección 2.1) (que se define en el espacio de articulación) que tiene la ventaja de no tener problemas con

las singularidades ( $J$  no inversible), pero no genera trayectorias simples en el espacio cartesiano. Por otro lado, se implementó el método del jacobiano (Sección 2.2) (que se define en el espacio cartesiano) que permite generar trayectorias simples en el espacio cartesiano y esto resulta muy importante cuando se desea evadir obstáculos. En el futuro nos proponemos desarrollar algoritmos que permitan evitar singularidades, ya que los torques de los motores divergen en esos puntos.

Luego de resolver el problema desde un punto de vista cinemático, se encaró el problema de identificación y control del Scorbot-ER V Plus. Este robot está diseñado como un sistema tipo caja cerrada y utiliza un software de control propietario, provisto por la empresa que fabrica el Scorbot que se encuentra obsoleto. Debido a esto no nos fue posible programar las estrategias de control propuestas sin eliminar el sistema de control provisto de fábrica. Por este motivo decidimos estudiar la dinámica del motor en el manipulador RB4 de la empresa nacional KitRobot que nos permitió tener completa libertad de control. Este manipulador utiliza servomotores de bajo costo que sacandole la placa de control, se transforman en un motor con reducción y un sensor de posición absoluto. Utilizando estos motores el costo de construcción baja mucho ya que no utilizamos encoders ni reducciones exteriores. En el Capítulo 3 estudiamos el modelo electro-mecánico de motores de corriente continua. En particular, identificamos el servo mencionado anteriormente y validamos el modelo electro-mecánico, comparando las simulaciones realizadas con las mediciones en el motor real. Encontramos que en este tipo de motores el rozamiento dinámico y estático son un factor importante a tener en cuenta, sin embargo, se logró modelarlos.

En el Capítulo 4, estudiamos y aplicamos el formalismo de Lagrange para obtener las ecuaciones de movimiento mecánicas del robot. Además, desarrollamos un programa que calcula automáticamente las ecuaciones simbólicas. Se encontró que cuando se agrega el modelo de los actuadores, al modelo dinámico de las articulaciones las ecuaciones de movimiento mantienen su estructura. Considerando el modelo dinámico completo (dinámica de las articulaciones más dinámica de los motores) podemos dimensionar los motores, dadas las condiciones de trabajo del robot. El programa desarro-

---

llado puede extenderse a más grados de libertad a costa de ir aumentando la cantidad de términos. Por este motivo es importante simular computacionalmente la relevancia de los términos de la ecuación dinámica. Observamos que el vector centrífugo y de coriolis  $V$  es despreciable en nuestras condiciones de trabajo y que el término más importante era el torque de gravedad. En el futuro se debe estudiar como acotar analíticamente los términos de la dinámica del robot para aplicar estrategias de control robusto.

Estudiamos distintos esquemas de control por computación de torque. Vimos que el esquema de compensación total asegura la anulación asintótica del error si no hay ruido y el sistema está perfectamente identificado. Por medio de simulaciones mostramos que este esquema es el que produce menor error. Por otro lado, nuestras simulaciones muestran que el método de compensación de gravedad genera también buenos resultados. Finalmente las simulaciones muestran que el control PID se mantiene estable para los parámetros óptimos. Pero tiene mayor error y tarda más en asegurar la convergencia.

En el Capítulo 5, mostramos experimentalmente que podemos controlar una articulación con gravedad utilizando un controlador PID programado en un microcontrolador de bajo costo. También vimos que nuestro modelo predice satisfactoriamente el comportamiento del controlador. Existen diversos métodos que permiten implementar controles más complejos que un PID, como los que se estudian en el Capítulo 4.

A futuro nos proponemos seguir los desarrollos trazados en este trabajo a un número mayor de grados de libertad.



# Apéndice A

## Software

```
clear , clc
q1=sym('q1','real');    q2=sym('q2','real');
q1p=sym('q1p','real');  q2p=sym('q2p','real');
q1pp=sym('q1pp','real');q2pp=sym('q2pp','real');
L1=sym('L1','real');    L2=sym('L2','real');
lc1=sym('lc1','real');  lc2=sym('lc2','real');
m1=sym('m1','real');    m2=sym('m2','real');
Ixx1=sym('Ixx1','real');Ixx2=sym('Ixx2','real');
Iyy1=sym('Iyy1','real');Iyy2=sym('Iyy2','real');
Izz1=sym('Izz1','real');Izz2=sym('Izz2','real');
g=sym('g','real');

% Vector de coordenadas generalizadas
vq=[q1,q2];
% Vector de velocidades generalizadas
vqp=[q1p;q2p];
% Vector de aceleraciones generalizadas
vqpp=[q1pp;q2pp];

T0_1=[cos(q1) -sin(q1)  0  0;...
```

```

        sin(q1)   cos(q1)   0   0;...
        0         0         1   0;...
        0         0         0   1];

T1_2=[cos(q2)  -sin(q2)  0   L1;...
      sin(q2)   cos(q2)  0   0;...
      0         0         1   0;...
      0         0         0   1];

T0_2=T0_1*T1_2;
T0_2=simplify(T0_2);

% Posicion del centro de masa de cada link respecto a su trama
pc1_1=[lc1 ;0;0;1];
pc2_2=[lc2 ;0;0;1];

% Calculo pc1 y pc2 en trama {0}
pc1_0=T0_1*pc1_1;
pc2_0=T0_2*pc2_2;

% Calculo jacobianos en trama {0} para las velocidades de traslacion
jac0_vc1=[diff(pc1_0(1),q1), diff(pc1_0(1),q2);...
          diff(pc1_0(2),q1), diff(pc1_0(2),q2);...
          diff(pc1_0(3),q1), diff(pc1_0(3),q2)];

jac0_vc2=[diff(pc2_0(1),q1), diff(pc2_0(1),q2);...
          diff(pc2_0(2),q1), diff(pc2_0(2),q2);...
          diff(pc2_0(3),q1), diff(pc2_0(3),q2)];

% Calculo jacobianos en trama {0} para las velocidades angulares

R0_1=T0_1(1:3,1:3);
R0_2=T0_2(1:3,1:3);

```

---

```

jac0_w1=[R0_1*[0;0;1],R0_2*[0;0;0]];
jac0_w2=[R0_1*[0;0;1],R0_2*[0;0;1]];

% Tensor de inercia
I1cm=[Ixx1,0,0;0,Iyy1,0;0,0,Izz1];
I2cm=[Ixx2,0,0;0,Iyy2,0;0,0,Izz2];

% Matriz M
M=simplify(m1*jac0_vc1'*jac0_vc1+m2*jac0_vc2'*jac0_vc2+...
    jac0_w1'*I1cm*jac0_w1+jac0_w2'*I2cm*jac0_w2);

% Energía Cinética
K=1/2*simplify([q1p;q2p]'*M*[q1p;q2p]);

% Energia potencial gravitatoria
vg=[0,-g,0,0];
U=simplify(-m1*vg*pc1_0-m2*vg*pc2_0);

% Vector G
G=-(jac0_vc1'*m1*[0;-g;0]+jac0_vc2'*m2*[0;-g;0]);

% Lagrangiano
L=simplify(K-U);

% Calculo los simbolos de Christoffel
for i=1:2
    for j=1:2
        for k=1:2

            C(i,j,k)=1/2*(diff(M(k,j),vq(i))+...
                diff(M(k,i),vq(j))-diff(M(i,j),vq(k)));
        end
    end
end

```

```
    end
end

% Calculo los términos de izquierda en las ecuaciones de Lagrange
for k=1:2
    term1(k)=M(k,:)*vqpp;
    term2(k)=vqp'*C(:, :, k)*vqp;
    term3(k)=G(k);
end

for k=1:2
    izq(k)=simplify(term1(k)+term2(k)+term3(k));
end

M,C,G
```



# Bibliografía

- [1] S. Mehta and T. Burks, “Vision-based control of robotic manipulator for citrus harvesting,” *Computers and Electronics in Agriculture*, vol. 102, pp. 146–158, 2014.
- [2] N. Wang, N. Zhang, and M. Wang, “Wireless sensors in agriculture and food industry recent development and future perspective,” *Computers and electronics in agriculture*, vol. 50, no. 1, pp. 1–14, 2006.
- [3] P.-C. Tung, M.-C. Wu, and Y.-R. Hwang, “An image-guided mobile robotic welding system for smaw repair processes,” *International Journal of Machine Tools and Manufacture*, vol. 44, no. 11, pp. 1223–1233, 2004.
- [4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*, vol. 3. Wiley New York, 2006.
- [5] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson/-Prentice Hall Upper Saddle River, NJ, USA:, 2005.
- [6] V. A. Deshpande and P. George, “Analytical solution for inverse kinematics of scorbot-er-vplus robot,”
- [7] A. Verma and V. A. Deshpande, “End-effector position analysis of scorbot-er vplus robot,” *International Journal of Smart Home*, vol. 5, no. 1, 2011.
- [8] H. Chaudhary, R. Prasad, and N. Sukavanum, “Position analysis based approach for trajectory tracking control of scorbot-er-v plus robot manipulator,” *International Journal of Advances in Engineering & Technology*, vol. 3, no. 2, pp. 253–264, 2012.

- [9] H. Chaudhary and R. Prasad, “intelligent inverse kinematic control of scorbot-er v plus robot manipulator,” 2011.
- [10] P. Corke, “A robotics toolbox for matlab,” *Robotics & Automation Magazine, IEEE*, vol. 3, no. 1, pp. 24–32, 1996.
- [11] E. Robotec, “Scorbot-er vplus manual de usuario,”
- [12] T. Krajnák, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, “A practical multirobot localization system,” *Journal of Intelligent and Robotic Systems*, 2014.
- [13] R. Paul and H. Zong, “Robot motion trajectory specification and generation,” in *2nd International Symposium on Robotics Research, Kyoto, Japan*, 1984.
- [14] R. H. Taylor, “Planning and execution of straight line manipulator trajectories,” *IBM Journal of Research and Development*, vol. 23, no. 4, pp. 424–436, 1979.
- [15] R. P. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.
- [16] D. E. Orin and W. Schrader, “Efficient jacobian determination for robot manipulators,” in *Robotics Research: The First International Symposium*, pp. 727–734, MIT Press, 1984.
- [17] F. MONASTERIO and Á. GUTIERREZ, “Modelo lineal de un motor de corriente continua,” 2012.
- [18] L. T. Dulger and S. Uyan, “Modelling, simulation and control of a four-bar mechanism with a brushless servo motor,” *Mechatronics*, vol. 7, no. 4, pp. 369–383, 1997.
- [19] C. C. De Wit, H. Olsson, K. Astrom, and P. Lischinsky, “Dynamic friction models and control design,” in *American Control Conference, 1993*, pp. 1920–1926, IEEE, 1993.

- [20] S. Cong, G. Li, and X. Feng, "Parameters identification of nonlinear dc motor model using compound evolution algorithms," in *Proceedings of the World Congress on Engineering*, vol. 1, 2010.
- [21] T. S. Hsia, T. Lasky, and Z. Guo, "Robust independent joint controller design for industrial robot manipulators," *Industrial Electronics, IEEE Transactions on*, vol. 38, no. 1, pp. 21–25, 1991.
- [22] J.-J. E. Slotine, "The robust control of robot manipulators," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 49–64, 1985.
- [23] M. Zhihong, A. Paplinski, and H. Wu, "A robust mimo terminal sliding mode control scheme for rigid robotic manipulators," *Automatic Control, IEEE Transactions on*, vol. 39, no. 12, pp. 2464–2469, 1994.
- [24] M. S. Kang, "Robust digital friction compensation," *Control Engineering Practice*, vol. 6, no. 3, pp. 359–367, 1998.
- [25] E. Castillo-Castaneda and Y. Takeda, "Improving path accuracy of a crank-type 6-dof parallel mechanism by stiction compensation," *Mechanism and Machine Theory*, vol. 43, no. 1, pp. 104–114, 2008.
- [26] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942.
- [27] L. Landau, "Mechanics: Volume 1 (course of theoretical physics) author: Ld landau, em lifshitz, publisher: Butterwor," 1976.



# Agradecimientos

A mis papás y hermanos, que son los pilares fundamentales en mi vida y más.

A Lau, que me acompaña todos los días y me brinda su apoyo incondicional.

A Damían por tanto trabajo y la buena predisposición en todo momento.

A Pablo, Maty, Taihú y Thomas (y el resto de los chicos/a del labo) que compartimos tantos buenos años juntos.

A los amigos, que siempre están ahí.

A los chicos de escalada, que son manija por lo físico.

Al borracho de Alan, que a pesar de vaciarme la heladera siempre, hace cafés ricos.

Facundo Pessacg Buenos Aires, Marzo de 2015

