

Robótica

Tesis Final

Estudio de la dinámica de un robot con 6 grados de libertad

Alumnos: Lopreto, Sebastián

Moras, Pablo

Rubstein, Jonathan

Profesor: Mas. Ing. Giannetta, Hernán

Curso: R-6055

Fecha de entrega: 10 de agosto de 2009

Índice

Introducción a la dinámica del robot.....	3
Introducción al diseño del control dinámico del robot.....	10
Desarrollo del diseño dinámico de Titán.....	25
Desarrollo del diseño del control dinámico de Titán.....	35
Implementación del control dinámico de Titán.....	36
Simulación del control dinámico de Titán.....	38
Simulación utilizando RoboWorks.....	40
Conclusiones.....	41
Referencias.....	42

Introducción a la Dinámica del Robot

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. Por lo tanto, el modelo dinámico de un robot tiene por objeto conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo.

Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

1. La localización del robot definida por sus variables articulares y sus derivas: velocidad y aceleración.
2. Las fuerzas y pares aplicadas en las articulaciones.
3. Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

La obtención del modelo, aumenta de complejidad al aumentar los grados de libertad. Por este motivo no siempre es posible obtener un modelo dinámico expresado en una forma cerrada, esto es, mediante una serie de ecuaciones, generalmente del tipo diferencial de 2º orden, cuya integración permita conocer qué movimiento surge al aplicar una fuerza. El modelo dinámico debe ser resuelto de manera iterativa mediante la utilización de un procesamiento numérico.

El problema de la obtención del modelo dinámico del robot es, por lo tanto, uno de los aspectos más complejos de la robótica que nos permite conseguir los siguientes fines:

1. Simulación del movimiento del robot.
2. Diseño y evaluación de la estructura mecánica del robot.
3. Dimensionamiento de los actuadores.
4. Diseño y evaluación del control dinámico del robot.

Es importante hacer notar que el modelo dinámico completo de un robot debe incluir no sólo la dinámica de sus elementos sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando.

A continuación vamos a desarrollar los métodos más utilizados considerando a los robots como rígidos.

Modelo dinámico de la estructura mecánica de un robot rígido

Este modelo se basa fundamentalmente en el planteamiento del equilibrio de fuerzas establecido en la segunda ley de Newton, o en su equivalente de rotación, la ley de Euler:

$$\sum F = m\dot{v}$$
$$\sum T = I\dot{\omega} + \omega \times (I\omega)$$

De este planteamiento del equilibrio de fuerzas y pares que intervienen sobre el robot se obtienen los denominados modelos dinámicos directo e inverso:

- Modelo dinámico directo: expresa la evolución temporal de las coordenadas articulares del robot en función de las fuerzas y pares que intervienen.
- Modelo dinámico inverso: expresa las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares y sus derivadas.

En el planteamiento del equilibrio de fuerzas en un robot real de 5 ó 6 grados de libertad, debe tenerse en cuenta que junto con las fuerzas de inercia y gravedad, aparecen fuerzas de Coriolis debidas al movimiento relativo existente entre los diversos elementos, así como de fuerzas centrípetas que dependen de la configuración instantánea del manipulador.

Un planteamiento alternativo para la obtención del modelo puede ser usar la formulación Lagrangiana, basada en consideraciones energéticas. Este método es más sistemático y facilita enormemente la formulación de un modelo.

Lagrange estable la ecuación:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau$$
$$L = k - \mu$$

Con:

q_i : coordenadas generalizadas

τ : vector de fuerzas y pares aplicados a las q_i

L: función lagrangiana

k: energía cinética

μ : energía potencial

Obtención del modelo dinámico de un robot mediante la formulación de Lagrange – Euler

Este planteamiento utiliza, por tanto, las matrices ${}^{i-1}A_i$ que relacionan el sistema de coordenadas de referencia del elemento i con el del elemento i-1. Se realizan en este caso operaciones de producto y suma innecesarias. Se trata de un procedimiento ineficiente desde el punto de vista computacional, donde tiene una complejidad de $O(n^4)$. Sin embargo, conduce a unas ecuaciones finales bien estructuradas donde aparecen de manera clara los diversos pares y fuerzas que intervienen en el movimiento.

Algoritmo computacional para el modelo dinámico por Lagrange – Euler

1. Asignar a cada eslabón un sistema de referencia de acuerdo a las normas D-H.

2. Obtener las matrices de transformación 0A_i para cada elemento i.
3. Obtener las matrices U_{ij} definidas por:

$$U_{ij} = \frac{\partial}{\partial q_j} {}^0A_i$$

4. Obtener las matrices U_{ijk} definidas por:

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k}$$

5. Obtener las matrices de pseudoinercias J_i para cada elemento, que vienen definidas por:

$$U_{ijk} = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int y_i x_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int z_i x_i dm & \int z_i y_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix}$$

Donde las integrales están extendidas al elemento i considerado, y (x_i y_i z_i) son las coordenadas del diferencial de masa dm respecto al sistema de coordenadas del elemento.

6. Obtener la matriz de inercias $D=[d_{ij}]$ cuyos elementos vienen definidos por:

$$d_{ij} = \sum_{k=(\max i, j)}^n \text{Traza}(U_{kj} J_k U_{ki}^T)$$

Con $i, j = 1, 2, \dots, n$

n : número de grados de libertad

7. Obtener los términos h_{ikm} por:

$$h_{ikm} = \sum_{j=(\max i, k, m)}^n \text{Traza}(U_{jkm} J_j U_{ji}^T)$$

Con $i, k, m = 1, 2, \dots, n$

8. Obtener la matriz columna de fuerzas de Coriolis y centrípeta $H = [h_i]^T$ cuyos elementos viene definidos por:

$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m$$

9. Obtener la matriz columna de fuerzas de gravedad $C = [c_i]^T$ cuyos elementos viene definidos por:

$$c_i = \sum_{j=1}^n (-m_j g U_{ji}^j r_j)$$

con $i = 1, 2, \dots, n$

g : es el vector de gravedad expresado en el sistema de la base $\{S_0\}$ y viene expresado en $(g_{x0}, g_{y0}, g_{z0}, 0)$

r_j : es el vector de coordenadas homogéneas del centro de masa del elemento j expresado en el sistema de referencia del elemento i .

10. La ecuación dinámica del sistema será:

$$\tau = D\ddot{q} + H + C$$

Donde τ es el vector de fuerzas y pares motores efectivos aplicados sobre cada coordenada q_i .

Modelo dinámico usando Newton-Euler

El modelo dinámico obtenido por el método de Lagrange nos lleva a un algoritmo sumamente complejo a nivel computacional ($O(n^4)$ siendo n el número de grados de libertad). Lo antes mencionado se puede comprender como que a mayor número de grados de libertad el algoritmo se vuelve inutilizable para aplicaciones en tiempo real.

La ecuación de Newton-Euler parte del equilibrio de fuerzas y pares como la segunda ley de Newton:

$$\sum F = m \frac{\partial v}{\partial t} \quad \sum T = I \frac{\partial \omega}{\partial t} + \omega \cdot (I \omega)$$

Siendo:

F la fuerza sobre la barra.

T los pares ejercidos sobre la barra.

m masa de la barra

I tensor de inercia de la barra entorno a su centro de masa expresado en el sistema de referencia.

v y ω son las velocidades lineales y angulares de las articulaciones.

La fórmula anterior se plantea para cada articulación en particular. La mejor forma de implementar este método es mediante una formulación recursiva, lo cual posibilita la obtención de la posición, velocidad y aceleración del eslabón i con respecto a la base a partir de los correspondientes del eslabón $i-1$ y del movimiento relativo de la articulación. Realizando lo antes dicho empezando del eslabón 1 se llega rápidamente al n . Con los datos hallados anteriormente se obtienen las fuerzas y pares que aparecen sobre el eslabón i referidos nuevamente a la base del robot a partir de los correspondientes al eslabón $i+1$, recorriendo de esta forma todos los eslabones desde el n al primero, es decir de manera inversa.

El método utiliza operaciones vectoriales, las cuales son más eficientes, a nivel computacional, que las operaciones matriciales utilizadas por la formulación de Lagrange. Se puede observar que el método reduce mucho la complejidad desde $O(n^4)$ a $O(n)$ (siendo n el número de grados de libertad).

Los pasos para realizar un análisis mediante el método de Newton-Euler son los siguientes:

1. Asignar a cada eslabón un sistema de referencia teniendo en cuenta las normas D-H.
2. Obtener las matrices de rotación:

$${}^{i-1}R = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i \\ S\theta_i & C\alpha_i S\theta_i & -S\alpha_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i \end{bmatrix}$$

3. Establecer las condiciones iniciales (tomando como sistema base el $\{S_0\}$):

$${}^0\omega : \text{velocidad angular} = [0 \ 0 \ 0]^T$$

$${}^0 \frac{\partial \omega}{\partial t} : \text{aceleración angular} = [0 \ 0 \ 0]^T$$

$${}^0v : \text{velocidad lineal} = [0 \ 0 \ 0]^T$$

$${}^0 \frac{\partial v}{\partial t} : \text{velocidad angular} = [g_x \ g_y \ g_z]^T$$

$${}^i z_0 = [0 \ 0 \ 1]^T$$

$${}^i p_i : \text{coordenadas del origen del sistema } s\{i\} \text{ respecto a } s\{i-1\} = [a_i \ d_i S_i \ d_i C_i]$$

$${}^i S_i : \text{coordenadas del centro de masa del eslabón } i \text{ respecto del sistema } \{S_i\}.$$

$${}^i I_i : \text{Matriz de inercia del eslabón } i \text{ respecto de su centro de masa expresado en } \{S_i\}$$

Para $i = 1$ hasta n hay que realizar los pasos 4 al 7.

4. Obtener la velocidad angular del sistema:

$${}^i \omega_i = {}^i R_{i-1} ({}^{i-1} \omega_{i-1} + z_0 \frac{\partial q_i}{\partial t}) \text{ Si el eslabón es de rotación}$$

$${}^i R_{i-1} {}^{i-1} \omega_{i-1} \text{ Si el eslabón es de traslación}$$

5. Obtener la aceleración angular del sistema:

$$\frac{\partial^i \omega_i}{\partial t} = {}^i R_{i-1} \left(\frac{\partial^{i-1} \omega_{i-1}}{\partial t} + z_0 \frac{\partial^2 q_i}{\partial t^2} \right) + {}^{i-1} \omega_{i-1} x z_0 \frac{\partial q_i}{\partial t} \quad \text{Si el eslabón es de rotación}$$

$${}^i R_{i-1} \frac{\partial^{i-1} \omega_{i-1}}{\partial t} \quad \text{Si el eslabón es de traslación}$$

6. Obtener la aceleración lineal del sistema:

$$\frac{\partial^i v_i}{\partial t} = {}^i R_{i-1} \frac{\partial^{i-1} \omega_{i-1}}{\partial t} + \left(\frac{\partial^i \omega_i}{\partial t} x^i p_i \right) + {}^i \omega_i x ({}^i \omega_i x^i p_i) \quad \text{Si el eslabón es de rotación}$$

$${}^i R_{i-1} \left(z_0 \frac{\partial^2 q_i}{\partial t^2} + \frac{\partial^{i-1} v_{i-1}}{\partial t} \right) + \left(\frac{\partial^i \omega_i}{\partial t} x^i p_i \right) + 2 {}^i \omega_i x {}^i R_{i-1} z_0 \frac{\partial q_i}{\partial t} + {}^i \omega_i x ({}^i \omega_i x^i p_i) \quad \text{Si el eslabón es de traslación}$$

7. Obtener la aceleración lineal del centro de gravedad del eslabón i:

$${}^i a_i = \frac{\partial^i \omega_i}{\partial t} x^i S_i + {}^i \omega_i x ({}^i \omega_i x^i S_i) + \frac{\partial^i v_i}{\partial t}$$

Para i = n hasta 1 hay que realizar los pasos 8 al 10.

8. Obtener la fuerza ejercida sobre el eslabón i:

$${}^i f_i = {}^i R_{i+1} {}^{i+1} f_{i+1} + {}^i a_i m_i$$

9. Obtener el par ejercida sobre el eslabón i:

$${}^i n_i = {}^i R_{i+1} \left[{}^{i+1} n_{i+1} + ({}^i R_{i+1} {}^i p_i) x^{i+1} f_{i+1} \right] + ({}^i p_i {}^i S_i) x^i a_i m_i + {}^i I_i \frac{d^i \omega_i}{dx} + {}^i \omega_i x ({}^i I_i {}^i \omega_i)$$

10. Obtener la fuerza o par aplicado a la articulación i.

$$\tau = {}^i n_i^T {}^i R_{i-1} z_0$$

$${}^i f_i^T {}^i R_{i-1} z_0$$

Donde τ es el par o fuerza efectivo (par motor menos pares de rozamiento o perturbación).

Modelo dinámico usando variables de estado

La siguiente ecuación define el modelo dinámico inverso de un robot, indicando los pares y fuerzas que deben realizar los actuadores para que las variables articulares realizan una trayectoria en particular $q(t)$:

$$\tau = D \frac{\partial^2 q}{\partial t^2} + H + C$$

En la formula anterior es necesario recordar que la matriz de inercias D y la matriz de columna C dependen de q y que la matriz de fuerzas de Coriolis y centrípetas H depende

De q y su derivada. Se debe considerar el vector de pares generalizados τ :

$$\tau = \tau_{motor} - \tau_{perturbador} - \tau_{rozamiento\ viscoso} - \tau_{rozamiento\ seco}$$

La expresión anterior es por lo tanto no lineal. Debido a lo antes mencionado se puede obtener el modelo dinámico que proporciona la trayectoria realizada como consecuencia de la aplicación de unos pares determinados.

Para obtener el modelo de estados del sistema, la mejor elección de variables de estados es usar la posición y la velocidad de las articulaciones. Realizando cálculos matriciales se obtiene la matriz final del modelo.

$$\frac{d}{dt} \begin{bmatrix} q \\ \frac{dq}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \frac{dq}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$u = D^{-1}(\tau - N)$$

Introducción al Diseño del Control Dinámico del Robot

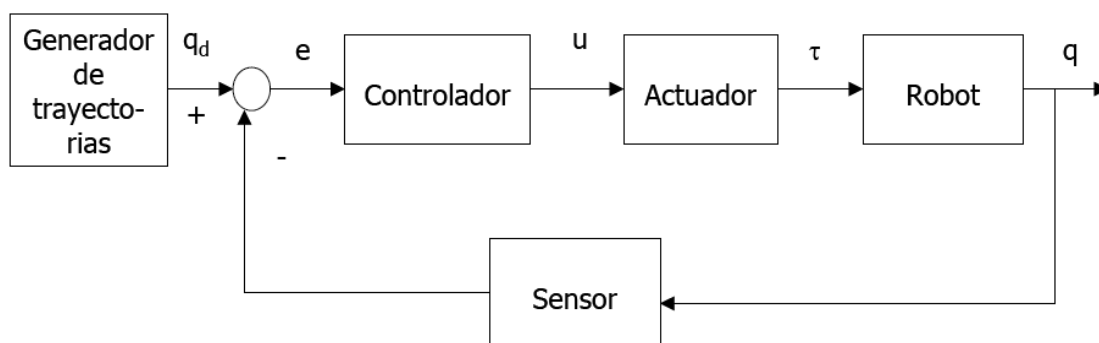
El control dinámico tiene por misión procurar que las trayectorias realmente seguidas por el robot sean lo mas parecidas posibles a las propuestas por el control cinemático. Para ellos hace uso del conocimiento del modelo dinámico del robot y de las herramientas de análisis y diseño aportadas por la teoría del servocontrol (representación interna, estado, estabilidad de Lyapunov, control PID, control adaptativo, etc.)

El modelo dinámico de un robot es fuertemente no lineal, multivariable, acoplado y de parámetros variantes, por lo que en general su control es extremadamente complejo. En la práctica ciertas simplificaciones, validas para un gran número de los robots comerciales existentes, facilitan el diseño del sistema de control, dando unos resultados razonablemente aceptables, aunque limitando en ciertas situaciones la calidad de sus prestaciones.

La utilización de técnicas de control más potentes, que podrían dar mejores resultados, pueden verse limitadas en la práctica por la elevada complejidad y coste computacional, que si bien es abordable con la tecnología actual, supone un mayor coste económico.

A continuación mencionamos algunas de las técnicas de control dinámico mas frecuentes:

Sistema de Control



- Actuador: motor de corriente continua
- q : posición de la articulación
- q_d : posición deseada de la articulación (generador de trayectorias)
- e : error = $q_d - q$

Control Dinámico

- Objetivo: Que la trayectoria seguida por el robot $[q(t)]$ sea lo más parecida a la propuesta por la planificación de trayectorias $[q_d(t)]$
- Solución inicial: Utilizar los modelos dinámicos del robot (Lagrange-Euler, Newton-Euler...)
- Inconveniente de los modelos dinámicos:
 - No-lineales
 - Multivariables
 - Acoplados
 - Parámetros variantes en el tiempo
- El control basado en los modelos dinámicos es complejo: mucho coste computacional.
- Solución práctica: Trabajar con un modelo para cada articulación por separado (aprovechando el factor de reducción para desacoplar las articulaciones) => **Control monoarticular**.
- Reductores: si la articulación presenta reductores (o engranajes) se reduce la velocidad del movimiento y se disminuye el acoplamiento entre ejes.
- Con reductores, la posición de las articulaciones $[q(t)]$ está relacionada con la posición del actuador: $[q_a(t)]$:

$$q_a = K q$$

- y K es una matriz diagonal con los factores de reducción $k_{ii} > 1$
- Consecuentemente se cumple que:

$$\dot{q}_a = K \dot{q} \quad \ddot{q}_a = K \ddot{q} \quad \tau_a = K^{-1} \tau$$

- El modelo dinámico del robot (Lagrange-Euler) en presencia de reductores:

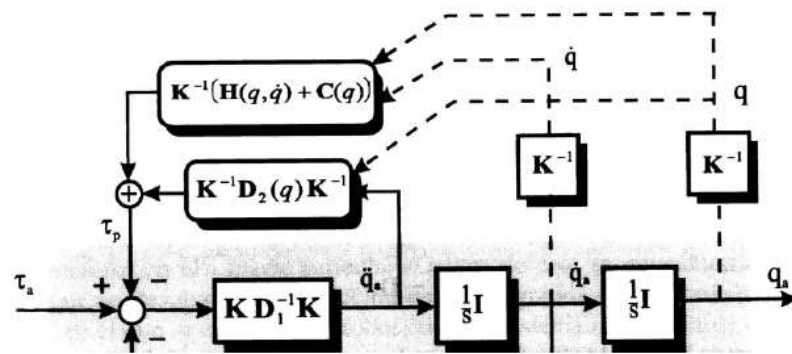
$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + c(q)$$

$$\tau_a = K^{-1} D K^{-1} \ddot{q}_a + K^{-1} h + K^{-1} c$$

- Si se separa en D_1 los términos constantes de D en la diagonal y en D_2 el resto de los términos:

$$\tau_a = K^{-1} D_1 K^{-1} \ddot{q}_a + \underbrace{K^{-1} D_2 K^{-1} \ddot{q}_a + K^{-1} h + K^{-1} c}_{\tau_p}$$

- Con τ_p es el par perturbador (el generado por el movimiento del resto de las articulaciones).



- Si los factores de reducción son suficientemente grandes: $\tau_p \ll \tau_a$, con lo que puede considerarse el movimiento de cada articulación desacoplado del movimiento del resto de las articulaciones
- ¿Cuándo puede plantearse el control separado de las articulaciones?:
 - Sólo para robots de accionamiento indirecto con reductores (o engranajes de elevada relación).
- Inconvenientes de los engranajes: Introducción de holguras en la transmisión, aumento del rozamiento.
- Ventajas: desacoplan al sistema, reducen la velocidad de las articulaciones, simplifican el modelo dinámico del robot.

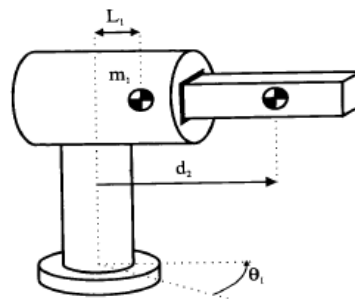
- Ejemplo:

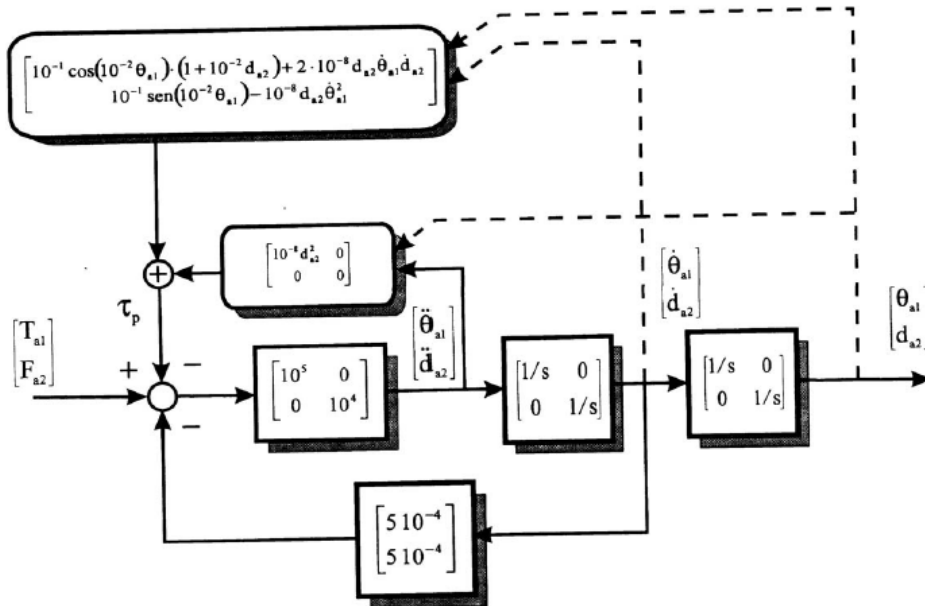
$$K_{11} = k_{22} = 100$$

$m_1=10, m_2=1$

$L1=0.1$

$g=10$

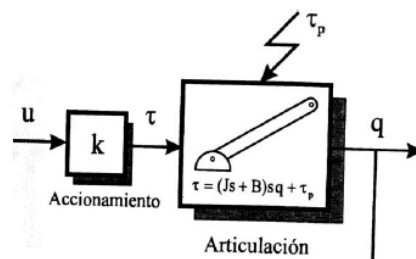
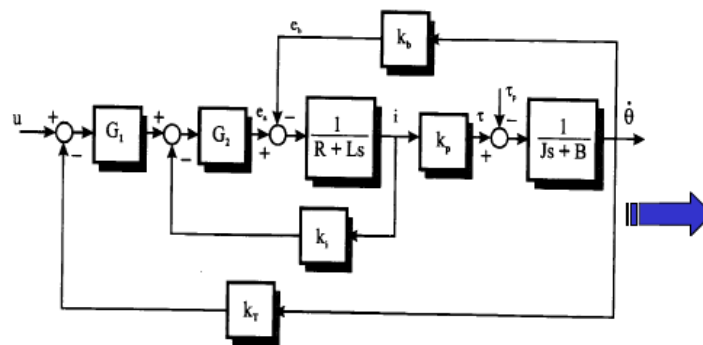




Actuador

- Diseño del controlador: utilizando el modelo dinámico del actuador:

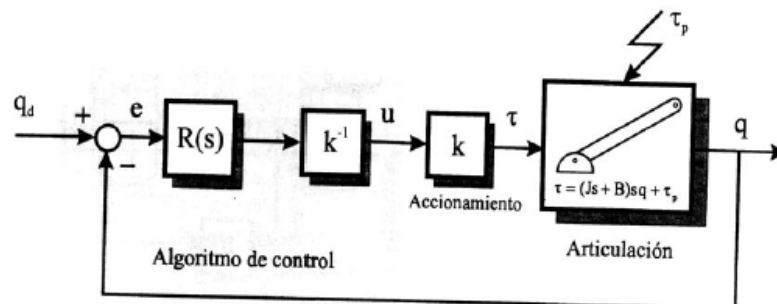
Motor eléctrico de corriente continua



- Modelo simplificado del actuador: el accionamiento eléctrico se supone suficientemente rápido como para aproximarlos por una ganancia constante: $\tau = Ku$.

CONTROL PID

- Se realimenta la diferencia entre la posición de la coordenada generalizada deseada y la real



- Control más frecuente: PID

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad \rightarrow$$

$$u = k^{-1} R(s) e(s) = k^{-1} \left(k_p + \frac{k_i}{s} + k_d s \right) e(s)$$

$$\left. \begin{array}{l} \tau = (J s + B) s q + \tau_p \\ \tau = k k^{-1} R(s) e \end{array} \right\} \rightarrow$$

$$e(s) = \frac{\tau_p + (J s + B) s q_d}{R(s) + (J s + B) s}$$

- Si no hay cambio en la referencia: $q_d = 0$

$$e(s) = \frac{\tau_p}{R(s) + (J s + B) s} \neq 0$$

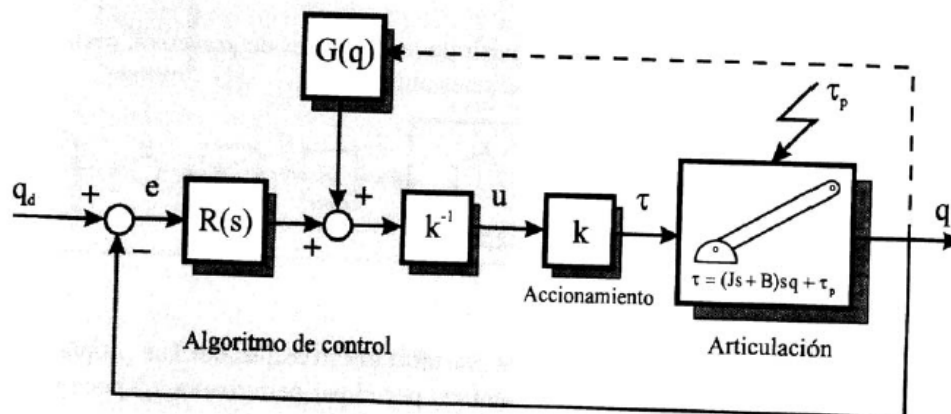
- Si no hay perturbaciones: $\tau_p = 0$

$$e(s) = \frac{(Js+B)sq_d}{R(s) + (Js+B)s} \neq 0$$

- El controlador PID tiene que actuar en ambos casos para hacer el error igual a 0.

CONTROL PID CON COMPENSACION DE LA GRAVEDAD

- De los términos que generan el par perturbador el más importante suele ser el de la gravedad (el único en el estacionario)
- Con el modelo dinámico puede eliminarse su efecto: $G(q)$



- $u = K^{-1} [R(s) e(s) + G(q)]$

$$\left. \begin{aligned} \tau &= K u = R(s) e(s) + G(q) \\ \tau &= (Js + B) s q + \tau_p \end{aligned} \right\} \rightarrow$$

$$e(s) = \frac{(Js+B)sq_d + \tau_p - G(q)}{R(s) + (Js+B)s}$$

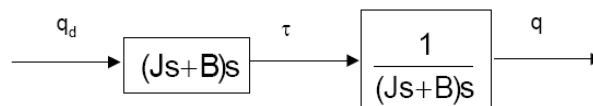
- Si no hay cambio en la referencia: $q_d = 0$

$$e(s) = \frac{\tau_p - G(q)}{R(s) + (Js+B)s} = 0 \rightarrow G(q) = \tau_p = C(q)$$

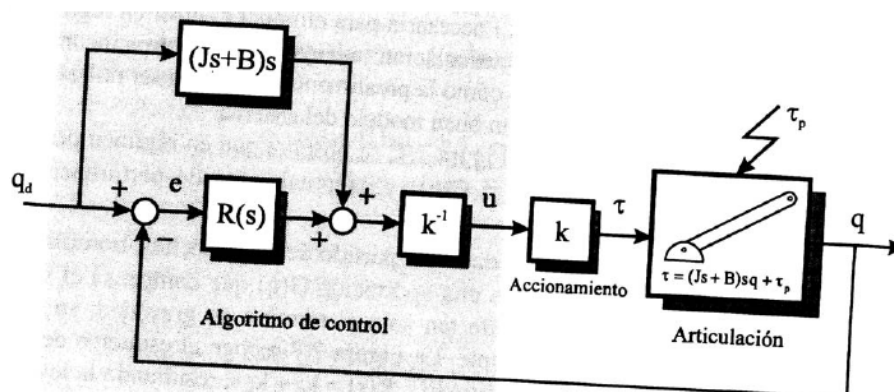
- Si no hay perturbaciones: $\tau_p = 0$

$$e(s) = \frac{(Js + B)sq_d - G(q)}{R(s) + (Js + B)s} \neq 0$$

- Si el modelo de la articulación fuera perfecto se podría diseñar por inversión del modelo un “compensador ideal”, con error de seguimiento nulo:



- Sin embargo:
 - el modelo nunca es perfecto
 - es necesario compensar el par perturbador
- Solución: se añade al modelo inverso una realimentación por PID:



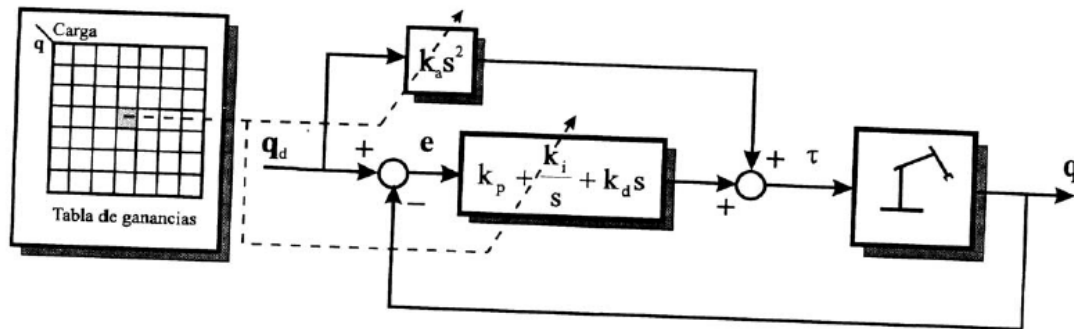
$$\left. \begin{aligned} \blacksquare u &= k^{-1} [R(s) e + (Js+B) s q_d] \\ \blacksquare \tau &= k u = [R(s) e + (Js+B) s q_d] \\ \blacksquare \tau &= (Js + B) s q + \tau_p \end{aligned} \right\}$$

$$\rightarrow e(s) = \frac{\tau_p}{R(s) + (Js + B)s}$$

- $q(t) = q_d(t) \Rightarrow$ para toda la trayectoria
- $e(s)$ solo depende de las perturbaciones

PLANIFICACION DE GANANCIAS

- Problema: los parámetros del modelo y la magnitud de los pares perturbadores a compensar cambian con el punto de trabajo del robot.
- Soluciones:
 - Ajustar un único controlador para que funcione adecuadamente en todas las circunstancias (robusto)
 - Cambiar los parámetros del controlador (de forma suave) al cambiar el punto de trabajo => planificación de ganancias.
- Planificación de ganancias: tabla precalculada de valores de ganancias con los parámetros:
 - Coordenadas generalizadas
 - Carga del robot



- Problema: gran numero de combinaciones posibles
- Solución: no utilizar como parámetros aquellas articulaciones que influyen poco en otras:
 - La muñeca
 - La cintura si es perpendicular al resto

CONTROL MULTIARTICULAR

Desacoplamiento por Inversión de Modelo

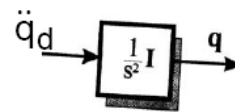
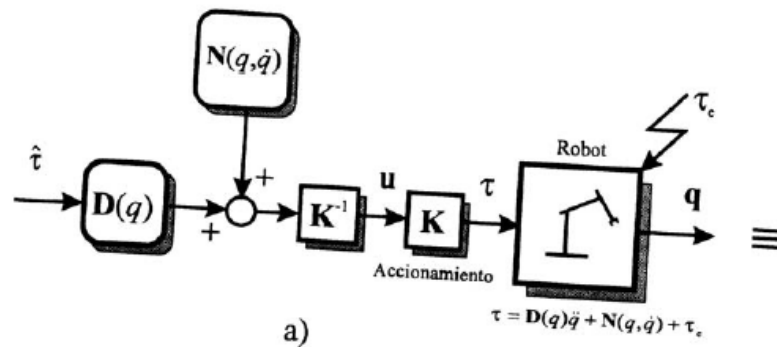
Si se considera el modelo de Lagrange-Euler:

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + c(q) = D(q)\ddot{q} + N(q, \dot{q})$$

Para desacoplar el modelo se utiliza un controlador:

$$\left. \begin{aligned} \tau &= k u = [D(q) \hat{\tau} + N(q, \dot{q})] \\ \tau &= D(q) \ddot{q} + N(q, \dot{q}) \end{aligned} \right\}$$

$$\xrightarrow{\text{red arrow}} \hat{\tau} = \ddot{q}_d$$

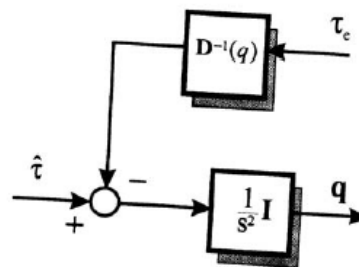
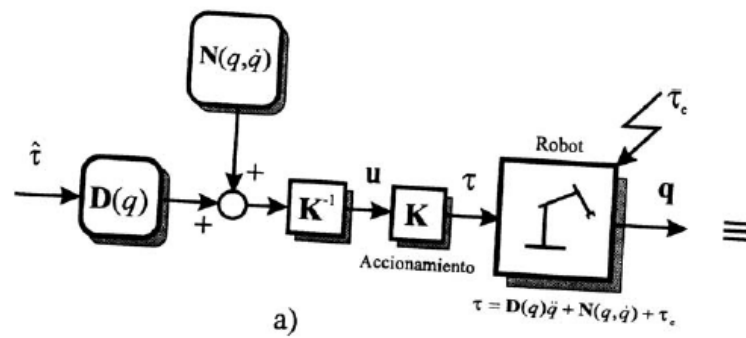


Control PID con prealimentación y desacoplado

En presencia de variaciones en el modelo y pares exteriores el desacoplamiento no es perfecto.

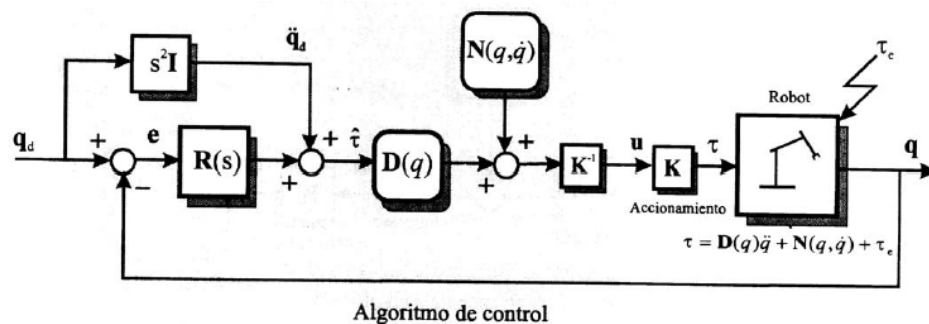
$$\left. \begin{aligned} \tau &= D(q)\ddot{q} + h(q, \dot{q}) + c(q) + \tau_e = D(q)\ddot{q} + N(q, \dot{q}) + \tau_e \\ \tau &= D(q) \hat{\tau} + N(q, \dot{q}) \end{aligned} \right\}$$

$$\begin{aligned} \hat{\tau} &= \ddot{q} - D^{-1}\tau_e \\ \xrightarrow{\text{red arrow}} \ddot{q}_d &= \ddot{q} - D^{-1}\tau_e \end{aligned}$$



Para compensar la diferencia entre \ddot{q} y \ddot{q}_d se realimenta su diferencia en cada lazo con cualquier técnica de control monoarticular.

Ej: Inversión de modelo + PID + compensación por adelanto



■ **Control por cal calculado:** $u = K^{-1}D(q)[s^2 I q_d + R(s)e(s)] + K^{-1}N(q, \dot{q})$

- 1.- $\hat{\tau} = s^2 I q_d + R(s)e(s)$
- 2.- $\tau = D(q) \hat{\tau} + N(q, \dot{q})$

⇒ $\tau = D(q)[s^2 I q_d + R(s)e(s)] + N(q, \dot{q})$

El modelo dinámico del robot es:

$$\tau = D(q)s^2Iq + N(q, \dot{q}) + \tau_e$$

Igualando ambas expresiones:

$$e = [D(q)(s^2I + R(s))]^{-1}\tau_e$$

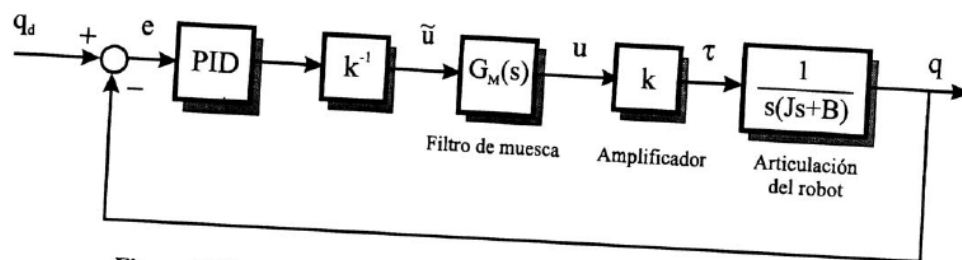


En ausencia de perturbaciones ($\tau_e=0$), $e(s)=0 \Rightarrow q=q_d$

ASPECTOS PRACTICOS DE LA IMPLEMENTACION

Controladores discretos: Si bien el diseño suele plantearse en continuo, la implementación del sistema de control se realiza en un microprocesador, con lo que el diseño continuo hay que convertirlo a discreto (con posible presencia de ZOH).

Flexión y oscilaciones estructurales: se coloca un filtro notch a la salida del regulador que elimine de la señal de control las frecuencias cercanas a la de resonancia del sistema



Cálculo de los parámetros del PID(k_p, k_i, k_d): si sólo se tiene un PD, la ecuación característica del sistema es:

$$p(s) = s^2 + \frac{B + k_d}{J}s + \frac{k_p}{J} \quad p(s) = s^2 + 2w_n \xi s + w_n^2 = 0$$

$$\omega_n = \sqrt{\frac{k_p}{J}} \quad \xi = \frac{B + k_d}{2\sqrt{k_p J}}$$

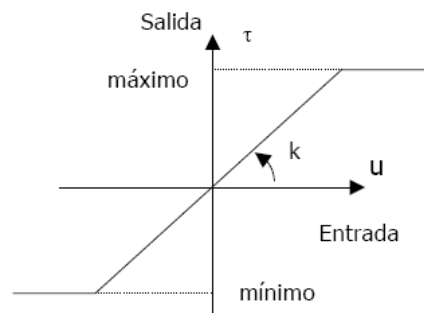
La estructura del robot no es rígida => tiene una frecuencia de resonancia w_r que se debe evitar =>

$$\omega_n < \frac{w_r}{2} \Rightarrow 2\sqrt{\frac{k_p}{J}} < w_r$$

El sistema tiene que ser críticamente amortiguado, porque si tiene oscilaciones puede chocar =>

$$\xi = 1 \Rightarrow \frac{B + k_d}{2\sqrt{k_p J}} = 1$$

Saturación de los actuadores: windup. Los dispositivos electrónicos destinados a suministrar la potencia a los actuadores del robot tienen un margen de actuación limitado.

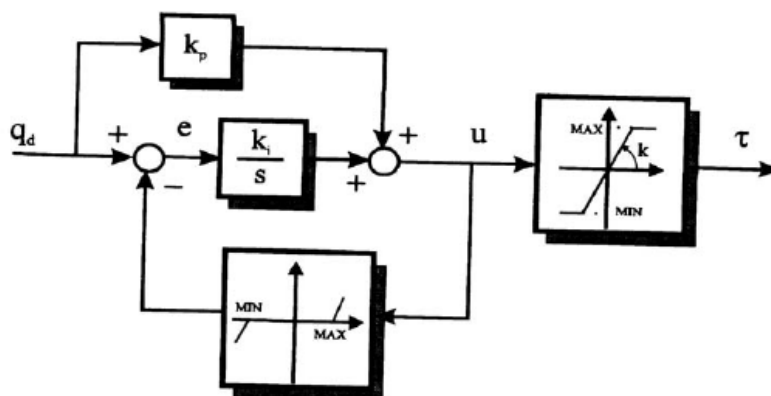


La presencia de errores mantenidos junto con la acción integral, puede hacer saturarse al controlador => la señal de control es constante independientemente del error. => Acción wind-up genera oscilaciones y se tarda más tiempo en estabilizarse.

Soluciones:

- Cuando se llegue a la saturación de los actuadores, anular la acción integral hasta que el error $e = q_d - q$ no entre dentro de los límites establecidos.

- Regulador PI con anti-windup:



- Otras formas de colocar el PID: I-PD, etc.

CONTROL ADAPTATIVO

Como se ha visto anteriormente, si se conoce perfectamente el modelo dinámico del robot, es posible utilizar la técnica de desacoplamiento por inversión del modelo para, en ausencia de perturbación, conseguir que la trayectoria real coincida exactamente con la deseada.

Sin embargo, en la práctica esta técnica tiene una aplicación limitada, bien porque los parámetros no son conocidos con precisión, o bien porque el cálculo en tiempo real del modelo es excesivamente lento.

La técnica del control adaptativo se puede aplicar con buenos resultados en aquellas ocasiones en las que el modelo, aun siendo conocido, cambia continuamente por variar las condiciones de funcionamiento. La idea básica del control adaptativo es modificar en tiempo real los parámetros que definen el regulador de acuerdo al comportamiento instantáneo del sistema.

Muchos robots industriales tienen un único controlador, ajustado para unas condiciones intermedias de funcionamiento y asegurando que los movimientos del robot se mantienen estables en otras condiciones, aunque con calidad disminuida.

Existen diferentes esquemas de control adaptativo. En todos ellos se utilizan procedimientos de identificación de los parámetros del modelo del sistema y algoritmos de cálculo de los parámetros del regulador.

Control adaptativo por planificación de ganancias

En este método, una vez seleccionado el tipo de control a utilizar, se ajustan los parámetros del mismo para un número determinado de condiciones de funcionamiento.

Las variables que hacen que estas condiciones de funcionamiento se modifiquen son la carga que manipula el robot y los valores de las coordenadas articulares. Durante el funcionamiento, los valores de las coordenadas articulares son conocidas por la unidad de control del robot, mientras que el valor de la carga puede ser conocido por el programa del robot (permite la introducción del valor) o por un sensor en el manipulador.

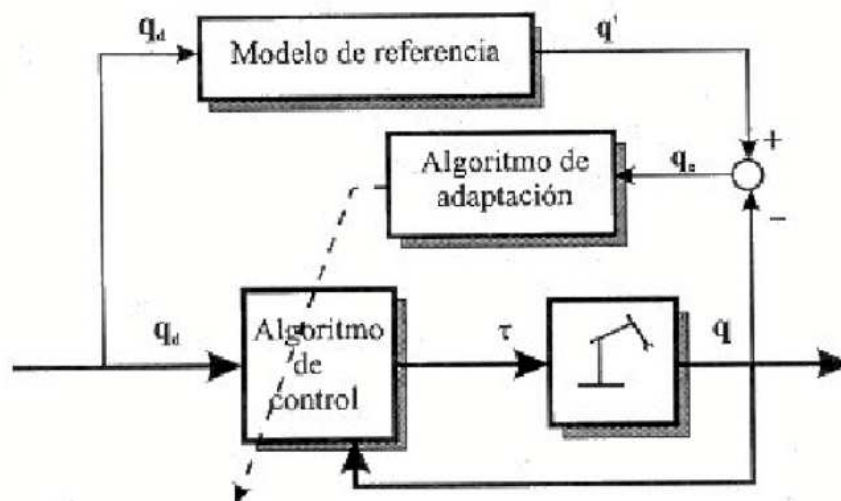
Con estos datos, el algoritmo de planificación de ganancias hace uso de una tabla en la que están especificados los parámetros de los reguladores a utilizar para cada uno de los grados de libertad. En el caso de que el punto de funcionamiento no coincida exactamente con las entradas de la tabla, puede utilizarse alguna técnica de interpolación.

Un problema con respecto a este método, es que al aumentar los grados de libertad, por ejemplo 6 grados, se tendría hasta 7 variables para modificar y establecer las diferentes condiciones de trabajo.

Un segundo problema a considerar es que si bien para los puntos de funcionamiento deseado se tiene asegurada la estabilidad y buen funcionamiento del sistema, para los puntos intermedios, no está garantizada ni siquiera la estabilidad.

Control adaptativo con modelo de referencia (MRAC)

En este tipo de control adaptativo se parte de un comportamiento ideal del sistema, definido por un modelo de referencia. La diferencia instantánea entre el comportamiento real del robot q y el modelo deseado q' se utiliza para calcular los parámetros del controlador utilizado.



La principal dificultad en la implementación de este tipo de control radica en la elección del algoritmo de adaptación que asegure la estabilidad y correcto funcionamiento del sistema.

Control por par calculado adaptativo

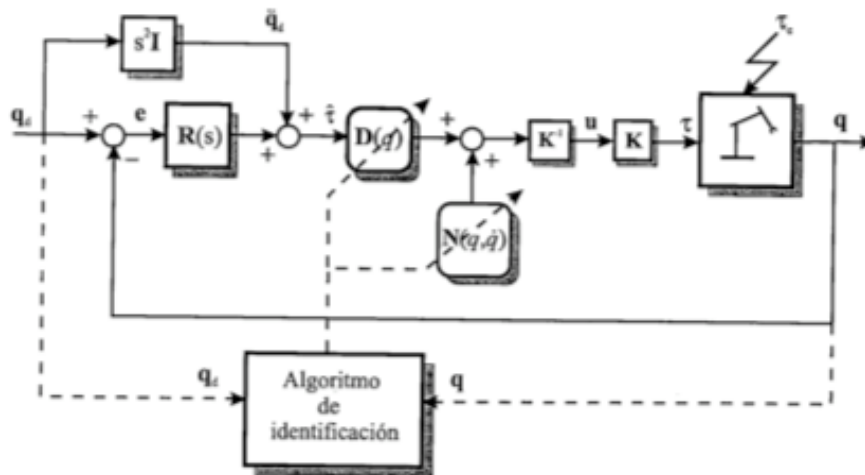
Anteriormente se examinó la técnica de desacoplamiento por inversión del modelo, en el cual el conocimiento de las matrices K , D , F_v , H y C del modelo dinámico del robot permitían aplicar la ley de control de la forma:

$$u = \left[(D(q)s^2 + F_v s) \ddot{q}_d + H(q, \dot{q}) + C(q) \right] K^{-1} + R(s) + e \quad \text{Expresión [1]}$$

Consiguiéndose que la evolución de las articulaciones respondiera a:

$$\ddot{q} = \ddot{q}_d - \left[D(q)s^2 + F_v s + K \cdot R(s) \right]^{-1} \cdot \tau_e$$

Esto es, en ausencia de perturbaciones, coincidían la trayectoria deseada q_d y la real q . Y en presencia de perturbaciones de tipo escalón, el error ($e = q_d - q$) tendería a cero.



Los problemas de este método surgen por una parte de la dificultad de calcular en tiempo real la Expresión [1] del par y por otra parte, del difícil conocimiento preciso del modelo dinámico del robot, que además de variar con la localización q del robot, depende de ciertos parámetros, como masas, inercias o rozamientos, de difícil obtención.

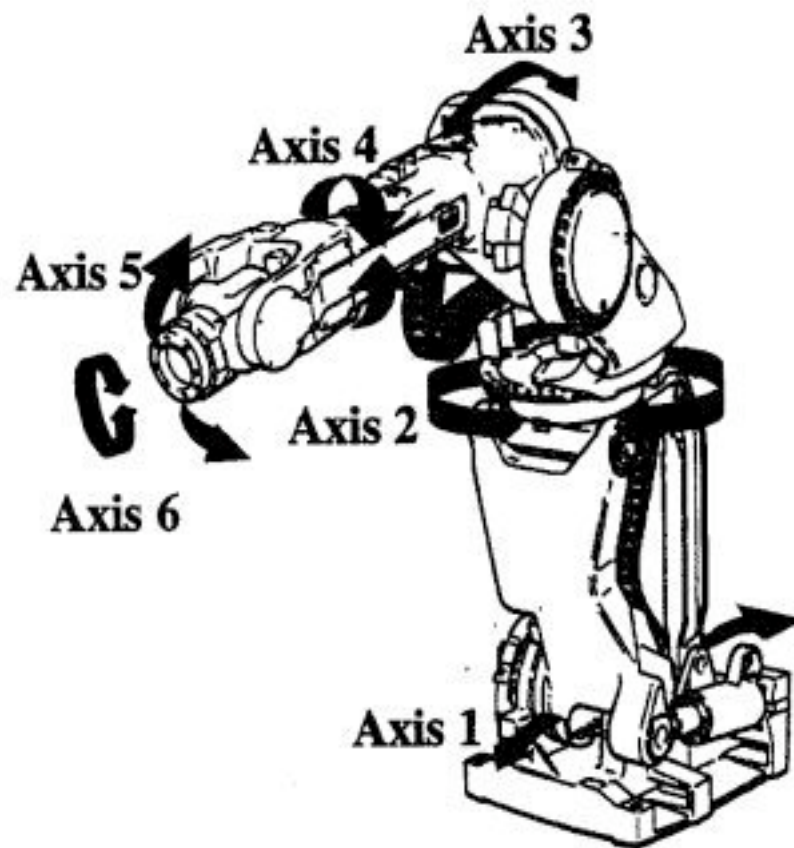
Desarrollo del diseño dinámico de Titán

Para empezar a hacer el desarrollo dinámico del robot, primero tenemos que tener en claro cuál robot estamos modelizando. Por esta razón, luego de buscar por Internet y de leer algunos libros hemos elegido el modelo **IRB6400C** de la firma ABB. A continuación mostramos una imagen del robot:



Imagen Robot IRB56400C

El mismo está compuesto por 6 articulaciones, todas de rotación. A continuación se muestra una imagen donde se pueden observar todos los ejes de las articulaciones y sus sentidos de rotación:



Ejes del Robot IRB6400C

Vamos a aplicar el método de Newton-Euler para llegar al modelo dinámico del robot, el cual ya fue ampliamente explicado y desarrollado en la introducción teórica:

1. Primer paso

Asignar a cada eslabón un sistema de referencia teniendo en cuenta las normas D-H.

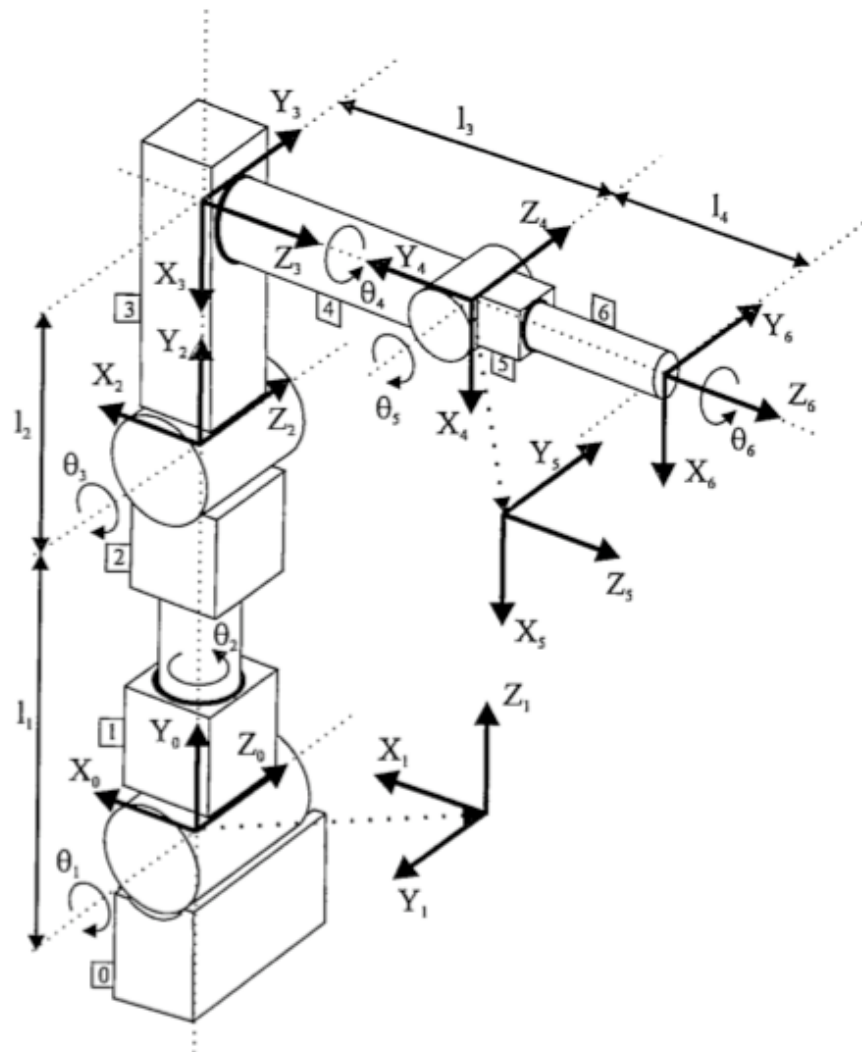


Tabla parámetros D-H

Tabla 4.2. Parámetros D-H para un robot IRB6400C

Articulación	θ	d	a	α
1	θ_1	0	0	-90
2	θ_2	l_1	0	90
3	θ_3-90	0	$-l_2$	90
4	θ_4	l_3	0	-90
5	θ_5	0	0	90
6	θ_6	l_4	0	0

2. Segundo Paso

A continuación vamos a definir los parámetros del robot, necesarios para continuar:

Masas

- $m_1=50\text{Kg}$
- $m_2=30\text{Kg}$
- $m_3=20\text{Kg}$
- $m_4=40\text{Kg}$
- $m_5=20\text{Kg}$
- $m_6=15\text{Kg}$

Longitudes

- $l_1=1,368\text{m}$
- $l_2=0,19\text{m}$
- $l_3=1,1\text{m}$
- $l_4=0,2\text{m}$

Centros de masas

Para simplificar el cálculo del modelo, vamos a suponer que las masas de las articulaciones se encuentran todas concentradas en el centro de masa, y ubicados éstos en los extremos de dichas articulaciones.

$$lcm_1 = \begin{bmatrix} 0 \\ 0 \\ 0,8m \end{bmatrix} \quad lcm_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad lcm_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad lcm_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad lcm_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad lcm_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

3. Tercer Paso

Debido a que estamos trabajando con un robot de 6 grados de libertad, la cantidad de operaciones de multiplicación y sumas a realizar se torna un número considerable (obviamente menor que utilizando el método de Lagrange-Euler), por lo que vamos a utilizar el ToolBox de Matlab "Hemero", desarrollado por el señor Aníbal Ollero Baturone, quien explica su funcionamiento en el libro *Robótica: "Manipuladores y robots móviles"*.

Para resolver las ecuaciones se utiliza una matriz llamada **dyn**. Dicha matriz posee $n \times 20$ elementos, en donde n es la cantidad de articulaciones del robot a estudiar.

Matriz dyn

Las columnas de dicha matriz se conforman con los siguientes datos:

1. $\alpha(i-1)$: Parámetros de Denavit-Hartenberg
2. $a(i-1)$
3. $\theta(i)$
4. $d(i)$
5. $\sigma(i)$: Tipo de articulación; 0 si es de rotación y 1 si es prismática
6. masa: Masa del enlace i
7. r_x : Centro de masas del enlace respecto al cuadro de referencia de dicho enlace
8. r_y
9. r_z
10. I_{xx} : Elementos del tensor de inercia referido al centro de masas del enlace
11. I_{yy}
12. I_{zz}
13. I_{xy}
14. I_{yz}
15. I_{xz}
16. J_m : Inercia de la armadura
17. G : Velocidad de la articulación / velocidad del enlace
18. B : Fricción viscosa, referida al motor
19. T_{c+} : Fricción de Coulomb (rotación positiva), referida al motor
20. T_{c-} : Fricción de Coulomb (rotación negativa), referida al motor

En nuestro caso, la matriz dyn correspondiente sería la siguiente:

$-\pi/2$	0	t_1	0	0	50	0	0	0.8	0	0	0	0	0	0	0	0	1	0	0	0
$\pi/2$	0	t_2	1.368	0	30	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\pi/2$	0.19	$(t_3 - \pi/2)$	0	0	20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$-\pi/2$	0	t_4	1.1	0	40	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$\pi/2$	0	t_5	0	0	20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	t_6	0.2	0	15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

4. Cuarto Paso

Luego, para calcular los τ correspondientes a cada articulación, se ejecutan en Matlab las siguientes instrucciones:

```
syms t1, t2, t3, t4 t5, t6 real; %Variables simbólicas de posición de
cada articulación
syms td1, td2, td3, td4 td5, td6 real; %Variables simbólicas de
velocidad de cada articulación
syms tdd1, tdd2, tdd3, tdd4 tdd5, tdd6 real; %Variables simbólicas de
aceleración de cada articulación
syms g real; %Aceleración de la gravedad
dyn=...
q=[t1 t2 t3 t4 t5 t6]; %vector de posiciones
qd=[td1 td2 td3 td4 td5 td6]; %vector de velocidades
qdd=[tdd1 tdd2 tdd3 tdd4 tdd5 tdd6]; %vector de posiciones
grav=[0 -g 0]; %vector de aceleración de la gravedad
tau=rne(dyn,q,qd,qdd,grav)
```

Con esta última instrucción obtenemos un vector de 1 fila con 6 columnas, en donde están los pares motores de cada articulación.

Al ejecutar estas instrucciones en Matlab, obtuvimos los siguientes pares motores, en forma simbólica. O sea, que de acuerdo a los valores de posición, velocidad y aceleración que le necesitemos obtener del robot, podemos calcular los pares motores que se deberán aplicar a las articulaciones del robot.

Resultados obtenidos con Matlab

- Articulación 1

$$\begin{aligned}\tau_1 = & 29316/125*tdd1 + 61731/2500*td2^2*cos(t2) + \\ & 188359/2000*tdd1*cos(t2)^2 + 1803/20*tdd1*cos(t3)^2 - 1827/20*tdd3*cos(t2) \\ & + 5643/25*tdd1*cos(t3) + 61731/2500*tdd2*sin(t2) + 57/50*td1*td2*cos(t4) + \\ & 6/5*td2*td3*sin(t2) - 5643/25*td1*td3*sin(t3) - 33/5*td1*td3*sin(t4) - \\ & 1803/20*tdd1*cos(t2)^2*cos(t3)^2 - 3/5*tdd1*cos(t2)^2*cos(t4)^2 + \\ & 3/5*tdd1*cos(t3)^2*cos(t4)^2 - 5643/50*tdd3*cos(t2)*cos(t3) + \\ & 513/125*tdd2*cos(t2)*cos(t4) + 33/10*tdd4*cos(t2)*cos(t4) + \\ & 627/40*tdd2*cos(t3)*sin(t2) - 627/40*tdd3*cos(t2)*sin(t3) + \\ & 5643/50*tdd2*sin(t2)*sin(t3) - 3/5*tdd4*sin(t2)*sin(t3) + \\ & 33/10*tdd2*sin(t2)*sin(t4) - 513/125*tdd4*sin(t2)*sin(t4) + \\ & 1026/125*tdd1*sin(t3)*sin(t4) + 627/40*td2^2*cos(t2)*cos(t3) - \\ & 627/40*td3^2*cos(t2)*cos(t3) + 3/5*tdd3*cos(t2)*cos(t4)^2 +\end{aligned}$$

$$\begin{aligned}
 & 5643/50*td2^2*cos(t2)*sin(t3) + 5643/50*td3^2*cos(t2)*sin(t3) + \\
 & 627/20*td1*cos(t2)^2*sin(t3) + 33/10*td2^2*cos(t2)*sin(t4) - \\
 & 513/125*td2^2*cos(t4)*sin(t2) - 33/10*td4^2*cos(t2)*sin(t4) - \\
 & 513/125*td4^2*cos(t4)*sin(t2) - 188359/1000*td1*td2*cos(t2)*sin(t2) + \\
 & 5643/25*td2*td3*cos(t3)*sin(t2) - 6/5*td2*td4*cos(t2)*sin(t3) - \\
 & 1803/10*td1*td3*cos(t3)*sin(t3) - 1026/125*td2*td4*cos(t2)*sin(t4) + \\
 & 33/5*td1*td2*cos(t4)*sin(t3) + 1026/125*td1*td3*cos(t3)*sin(t4) + \\
 & 1026/125*td1*td4*cos(t4)*sin(t3) - 33/5*td2^2*cos(t2)*cos(t3)^2*sin(t4) + \\
 & 627/20*td1*td3*cos(t2)^2*cos(t3) - 57/25*td1*td2*cos(t2)^2*cos(t4) + \\
 & 33/10*td2^2*cos(t2)*cos(t3)*cos(t4) + 513/125*td4^2*cos(t2)*cos(t3)*cos(t4) + \\
 & 1803/10*td2*td3*cos(t3)^2*sin(t2) + 33/5*td1*td3*cos(t2)^2*sin(t4) - \\
 & 6/5*td2*td3*cos(t4)^2*sin(t2) + 66/5*td1*td3*cos(t3)^2*sin(t4) - \\
 & 57/50*td1*cos(t2)*cos(t4)*sin(t2) + 57/100*td3*cos(t2)*cos(t3)*sin(t4) + \\
 & 57/100*td4*cos(t2)*cos(t4)*sin(t3) - 3/5*td1*cos(t2)^2*cos(t3)^2*cos(t4)^2 + \\
 & 1803/20*td2^2*cos(t3)*sin(t2)*sin(t3) - 513/125*td2^2*cos(t3)*sin(t2)*sin(t4) - \\
 & 513/125*td3^2*cos(t2)*sin(t3)*sin(t4) + 33/10*td3^2*cos(t4)*sin(t2)*sin(t3) - \\
 & 33/10*td4^2*cos(t3)*sin(t2)*sin(t4) + 33/5*td1*cos(t3)*sin(t3)*sin(t4) + \\
 & 57/100*td2^2*sin(t2)*sin(t3)*sin(t4) + 1803/20*td2^2*cos(t2)*cos(t3)*sin(t3) - \\
 & 513/125*td2^2*cos(t2)*cos(t3)*sin(t4) - 33/10*td2^2*cos(t3)*cos(t4)*sin(t2) - \\
 & 513/125*td3^2*cos(t2)*cos(t3)*sin(t4) + 33/10*td3^2*cos(t3)*cos(t4)*sin(t2) - \\
 & 513/125*td4^2*cos(t2)*cos(t3)*sin(t4) - 33/10*td4^2*cos(t3)*cos(t4)*sin(t2) - \\
 & 57/50*td1*cos(t2)^2*cos(t3)*sin(t4) + 57/100*td2^2*cos(t2)*sin(t3)*sin(t4) - \\
 & 57/100*td3^2*cos(t2)*sin(t3)*sin(t4) - 57/100*td4^2*cos(t2)*sin(t3)*sin(t4) - \\
 & 33/5*td2^2*cos(t3)^2*sin(t2)*sin(t4) + \\
 & 3/5*td2^2*cos(t2)*cos(t3)*cos(t4)^2*sin(t3) - \\
 & 57/50*td1*td4*cos(t2)^2*cos(t3)*cos(t4) + \\
 & 1803/10*td1*td2*cos(t2)*cos(t3)^2*sin(t2) + \\
 & 6/5*td1*td2*cos(t2)*cos(t4)^2*sin(t2) + \\
 & 1803/10*td1*td3*cos(t2)^2*cos(t3)*sin(t3) - \\
 & 66/5*td1*td2*cos(t2)^2*cos(t4)*sin(t3) + 6/5*td2*td4*cos(t2)*cos(t4)^2*sin(t3) - \\
 & 33/5*td2*td4*cos(t3)^2*cos(t4)*sin(t2) - 6/5*td3*td4*cos(t3)*cos(t4)^2*sin(t2) - \\
 & 6/5*td1*td3*cos(t3)*cos(t4)^2*sin(t3) + 6/5*td1*td4*cos(t2)^2*cos(t4)*sin(t4) - \\
 & 6/5*td1*td4*cos(t3)^2*cos(t4)*sin(t4) + 57/50*td1*td3*cos(t2)^2*sin(t3)*sin(t4) \\
 & - 33/5*td1*cos(t2)*cos(t4)*sin(t2)*sin(t3) + \\
 & 3/5*td2^2*cos(t2)*cos(t4)*sin(t3)*sin(t4) - \\
 & 3/5*td3^2*cos(t3)*cos(t4)*sin(t2)*sin(t4) - \\
 & 66/5*td1*td3*cos(t2)^2*cos(t3)^2*sin(t4) + \\
 & 6/5*td2*td3*cos(t3)^2*cos(t4)^2*sin(t2) - \\
 & 33/5*td1*cos(t2)^2*cos(t3)*sin(t3)*sin(t4) + \\
 & 3/5*td2^2*cos(t3)*cos(t4)^2*sin(t2)*sin(t3) - \\
 & 3/5*td2^2*cos(t4)*sin(t2)*sin(t3)*sin(t4) + \\
 & 3/5*td3^2*cos(t4)*sin(t2)*sin(t3)*sin(t4) + \\
 & 57/50*td3*td4*cos(t2)*cos(t3)*cos(t4) - 6/5*td1*td4*cos(t2)*cos(t3)*sin(t2) - \\
 & 33/5*td2*td4*cos(t2)*cos(t3)*sin(t4) - 1026/125*td2*td4*cos(t3)*cos(t4)*sin(t2) \\
 & - 1026/125*td3*td4*cos(t2)*cos(t4)*sin(t3) + \\
 & 33/5*td1*td4*cos(t3)*cos(t4)*sin(t3) - 6/5*td3*td4*cos(t2)*cos(t4)*sin(t4) - \\
 & 6/5*td1*td2*cos(t3)*cos(t4)*sin(t4) - 627/10*td1*td2*cos(t2)*sin(t2)*sin(t3) + \\
 & 57/50*td1*td4*cos(t2)*sin(t2)*sin(t4) + 1026/125*td2*td3*sin(t2)*sin(t3)*sin(t4) \\
 & + 12/5*td1*td4*cos(t2)*cos(t3)*cos(t4)^2*sin(t2) -
 \end{aligned}$$

$$\begin{aligned}
 & 33/5*td1*td4*cos(t2)^2*cos(t3)*cos(t4)*sin(t3) + \\
 & 12/5*td1*td2*cos(t2)^2*cos(t3)*cos(t4)*sin(t4) + \\
 & 6/5*td1*td2*cos(t2)*cos(t3)^2*cos(t4)^2*sin(t2) + \\
 & 6/5*td1*td3*cos(t2)^2*cos(t3)*cos(t4)^2*sin(t3) + \\
 & 6/5*td1*td4*cos(t2)^2*cos(t3)^2*cos(t4)*sin(t4) - \\
 & 33/5*td1*td3*cos(t2)*cos(t3)*cos(t4)*sin(t2) + \\
 & 57/25*td1*td2*cos(t2)*cos(t3)*sin(t2)*sin(t4) + \\
 & 33/5*td1*td4*cos(t2)*sin(t2)*sin(t3)*sin(t4) + \\
 & 66/5*td2*td3*cos(t3)*sin(t2)*sin(t3)*sin(t4) + \\
 & 66/5*td1*td2*cos(t2)*cos(t3)*sin(t2)*sin(t3)*sin(t4) - \\
 & 6/5*td1*td3*cos(t2)*cos(t4)*sin(t2)*sin(t3)*sin(t4) - \\
 & 6/5*td2*td4*cos(t3)*cos(t4)*sin(t2)*sin(t3)*sin(t4)
 \end{aligned}$$

- Articulación 2

$$\begin{aligned}
 \tau_2 = & (188359*td2)/2000 - (17689*cos(t2))/100 - (1803*td2*cos(t3)^2)/20 - \\
 & (57*td1^2*cos(t4))/100 - (57*td4^2*cos(t4))/100 + (3*td2*cos(t4)^2)/5 - \\
 & (1617*cos(t2)*sin(t3))/2 + (147*cos(t4)*sin(t2))/5 + (3*td4*cos(t3))/5 + \\
 & (61731*td1*sin(t2))/2500 + (627*td2*sin(t3))/20 - (57*td4*sin(t4))/100 + \\
 & (627*td2*td3*cos(t3))/20 - (363*td1*td3*sin(t2))/2 + (33*td2*td3*sin(t4))/5 + \\
 & (57*td1^2*cos(t2)^2*cos(t4))/50 - (3*td2*cos(t3)^2*cos(t4)^2)/5 + \\
 & (513*td1*cos(t2)*cos(t4))/125 - (33*td3*cos(t3)*cos(t4))/10 + \\
 & (627*td1*cos(t3)*sin(t2))/40 - (57*td2*cos(t3)*sin(t4))/50 + \\
 & (5643*td1*sin(t2)*sin(t3))/50 + (33*td1*sin(t2)*sin(t4))/10 - \\
 & (33*td4*sin(t3)*sin(t4))/10 + (188359*td1^2*cos(t2)*sin(t2))/2000 - \\
 & (33*td1^2*cos(t4)*sin(t3))/10 + (33*td3^2*cos(t4)*sin(t3))/10 - \\
 & (33*td4^2*cos(t4)*sin(t3))/10 + (147*cos(t2)*cos(t3)*sin(t4))/5 + \\
 & (33*td1*td4*cos(t4)*sin(t2))/5 + (1803*td2*td3*cos(t3)*sin(t3))/10 - \\
 & (6*td2*td4*cos(t4)*sin(t4))/5 - (627*td1*td3*sin(t2)*sin(t3))/20 + \\
 & (57*td2*td3*sin(t3)*sin(t4))/50 - (1803*td1^2*cos(t2)*cos(t3)^2*sin(t2))/20 - \\
 & (3*td1^2*cos(t2)*cos(t4)^2*sin(t2))/5 + (33*td1^2*cos(t2)^2*cos(t4)*sin(t3))/5 + \\
 & (33*td1*cos(t2)*cos(t3)*cos(t4))/10 + (1803*td1*td3*cos(t3)^2*sin(t2))/10 - \\
 & (66*td2*td3*cos(t3)^2*sin(t4))/5 - (6*td3*td4*cos(t4)^2*sin(t3))/5 + \\
 & (1803*td1*cos(t3)*sin(t2)*sin(t3))/20 - (513*td1*cos(t3)*sin(t2)*sin(t4))/125 - \\
 & (33*td2*cos(t3)*sin(t3)*sin(t4))/5 - (3*td3*cos(t4)*sin(t3)*sin(t4))/5 + \\
 & (57*td1*sin(t2)*sin(t3)*sin(t4))/100 + (3*td1^2*cos(t3)*cos(t4)*sin(t4))/5 - \\
 & (3*td3^2*cos(t3)*cos(t4)*sin(t4))/5 + (627*td1^2*cos(t2)*sin(t2)*sin(t3))/20 - \\
 & (33*td1*cos(t3)^2*sin(t2)*sin(t4))/5 - (57*td2*td4*cos(t3)*cos(t4))/50 - \\
 & (6*td1^2*cos(t2)^2*cos(t3)*cos(t4)*sin(t4))/5 + \\
 & (6*td1*td4*cos(t2)*cos(t4)^2*sin(t3))/5 - \\
 & (33*td1*td4*cos(t3)^2*cos(t4)*sin(t2))/5 + \\
 & (6*td2*td3*cos(t3)*cos(t4)^2*sin(t3))/5 + (6*td2*td4*cos(t3)^2*cos(t4)*sin(t4))/5 \\
 & + (3*td1*cos(t2)*cos(t4)*sin(t3)*sin(t4))/5 - \\
 & (3*td1^2*cos(t2)*cos(t3)^2*cos(t4)^2*sin(t2))/5 + \\
 & (6*td1*td3*cos(t3)^2*cos(t4)^2*sin(t2))/5 - \\
 & (57*td1^2*cos(t2)*cos(t3)*sin(t2)*sin(t4))/50 + \\
 & (3*td1*cos(t3)*cos(t4)^2*sin(t2)*sin(t3))/5 - \\
 & (33*td1*td3*cos(t2)*cos(t4)*sin(t3))/5 - (33*td2*td4*cos(t3)*cos(t4)*sin(t3))/5 + \\
 & (57*td1*td3*cos(t3)*sin(t2)*sin(t4))/50 + (57*td1*td4*cos(t4)*sin(t2)*sin(t3))/50
 \end{aligned}$$

$$\begin{aligned}
 & - (33*td1^2*cos(t2)*cos(t3)*sin(t2)*sin(t3)*sin(t4))/5 + \\
 & (6*td1*td3*cos(t2)*cos(t3)*cos(t4)*sin(t4))/5 + \\
 & (66*td1*td3*cos(t3)*sin(t2)*sin(t3)*sin(t4))/5 - \\
 & (6*td1*td4*cos(t3)*cos(t4)*sin(t2)*sin(t3)*sin(t4))/5
 \end{aligned}$$

- Articulación 3

$$\begin{aligned}
 \tau_3 = & (1827*td3)/20 - (627*td2^2*cos(t3))/40 - (3*td3*cos(t4)^2)/5 + \\
 & (5643*td1^2*sin(t3))/50 + (33*td1^2*sin(t4))/10 - (33*td2^2*sin(t4))/10 + \\
 & (33*td4^2*sin(t4))/10 - (1617*cos(t3)*sin(t2))/2 - (1827*td1*cos(t2))/20 - \\
 & (33*td4*cos(t4))/10 - (57*td2^2*sin(t3)*sin(t4))/100 - \\
 & (147*sin(t2)*sin(t3)*sin(t4))/5 + (363*td1*td2*sin(t2))/2 + (6*td2*td4*sin(t3))/5 - \\
 & (627*td1^2*cos(t2)^2*cos(t3))/40 - (33*td1^2*cos(t2)^2*sin(t4))/10 - \\
 & (33*td1^2*cos(t3)^2*sin(t4))/5 + (33*td2^2*cos(t3)^2*sin(t4))/5 - \\
 & (5643*td1*cos(t2)*cos(t3))/50 - (33*td2*cos(t3)*cos(t4))/10 - \\
 & (627*td1*cos(t2)*sin(t3))/40 + (3*td1*cos(t2)*cos(t4)^2)/5 + \\
 & (1803*td1^2*cos(t3)*sin(t3))/20 - (1803*td2^2*cos(t3)*sin(t3))/20 - \\
 & (513*td1^2*cos(t3)*sin(t4))/125 + (6*td1*td4*cos(t3)*sin(t2))/5 + \\
 & (33*td2*td4*cos(t3)*sin(t4))/5 + (6*td3*td4*cos(t4)*sin(t4))/5 + \\
 & (627*td1*td2*sin(t2)*sin(t3))/20 - (1803*td1^2*cos(t2)^2*cos(t3)*sin(t3))/20 + \\
 & (3*td1^2*cos(t3)*cos(t4)^2*sin(t3))/5 - (3*td2^2*cos(t3)*cos(t4)^2*sin(t3))/5 - \\
 & (57*td1^2*cos(t2)^2*sin(t3)*sin(t4))/100 - (1803*td1*td2*cos(t3)^2*sin(t2))/10 - \\
 & (6*td2*td4*cos(t4)^2*sin(t3))/5 + (57*td1*cos(t2)*cos(t3)*sin(t4))/100 - \\
 & (513*td1*cos(t2)*sin(t3)*sin(t4))/125 + (33*td1*cos(t4)*sin(t2)*sin(t3))/10 - \\
 & (3*td2*cos(t4)*sin(t3)*sin(t4))/5 + (33*td1^2*cos(t2)^2*cos(t3)^2*sin(t4))/5 - \\
 & (6*td1*td4*cos(t3)*cos(t4)^2*sin(t2))/5 - \\
 & (3*td1*cos(t3)*cos(t4)*sin(t2)*sin(t4))/5 - \\
 & (3*td1^2*cos(t2)^2*cos(t3)*cos(t4)^2*sin(t3))/5 - \\
 & (6*td1*td2*cos(t3)^2*cos(t4)^2*sin(t2))/5 + \\
 & (33*td1^2*cos(t2)*cos(t3)*cos(t4)*sin(t2))/10 + \\
 & (33*td1*td2*cos(t2)*cos(t4)*sin(t3))/5 - (6*td1*td4*cos(t2)*cos(t4)*sin(t4))/5 - \\
 & (57*td1*td2*cos(t3)*sin(t2)*sin(t4))/50 - (33*td1*td4*sin(t2)*sin(t3)*sin(t4))/5 + \\
 & (3*td1^2*cos(t2)*cos(t4)*sin(t2)*sin(t3)*sin(t4))/5 - \\
 & (6*td1*td2*cos(t2)*cos(t3)*cos(t4)*sin(t4))/5 - \\
 & (66*td1*td2*cos(t3)*sin(t2)*sin(t3)*sin(t4))/5
 \end{aligned}$$

- Articulación 4

$$\begin{aligned}
 \tau_4 = & (3*td4)/5 + (147*cos(t2)*sin(t4))/5 + (3*td2*cos(t3))/5 - \\
 & (33*td3*cos(t4))/10 - (57*td2*sin(t4))/100 - (6*td2*td3*sin(t3))/5 + \\
 & (33*td1*cos(t2)*cos(t4))/10 - (3*td1*sin(t2)*sin(t3))/5 - \\
 & (513*td1*sin(t2)*sin(t4))/125 - (33*td2*sin(t3)*sin(t4))/10 + \\
 & (57*td2^2*cos(t3)*cos(t4))/100 - (513*td1^2*cos(t4)*sin(t3))/125 + \\
 & (3*td2^2*cos(t4)*sin(t4))/5 - (3*td3^2*cos(t4)*sin(t4))/5 + \\
 & (147*cos(t3)*cos(t4)*sin(t2))/5 - (6*td1*td3*cos(t3)*sin(t2))/5 - \\
 & (33*td1*td2*cos(t4)*sin(t2))/5 - (33*td2*td3*cos(t3)*sin(t4))/5 + \\
 & (57*td1^2*cos(t2)^2*cos(t3)*cos(t4))/100 - \\
 & (3*td1^2*cos(t2)^2*cos(t4)*sin(t4))/5 + (3*td1^2*cos(t3)^2*cos(t4)*sin(t4))/5 - \\
 & (3*td2^2*cos(t3)^2*cos(t4)*sin(t4))/5 + (513*td1*cos(t2)*cos(t3)*cos(t4))/125 + \\
 & (6*td2*td3*cos(t4)^2*sin(t3))/5 + (57*td1*cos(t2)*cos(t4)*sin(t3))/100 -
 \end{aligned}$$

$$\begin{aligned} & (33*td1*cos(t3)*sin(t2)*sin(t4))/10 + (3*td1^2*cos(t2)*cos(t3)*sin(t2))/5 - \\ & (33*td1^2*cos(t3)*cos(t4)*sin(t3))/10 + (33*td2^2*cos(t3)*cos(t4)*sin(t3))/10 - \\ & (57*td1^2*cos(t2)*sin(t2)*sin(t4))/100 - \\ & (6*td1^2*cos(t2)*cos(t3)*cos(t4)^2*sin(t2))/5 + \\ & (33*td1^2*cos(t2)^2*cos(t3)*cos(t4)*sin(t3))/10 - \\ & (6*td1*td2*cos(t2)*cos(t4)^2*sin(t3))/5 + \\ & (33*td1*td2*cos(t3)^2*cos(t4)*sin(t2))/5 + \\ & (6*td1*td3*cos(t3)*cos(t4)^2*sin(t2))/5 - \\ & (3*td1^2*cos(t2)^2*cos(t3)^2*cos(t4)*sin(t4))/5 - \\ & (33*td1^2*cos(t2)*sin(t2)*sin(t3)*sin(t4))/10 + \\ & (6*td1*td3*cos(t2)*cos(t4)*sin(t4))/5 - (57*td1*td2*cos(t4)*sin(t2)*sin(t3))/50 + \\ & (33*td1*td3*sin(t2)*sin(t3)*sin(t4))/5 + \\ & (6*td1*td2*cos(t3)*cos(t4)*sin(t2)*sin(t3)*sin(t4))/5 \end{aligned}$$

Articulación 5

$$\tau_5 = 0$$

Articulación 6

$$\tau_6 = 0$$

Aclaración: Las dos últimas articulaciones tienen el par motor igual a cero, ya que el extremo del robot está sin carga.

Desarrollo del diseño del control dinámico de Titán

Los robots utilizan normalmente el control clásico tipo PID. Pero los brazos manipuladores, los cuales se componen de varias articulaciones unidas entre sí, poseen una dinámica altamente no lineal con un fuerte acoplamiento entre sus respectivas articulaciones. Esto complica la tarea del controlador PID así como su sintonización.

Debido a lo antes mencionado se selecciono la técnica de par computado. Esta técnica permite compensar los efectos gravitatorios, centrífugo y rozamientos generando un par apropiado.

El modelo dinámico de cualquier manipulador puede expresarse como:

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

Donde τ , G , V , F son vectores de n elementos (donde n es la cantidad de grados de libertad que posea el robot), y $M(q)$ es una matriz de $n \times n$ denominada matriz de masas. El término $G(q)$ representa los efectos gravitatorios y depende de las variables articulares. $V(\dot{q})$ representa los pares generados por las fuerzas centrífugas y de Coriolis y en el caso de que la velocidad sea pequeña es despreciable. Por último el término $F(\dot{q})$ se denomina vector de fricciones y en el análisis realizado no se considera.

La matriz de masas $M(q)$ es una matriz simétrica, definida positiva y está acotada superiormente e inferiormente ya que depende de las variables q a través de funciones trigonométricas. Un aspecto importante de dicha matriz es que es siempre invertible y su inversa es también definida positiva y acotada.

El vector $G(q)$ contiene los términos que dependen de la gravedad g , y también depende de las variables articulares a través de senos y cosenos por lo que está acotado. El término de fricciones $F(\dot{q})$ es complejo y normalmente difícil de aproximar. En principio depende de las variables articulares, y lo que se hace es proponer modelos e intentar ver si se ajusta a la realidad.

Para la obtención del modelo dinámico del manipulador, se ha utilizado la herramienta Matlab. La función permite obtenerlo, pasándole una serie de parámetros que caracterizan al manipulador. La implementación se muestra a continuación.

Implementación del control dinámico de Titán

Utilizando los valores obtenidos en el punto anterior, mediante el algoritmo DH, se pueden obtener las matrices necesarias para realizar el control de par computado. Para lograrlo utilizamos la ToolBox HEMERO que nos ayuda a calcular las matrices indicadas a continuación:

$M(q)$ una matriz de masas $n \times n$ simétrica definida positiva,

$V(q, \dot{q})$ un vector de n fuerzas centrífugas y de Coriolis,

$G(q)$ un vector de n términos de gravedad,

El siguiente script de Matlab es el utilizado para realizar lo dicho anteriormente:

```
%% script para calcular las matrices de robótica
g = -9.81;
syms t1 t2 t3 t4 t5 t6 real; % Variables articulares
syms td1 td2 td3 td4 td5 td6 real; % Velocidades articulares
syms tdd1 tdd2 tdd3 tdd4 tdd5 tdd6 real; % Aceleraciones articulares

dyn = [-pi/2 0 t1 0 0 50 0 0 0.8 0 0 0 0 0 0 1 0 0 0; ...
        pi/2 0 t2 1.368 0 30 0 0 0 0 0 0 0 0 0 0 1 0 0 0; ...
        pi/2 -0.19 (t3-pi/2) 0 0 20 0 0 0 0 0 0 0 0 0 0 1 0 0 0; ...
        -pi/2 0 t4 1.1 0 40 0 0 0 0 0 0 0 0 0 0 1 0 0 0; ...
        pi/2 0 t5 0 0 20 0 0 0 0 0 0 0 0 0 0 1 0 0 0;...
        0 0 t6 0.2 0 15 0 0 0 0 0 0 0 0 0 0 1 0 0 0];

q = [t1 t2 t3 t4 t5 t6]; % Vector de variables articulares
qd = [td1 td2 td3 td4 td5 td6]; % Vector de velocidades articulares
qdd = [tdd1 tdd2 tdd3 tdd4 tdd5 tdd6]; % Vector de aceleraciones articulares
grav = [0 g 0]; % Vector aceleración de la gravedad
tau = rne (dyn, q, qd, qdd, grav);
resultado = simple (tau);
```

%% calculo de las matrices extras

M = inertia (dyn, q); % Matriz de masas M(e)

G = gravity (dyn, q, grav); % Término gravitatorio G(e)

V = coriolis (dyn, q, qd); % Términos centrífugos y de Coriolis V(e,e')

Una vez ejecutado el script se pueden utilizar los parámetros para calcular a mano el control para el robot.

Simulación del control dinámico de Titán

Se eligió utilizar un control PID debido a sus grandes beneficios en cuanto a poder obtener cualquier tipo de respuesta con facilidad.

Una vez obtenido los parámetros del algoritmo de DH se debe realizar la matriz dyn. La cual está formada de la manera indicada anteriormente. Se debe colocar la matriz exacta en los bloques en donde está la etiqueta "REAL" mientras que los bloques que están etiquetados como "estimada" son los valores medidos. Esto permite corregir los defectos medidos.

Los coeficientes del control PID son extraídos de las siguientes ecuaciones:

$$k_p = \omega_n$$

$$k_v = 2\omega_n$$

Seleccionando adecuadamente ω_n se puede obtener un el resultado deseado. En cuanto a K_i se debe calcular dependiendo del tiempo de establecimiento que se requiere y eliminar el error en estado estacionario pero sin hacer inestable el sistema. En este caso particular se eligió $\omega_n = 5$ y $k_i = 100$.

Se puede observar claramente la composición del τ (tau) vista anteriormente. Se compone de la suma de las Salidas de los bloques G, M y V (estimada).

Una vez calculados todos los parámetros necesarios se pasa a la realización del modelo en el Simulink. El modelo se muestra en la figura que se encuentra a continuación.

Aclaración: Las trayectorias mostradas en modelo son de prueba. Las salidas deben copiarse al workspace para poder ser estudiadas más a fondo. Dichas variables son q , q' y q'' .

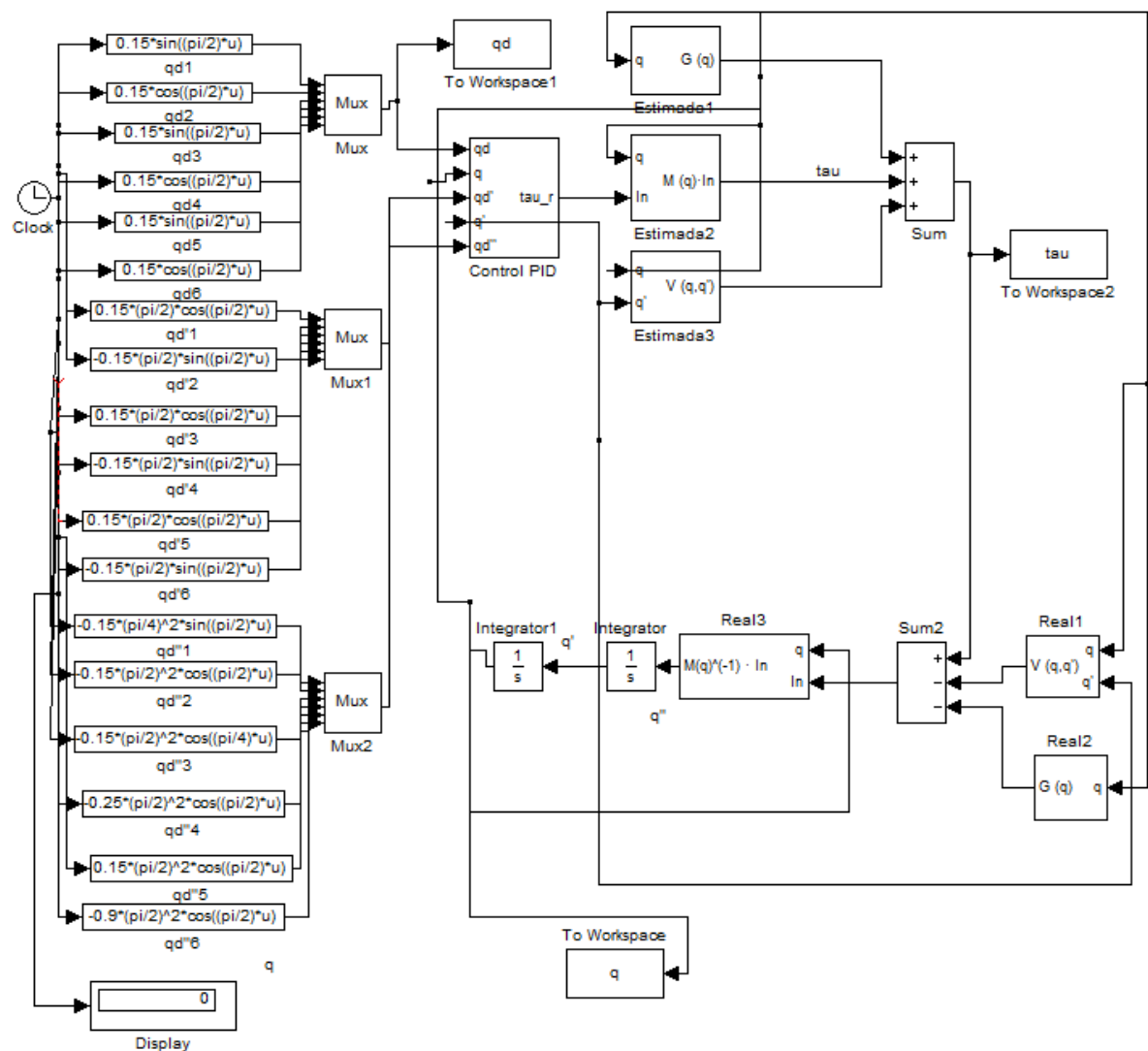


Fig. 1 – Modelo del control dinámico

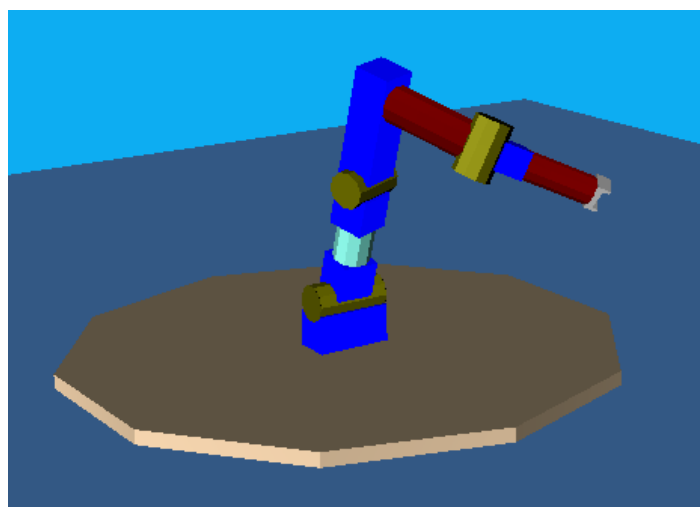
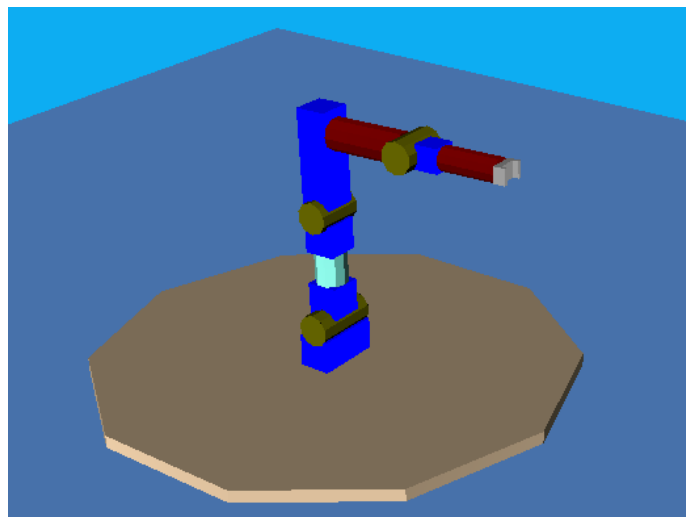
Simulación utilizando RoboWorks

Para visualizar gráficamente al robot, utilizamos un software llamado RoboWorks, que facilita el diseño gráfico en 3D y la visualización de los movimientos resultantes.

En este software es posible graficar el robot con todas sus articulaciones, definiendo los parámetros correspondientes (longitudes, ángulos máximos de rotación, etc.). A su vez, se le asignará una trayectoria a cada articulación y el programa permitirá simularlas, con el objeto de verificar el movimiento final del robot.

Este programa es de gran ayuda, ya que nos permite visualizar a simple vista si el robot está trabajando dentro del área permitida o hay algún conflicto en su movimiento.

A continuación se ven dos fotos de nuestro robot graficado en RoboWorks:



Conclusiones

La realización de esta tesis nos resultó de gran ayuda para la comprensión e incorporación de la teoría dinámica para la robótica. Pudimos evaluar los distintos métodos de desarrollo del modelo dinámico, comparando sus beneficios y complejidades, y obtuvimos como resultado que el método de Newton-Euler permite la resolución de una manera sistemática, disminuyendo considerablemente la complejidad, lo cual permite la generación de algoritmos computacionales para la resolución del problema.

Pudimos comprobar que el Matlab es una herramienta esencial en estos casos, ya que ofrece ToolBoxes que permiten plantear los problemas de robótica y ofrecen funciones para resolverlos.

El uso de la aplicación Simulink, permite fácilmente esquematizar los sistemas del control del robot, obteniendo así las respuestas en cada punto del sistema. Esto es de gran ayuda, ya que permite graficar las señales y poder visualizar más fácilmente donde se encuentran los errores.

Lamentablemente, no pudimos realizar la simulación correspondiente en Simulink debido a algunos problemas internos en la programación de los bloques, pero pudimos esquematizar el control y comprender la función de cada uno de sus bloques constitutivos.

Para realizar la interfaz gráfica utilizamos el software RoboWorks, el cual nos permitió dibujar en 3D el robot completo con sus parámetros característicos. Esta es una herramienta importante, y que nos permite asignarle una trayectoria a cada articulación, y de este modo observar la trayectoria final del robot en 3D.

Luego de haber realizado el trabajo completo llegamos a la conclusión de que la robótica es una rama interesante de la electrónica, pero a su vez también compleja. Creemos que este trabajo es importante para formar una base teórica, pero seguramente será necesario profundizar más para poder llevar nuestro conocimiento a la práctica.

Referencias

- Barrientos Antonio, “Fundamentos de Robótica”, McGraw Hill, 1997.
- Craig John, “Robótica”, Pearson Education, 2006.
- Olleros Baturone Aníbal, Alfaomega, 2007.
- Spong, “Robot modeling and control”, Wiley, 2005.