

# Modelado Dinámico

CONTROL Y PROGRAMACIÓN DE ROBOTS

---

Grado en Electrónica, Robótica y Mecatrónica

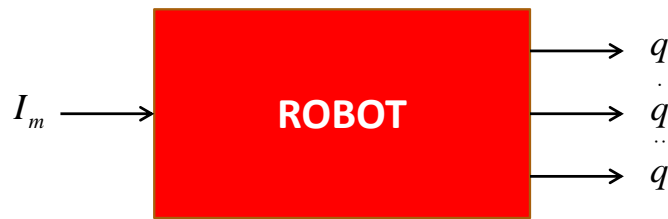
## Índice

---

1. Objetivo para el trabajo de curso
2. Simulación del robot
3. Modelo basado en objeto *robot* (con *Robotics Toolbox*)
4. Ecuaciones para el modelado
5. Implementación del modelo
6. Linealización del modelo

# Objetivo para trabajo de curso

## Simulador para control de un robot de N g.d.l.:



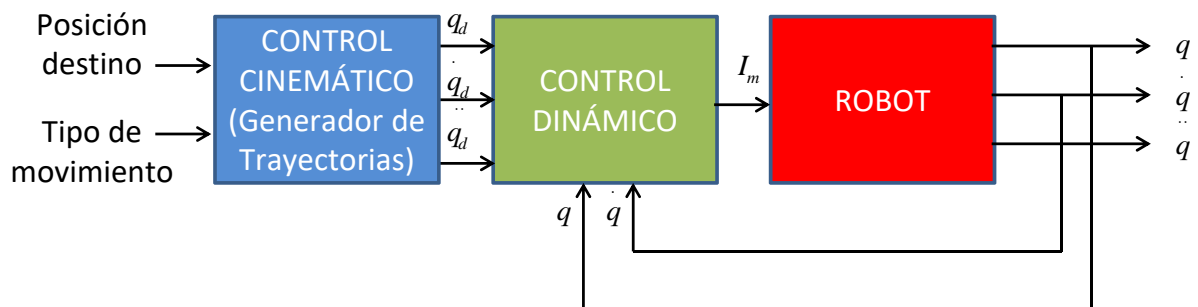
$$K_t R I_m = \left( M(q) + J_m R^2 \right) \ddot{q} + \left( C(q, \dot{q}) + B_m R^2 \right) \dot{q} + G(q) + F(\dot{q})$$

$$K_t I_m / R = \left( M(q) / R^2 + J_m \right) \ddot{q} + \left( C(q, \dot{q}) / R^2 + B_m \right) \dot{q} + G(q) / R^2 + F(\dot{q}) / R^2$$

# Objetivo para trabajo de curso

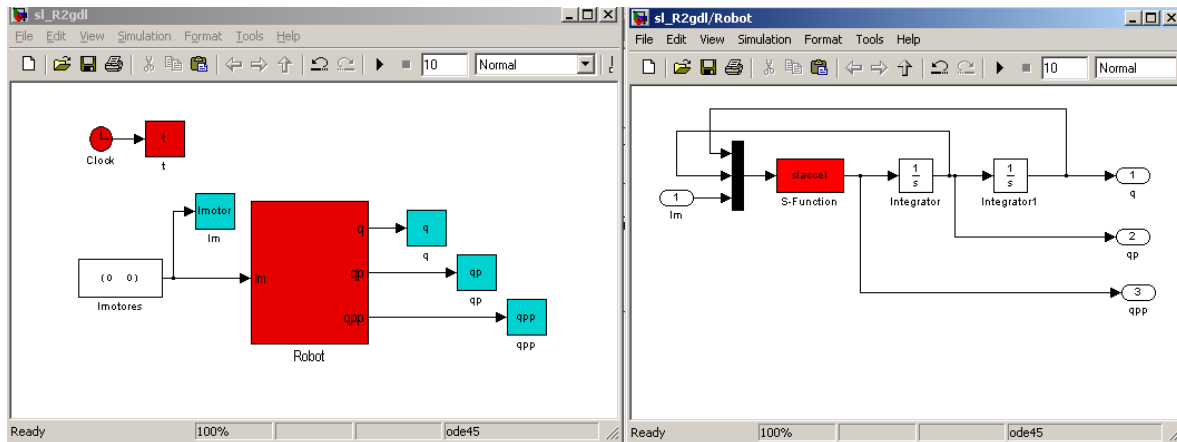
## Esquema general de control:

- Simulación de comandos como MOVE, MOVEL, MOVED, ...
- Entorno de simulación en **Matlab** y **Simulink**.



# Simulación del robot

Sin ecuaciones analíticas:

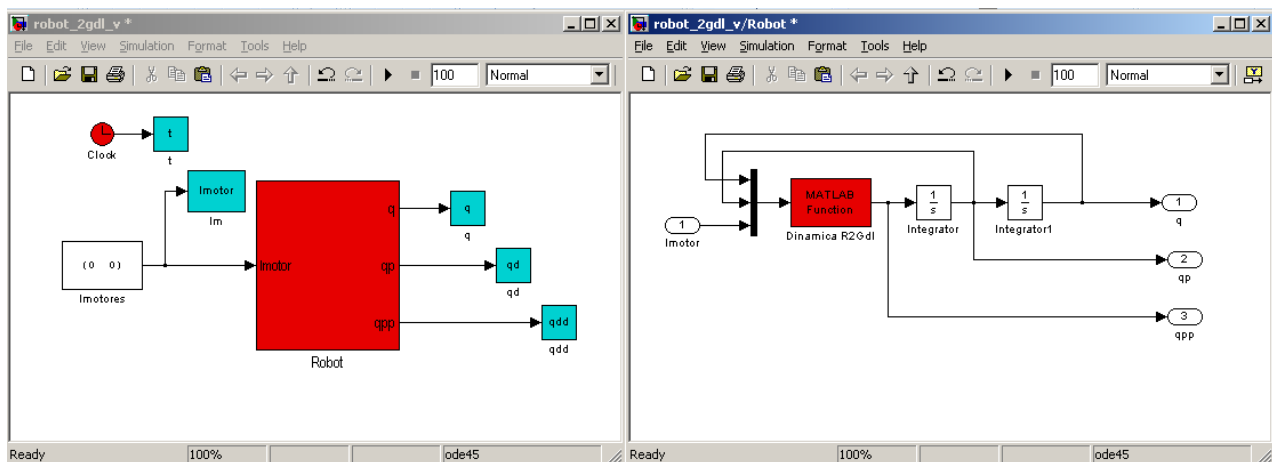


Basada en la *S-function* **slaccel** de *Robotics Toolbox*  
(ejecución lenta)

Condiciones iniciales en los integradores

# Simulación del robot

Con ecuaciones analíticas:



Basada en una *M-function* creada por el alumno  
(ejecución más rápida)

Condiciones iniciales en los integradores

Otra posibilidad es crear una S-function, compilada en C

# Modelo basado en objeto *robot*

- Uso del *Robotis Toolbox* para crear un objeto robot
- Información cinemática de eslabones y articulaciones:
  - Basada en parámetros de *Denavit-Hartenberg*

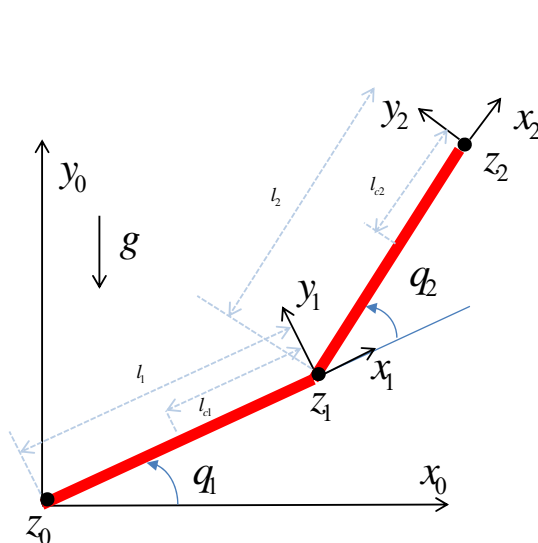
$$X_i = Z_i \times Z_{i-1}$$

- Información dinámica:
  - Eslabones: masas, centro de masas, tensores de inercia, ...
  - Actuadores: inercias, fricciones, factores de reducción, ...
  - Simulador recibe “pares en las articulaciones” (no en motores)

Cuidado con el **modelo dinámico** y la **orientación de la base** cuando el eje **Z0 no está en vertical**

# Modelo basado en objeto *robot*

- Ejemplo: **Robot plano RR en vertical**



$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

## Eslabones

$$m_1 = m_2 = 3 \text{ Kg} \quad l_1 = l_2 = 1 \text{ m}$$

$$I_1 = I_2 = 0.2536 \text{ Kg.m}^2 \quad l_{c1} = l_{c2} = 0.5 \text{ m}$$

## Motores

$$J_{m1} = J_{m2} = 0.025 \text{ Kg.m}^2$$

$$B_{m1} = B_{m2} = 3.6 \times 10^{-6} \text{ Nm/(rad/s)}$$

$$K_{t1} = K_{t2} = 10 \text{ Nm/A}$$

$$R_1 = R_2 = 25 \text{ vs} \quad R_1 = R_2 = 1$$

$${}^{i-1}A_i = \text{Rotz}(\theta_i) T(0,0,d_i) T(a_i,0,0) \text{Rotx}(\alpha_i)$$

## Modelo basado en objeto *robot*

- Ejemplo: Robot plano RR en vertical

```

26/09/14 22:09 D:\DATOS\clases\CPR 461\Robot XRW RT... 1 of 2
% ejemplo con bobinas twobars y para la robot 88
% plano en vertical con gravedad en el eje Y
clear

% Parametros del Robot (Robot de dos grados de libertad NN)
% Longitudes de los eslabones [m]
L1=1; L2=1;

% Masas de los eslabones [kg]
m1=3; m2=3;

% Momentos del centro de gravedad respecto
% al eje vertical de referencia local
J1=0.5; J2=0.5;

% Teoría respuesta a ejes paralelos al marco de referencia
% (seg N°2)
l1=0.1336; l2=0.1; l1p=0.1; l1p=0; l1=0; l1p=0;
l2=0.1336; l2=0.1; l2p=0.1; l2p=0; l2=0; l2p=0;

% Teoría de los motores (seg N°2)
Jm1=0.025; Jm2=0.025;

% Posición máxima de las motoras (mm / rad/s)
lim1=3.6e-4; lim2=3.6e-4;

% Factor de reducción del motor
N1=5; N2=5;

% Aceleración de la gravedad
g = -9.8;

% Parámetros de Denavit-Hartenberg DH = (TIERA O A ALTOS SIGNO OFFSET)
% DH1= [a1 d1 a2 d2]
% DH2= [a3 d3 a4 d4]
% DH3= [a5 d5 a6 d6]
% DH4= [a7 d7 a8 d8]
% DH5= [a9 d9 a10 d10]
% DH6= [a11 d11 a12 d12]
% DH7= [a13 d13 a14 d14]
% DH8= [a15 d15 a16 d16]
% DH9= [a17 d17 a18 d18]
% DH10= [a19 d19 a20 d20]
% DH11= [a21 d21 a22 d22]
% DH12= [a23 d23 a24 d24]
% DH13= [a25 d25 a26 d26]
% DH14= [a27 d27 a28 d28]
% DH15= [a29 d29 a30 d30]
% DH16= [a31 d31 a32 d32]
% DH17= [a33 d33 a34 d34]
% DH18= [a35 d35 a36 d36]
% DH19= [a37 d37 a38 d38]
% DH20= [a39 d39 a40 d40]
% DH21= [a41 d41 a42 d42]
% DH22= [a43 d43 a44 d44]
% DH23= [a45 d45 a46 d46]
% DH24= [a47 d47 a48 d48]
% DH25= [a49 d49 a50 d50]
% DH26= [a51 d51 a52 d52]
% DH27= [a53 d53 a54 d54]
% DH28= [a55 d55 a56 d56]
% DH29= [a57 d57 a58 d58]
% DH30= [a59 d59 a60 d60]
% DH31= [a61 d61 a62 d62]
% DH32= [a63 d63 a64 d64]
% DH33= [a65 d65 a66 d66]
% DH34= [a67 d67 a68 d68]
% DH35= [a69 d69 a70 d70]
% DH36= [a71 d71 a72 d72]
% DH37= [a73 d73 a74 d74]
% DH38= [a75 d75 a76 d76]
% DH39= [a77 d77 a78 d78]
% DH40= [a79 d79 a80 d80]
% DH41= [a81 d81 a82 d82]
% DH42= [a83 d83 a84 d84]
% DH43= [a85 d85 a86 d86]
% DH44= [a87 d87 a88 d88]
% DH45= [a89 d89 a90 d90]
% DH46= [a91 d91 a92 d92]
% DH47= [a93 d93 a94 d94]
% DH48= [a95 d95 a96 d96]
% DH49= [a97 d97 a98 d98]
% DH50= [a99 d99 a100 d100]
% DH51= [a101 d101 a102 d102]
% DH52= [a103 d103 a104 d104]
% DH53= [a105 d105 a106 d106]
% DH54= [a107 d107 a108 d108]
% DH55= [a109 d109 a110 d110]
% DH56= [a111 d111 a112 d112]
% DH57= [a113 d113 a114 d114]
% DH58= [a115 d115 a116 d116]
% DH59= [a117 d117 a118 d118]
% DH60= [a119 d119 a120 d120]
% DH61= [a121 d121 a122 d122]
% DH62= [a123 d123 a124 d124]
% DH63= [a125 d125 a126 d126]
% DH64= [a127 d127 a128 d128]
% DH65= [a129 d129 a130 d130]
% DH66= [a131 d131 a132 d132]
% DH67= [a133 d133 a134 d134]
% DH68= [a135 d135 a136 d136]
% DH69= [a137 d137 a138 d138]
% DH70= [a139 d139 a140 d140]
% DH71= [a141 d141 a142 d142]
% DH72= [a143 d143 a144 d144]
% DH73= [a145 d145 a146 d146]
% DH74= [a147 d147 a148 d148]
% DH75= [a149 d149 a150 d150]
% DH76= [a151 d151 a152 d152]
% DH77= [a153 d153 a154 d154]
% DH78= [a155 d155 a156 d156]
% DH79= [a157 d157 a158 d158]
% DH80= [a159 d159 a160 d160]
% DH81= [a161 d161 a162 d162]
% DH82= [a163 d163 a164 d164]
% DH83= [a165 d165 a166 d166]
% DH84= [a167 d167 a168 d168]
% DH85= [a169 d169 a170 d170]
% DH86= [a171 d171 a172 d172]
% DH87= [a173 d173 a174 d174]
% DH88= [a175 d175 a176 d176]
% DH89= [a177 d177 a178 d178]
% DH90= [a179 d179 a180 d180]
% DH91= [a181 d181 a182 d182]
% DH92= [a183 d183 a184 d184]
% DH93= [a185 d185 a186 d186]
% DH94= [a187 d187 a188 d188]
% DH95= [a189 d189 a190 d190]
% DH96= [a191 d191 a192 d192]
% DH97= [a193 d193 a194 d194]
% DH98= [a195 d195 a196 d196]
% DH99= [a197 d197 a198 d198]
% DH100= [a199 d199 a200 d200]
% DH101= [a201 d201 a202 d202]
% DH102= [a203 d203 a204 d204]
% DH103= [a205 d205 a206 d206]
% DH104= [a207 d207 a208 d208]
% DH105= [a209 d209 a210 d210]
% DH106= [a211 d211 a212 d212]
% DH107= [a213 d213 a214 d214]
% DH108= [a215 d215 a216 d216]
% DH109= [a217 d217 a218 d218]
% DH110= [a219 d219 a220 d220]
% DH111= [a221 d221 a222 d222]
% DH112= [a223 d223 a224 d224]
% DH113= [a225 d225 a226 d226]
% DH114= [a227 d227 a228 d228]
% DH115= [a229 d229 a230 d230]
% DH116= [a231 d231 a232 d232]
% DH117= [a233 d233 a234 d234]
% DH118= [a235 d235 a236 d236]
% DH119= [a237 d237 a238 d238]
% DH120= [a239 d239 a240 d240]
% DH121= [a241 d241 a242 d242]
% DH122= [a243 d243 a244 d244]
% DH123= [a245 d245 a246 d246]
% DH124= [a247 d247 a248 d248]
% DH125= [a249 d249 a250 d250]
% DH126= [a251 d251 a252 d252]
% DH127= [a253 d253 a254 d254]
% DH128= [a255 d255 a256 d256]
% DH129= [a257 d257 a258 d258]
% DH130= [a259 d259 a260 d260]
% DH131= [a261 d261 a262 d262]
% DH132= [a263 d263 a264 d264]
% DH133= [a265 d265 a266 d266]
% DH134= [a267 d267 a268 d268]
% DH135= [a269 d269 a270 d270]
% DH136= [a271 d271 a272 d272]
% DH137= [a273 d273 a274 d274]
% DH138= [a275 d275 a276 d276]
% DH139= [a277 d277 a278 d278]
% DH140= [a279 d279 a280 d280]
% DH141= [a281 d281 a282 d282]
% DH142= [a283 d283 a284 d284]
% DH143= [a285 d285 a286 d286]
% DH144= [a287 d287 a288 d288]
% DH145= [a289 d289 a290 d290]
% DH146= [a291 d291 a292 d292]
% DH147= [a293 d293 a294 d294]
% DH148= [a295 d295 a296 d296]
% DH149= [a297 d297 a298 d298]
% DH150= [a299 d299 a300 d300]
% DH151= [a301 d301 a302 d302]
% DH152= [a303 d303 a304 d304]
% DH153= [a305 d305 a306 d306]
% DH154= [a307 d307 a308 d308]
% DH155= [a309 d309 a310 d310]
% DH156= [a311 d311 a312 d312]
% DH157= [a313 d313 a314 d314]
% DH158= [a315 d315 a316 d316]
% DH159= [a317 d317 a318 d318]
% DH160= [a319 d319 a320 d320]
% DH161= [a321 d321 a322 d
```

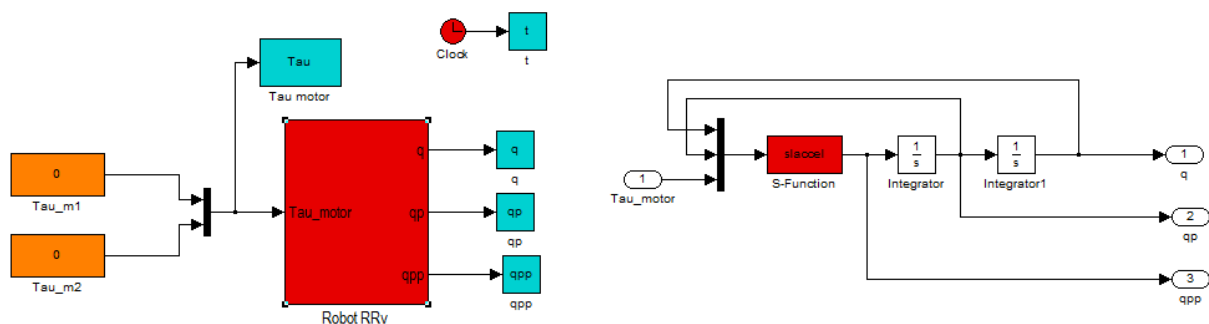
El modelo se puede hacer con variables simbólicas (utilizando *Symbolic Toolbox* de *Matlab*), pero la ejecución en Simulink es extremadamente lenta.

Si se recomienda el uso de variables simbólicas con *Robotics Toolbox* para el cálculo de la *cinemática directa*.

```
syms x y z q1 q2 real
...
Robot.fkine([q1 q2])
```

## Modelo basado en objeto *robot*

- Ejemplo: Robot plano RR en vertical



Basada en la *S-function* **slaccel** de *Robotics Toolbox*  
(ejecución lenta)

### Condiciones iniciales en los integradores

# Ecuaciones para el modelado

## Formulación de Lagrange

- Basado en balance energético

$$\begin{cases} L = E_c - E_p \\ \tau_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \end{cases}$$

## Formulación de Newton-Euler

- Basado en la segunda ley de Newton
- Además de ecuaciones de movimiento, aporta esfuerzos en las articulaciones

$$\begin{cases} \sum \mathbf{F} = \frac{d}{dt} (m \mathbf{v}) \\ \sum \mathbf{T} = \frac{d}{dt} (\mathbf{I} \boldsymbol{\omega}) = \mathbf{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega}) \end{cases}$$

**Obtención de ecuaciones analíticas con variables simbólicas (*Symbolic Toolbox*)**

# Ecuaciones para el modelado

- Algoritmo de Newton-Euler:

Basado en la 2ª ley de Newton  $\Sigma \mathbf{F} = m\mathbf{a}$

Es un procedimiento recursivo en el que para

$i=1, \dots, n$  se calcula:

$\mathbf{W}_i$  (Velocidades angular del sistema  $\{S_i\}$ :  $\dot{q}_i'$ )

$\mathbf{W}_i'$  (Aceleraciones angular del sistema  $\{S_i\}$ :  $\ddot{q}_i''$ )

$\mathbf{V}_i', \mathbf{a}_i$  (Aceleraciones lineales del sistema  $\{S_i\}$  y del centro de masa de la barra  $i$ )

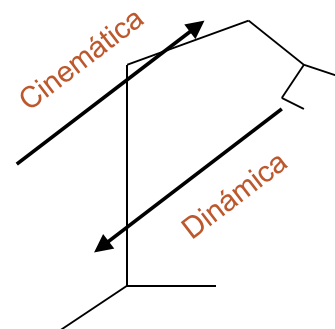
$i=n, \dots, 1$  se calcula:

$\mathbf{F}_i$  (Fuerza ejercida en el centro de masa de la barra  $i$ )

$\mathbf{N}_i$  (Par ejercido en el centro de masa de la barra  $i$ )

$\mathbf{f}_i$  (Fuerza ejercida sobre la articulación  $i-1$ , unión barra  $i-1$  con barra  $i$ , expresada en  $\{i-1\}$ )

$\mathbf{n}_i$  (Par ejercido sobre la articulación  $i-1$ , unión barra  $i-1$  con barra  $i$ , expresada en  $\{i-1\}$ )



# Ecuaciones para el modelado

## ◦ Pasos del algoritmo de Newton-Euler:

- 1.- Asignar a cada eslabón un sistema de referencia de acuerdo con las normas de D-H
- 2.- Establecer las condiciones iniciales:

### Base del robot

${}^0\omega_0$ : velocidad angular= $[0,0,0]^T$

${}^0\dot{\omega}_0$ : aceleración angular= $[0,0,0]^T$

${}^0v_0$ : velocidad lineal= $[0,0,0]^T$

${}^0\dot{v}_0$ : aceleración lineal= $[-g_{x0}, g_{y0}, g_{z0}]^T$

${}^0\omega_0, {}^0\dot{\omega}_0$  y  ${}^0v_0$  son típicamente nulos salvo que la base del robot esté en movimiento

$[g_{x0}, g_{y0}, g_{z0}]$  es el vector de gravedad expresado en el sistema  $\{S_0\}$  (habitualmente toma el valor  $[0, 0, -9.8]$  pues  $z_0$  se sitúa vertical hacia arriba).

### Extremo del robot

Para el extremo del robot se conocerá la fuerza y el par ejercidos externamente  ${}^{n+1}f_{n+1}$  y  ${}^{n+1}n_{n+1}$

# Ecuaciones para el modelado

## ◦ Pasos del algoritmo de Newton-Euler:

- 2.- Establecer las condiciones iniciales (continuación):

### Configuración del robot

$z_0 = [0, 0, 1]^T$

${}^i p_i$  = Vector que une el origen de  $\{S_{i-1}\}$  con el de  $\{S_i\}$  expresadas en  $\{S_i\} = [a_i, d_i \sin(\alpha_i), d_i \cos(\alpha_i)]$

${}^i s_i$  = Coordenadas del centro de masas del eslabón  $i$  respecto del sistema  $\{S_i\}$

${}^i I_i$  = Matriz de inercia del eslabón  $i$  expresado en un sistema paralelo al  $\{S_i\}$  y con el origen en el centro de masas del eslabón.

- 3.- Obtener las matrices de rotación  ${}^{i-1}R_i$  y sus inversas

$${}^i R_{i-1} = ({}^{i-1} R_i)^{-1} = ({}^{i-1} R_i)^T \quad {}^{i-1} R_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i \\ 0 & S\alpha_i & C\alpha_i \end{bmatrix}$$

# Ecuaciones para el modelado

## ◦ Pasos del algoritmo de Newton-Euler:

Desde  $i=1$  hasta  $n$  (iteración cinemática)

- 4.- Obtener la velocidad angular del sistema  $\{S_i\}$  :

$${}^i\omega_i = \begin{cases} {}^iR_{i-1} \left( {}^{i-1}\omega_{i-1} + z_0 \dot{q}_i \right) & \text{si el eslabón } i \text{ es de rotación} \\ {}^iR_{i-1} {}^{i-1}\omega_{i-1} & \text{si el eslabón } i \text{ es de traslación} \end{cases}$$

- 5.- Obtener la aceleración angular del sistema  $\{S_i\}$

$${}^i\dot{\omega}_i = \begin{cases} {}^iR_{i-1} \left( {}^{i-1}\dot{\omega}_{i-1} + z_0 \ddot{q}_i + {}^{i-1}\omega_{i-1} \times z_0 \dot{q}_i \right) & \text{si el eslabón } i \text{ es de rotación} \\ {}^iR_{i-1} {}^{i-1}\dot{\omega}_{i-1} & \text{si el eslabón } i \text{ es de traslación} \end{cases}$$

- 6.- Obtener la aceleración lineal del sistema  $\{S_i\}$

$${}^i\dot{v}_i = \begin{cases} {}^i\dot{\omega}_i \times {}^i p_i + {}^i\omega_i \times ({}^i\omega_i \times {}^i p_i) + {}^iR_{i-1} {}^{i-1}\dot{v}_{i-1} & \text{si el eslabón } i \text{ es de rotación} \\ {}^iR_{i-1} (z_0 \ddot{q}_i + {}^{i-1}\dot{v}_{i-1}) + {}^i\dot{\omega}_i \times {}^i p_i + 2 {}^i\omega_i \times {}^iR_{i-1} z_0 \dot{q}_i + \\ \quad + {}^i\omega_i \times ({}^i\omega_i \times {}^i p_i) & \text{si el eslabón } i \text{ es de traslación} \end{cases}$$

# Ecuaciones para el modelado

## ◦ Pasos del algoritmo de Newton-Euler:

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

$${}^i a_i = {}^i\dot{\omega}_i \times {}^i s_i + {}^i\omega_i \times ({}^i\omega_i \times {}^i s_i) + {}^i\dot{v}_i$$

Fin iteración cinemática

Desde  $i=n$  hasta 1 (iteración dinámica)

- 8.- Obtener la fuerza ejercida sobre el eslabón  $i$

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

- 9.- Obtener el par ejercido sobre el eslabón  $i$

$${}^i n_i = {}^iR_{i+1} \left[ {}^{i+1}n_{i+1} + ({}^{i+1}R_i {}^i p_i) \times {}^{i+1}f_{i+1} \right] + ({}^i p_i + {}^i s_i) \times m_i {}^i a_i + {}^i I_i {}^i\dot{\omega}_i + {}^i\omega_i \times ({}^i I_i {}^i\omega_i)$$

- 10.- Obtener la fuerza o par aplicado sobre la articulación

$$\tau_i = \begin{cases} {}^i n_i^T {}^iR_{i-1} z_0 & \text{si el eslabón } i \text{ es de rotación} \\ {}^i f_i^T {}^iR_{i-1} z_0 & \text{si el eslabón } i \text{ es de traslación} \end{cases}$$

Fin iteración dinámica



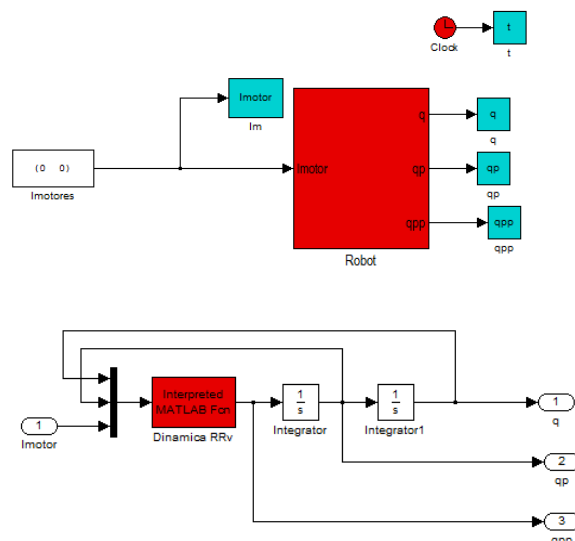
# Ecuaciones para el modelado

- Ejemplo: Robot plano RR en vertical

[illegible][illegible]

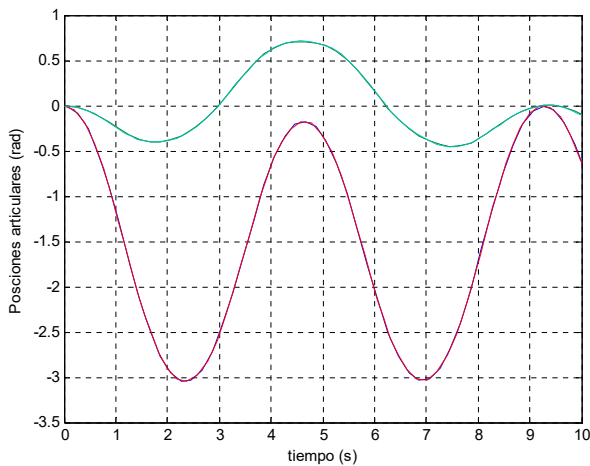
## Implementación del modelo

- Utilizar en Simulink una función .m (*MATLAB function*) para simular modelo dinámico directo (similar a *slaccel*):

[illegible]

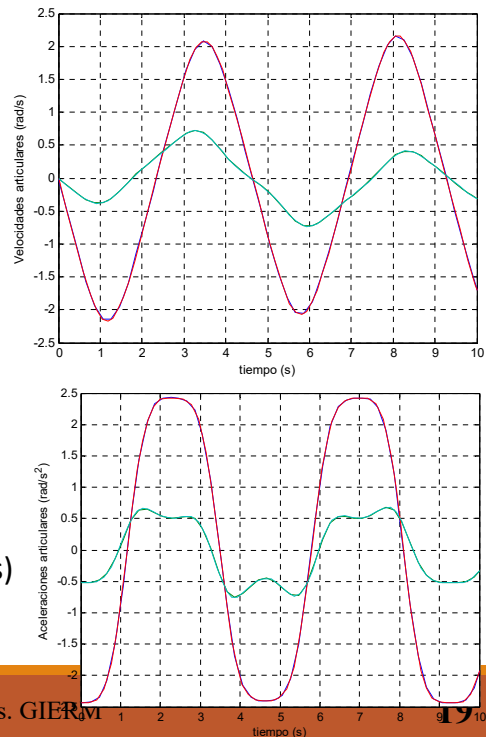
# Implementación del modelo

- Comparar resultados entre los proporcionados por el *Robotics Toolbox* y los implementados en *MATLAB Function*



Cuidado con las comparativas:

- Señales de control distintas (pares vs. Intensidades)
- Condiciones iniciales en los integradores



Control y Programación de Robots. GIERM

19

## Linealización de modelos

- Ecuación dinámica del robot:

$$\tau = RK_t I_m = (M(q) + R^2 J_m) \ddot{q} + (C(q, \dot{q}) + R^2 B_m) \dot{q} + G(q) + F(q, \dot{q})$$

$$I_m = \underbrace{(RK_t)^{-1} (M(q) + R^2 J_m)}_{M_A(q)} \ddot{q} + \underbrace{(RK_t)^{-1} (C(q, \dot{q}) + R^2 B_m)}_{V_A(q, \dot{q})} \dot{q} + \underbrace{(RK_t)^{-1} G(q)}_{G_A(q)} + \underbrace{(RK_t)^{-1} F(q, \dot{q})}_{F_A(q, \dot{q})}$$

- Consideraciones para la linealización:

- Linealizar en torno a velocidades nulas:

- Velocidades de equilibrio:

$$\dot{q}_{eq} = 0 \text{ unidades velocidad articular}$$

$$\dot{q} = \dot{q}_{eq} + \Delta \dot{q} = \Delta \dot{q}$$

- Aceleraciones de equilibrio:

$$\ddot{q}_{eq} = 0 \text{ unidades aceleración articular}$$

$$\ddot{q} = \ddot{q}_{eq} + \Delta \ddot{q} = \Delta \ddot{q}$$

# Linealización de modelos

- **Linealización de términos de la matriz de inercia junto con la aceleración:**

- $$M_A(q)\ddot{q} = \underbrace{M_A(q_{eq})}_{=0}\ddot{q}_{eq} + \underbrace{\ddot{q}_{eq}}_{=0} \frac{\partial M_A(q)}{\partial q} \bigg|_{q=q_{eq}} \Delta q + M_A(q_{eq})\Delta\ddot{q} + \dots \approx M_A(q_{eq})\Delta\ddot{q}$$

- Despreciar términos no diagonales de la matriz de inercia frente a los diagonales:

$$M_A(q_{eq}) \approx \text{diag}(M_A(q_{eq}))$$

- Especialmente cierto cuando los factores de reducción son altos ( $R \gg 1$ )
- Elección valores de inercia máximos de los términos de la diagonal de  $M_A(q)$
- Con estas suposiciones se está desacoplando el sistema (a nivel de modelo teórico)

$$G_{ij}(s) = \frac{\Delta q_i(s)}{\Delta \tau_j(s)} \quad G_{ij}(s) = 0 \quad \text{si } i \neq j$$

# Linealización de modelos

- **Linealización de términos centrípetos y de Coriolis:**

- Términos de Coriolis en el equilibrio:

$$C(q, \dot{q}_{eq}) = C(q, 0) = 0$$

- No hay términos lineales aportados por  $V(q, \dot{q}) = C(q, \dot{q})\dot{q}$  en la linealización, ya que es cuadrático con la velocidad articular:

$$\begin{aligned} V(q, \dot{q}) = C(q, \dot{q})\dot{q} &= \underbrace{C(q_{eq}, \dot{q}_{eq})}_{=0} \underbrace{\dot{q}_{eq}}_{=0} + \\ &+ \left( \frac{\partial C(q, \dot{q})}{\partial q} \Delta q + \frac{\partial C(q, \dot{q})}{\partial \dot{q}} \Delta \dot{q} \right) \underbrace{\dot{q}_{eq}}_{=0} + \underbrace{C(q_{eq}, \dot{q}_{eq})}_{=0} \Delta q + \dots \approx 0 \end{aligned}$$

- Los términos de la fricción viscosa ( $R^2 B_m \dot{q} = R^2 B_m \dot{q}_{eq} + R^2 B_m \Delta \dot{q}$ ) sí aporta términos en la linealización.

# Linealización de modelos

## ◦ Consideraciones sobre los términos gravitatorios:

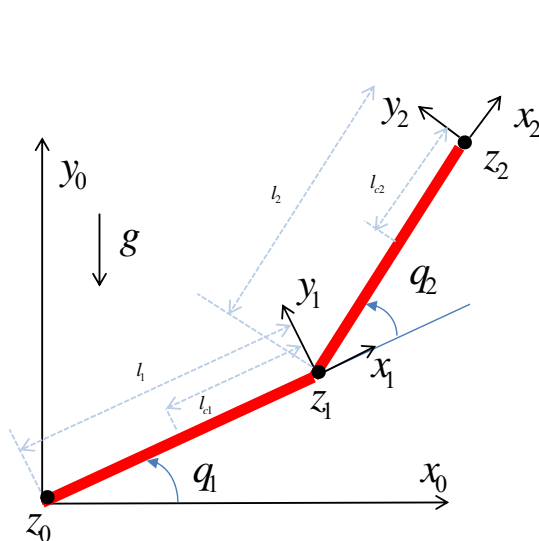
- En técnicas de control básico, no se suele linealizar en torno a ninguna posición en particular.
- Los términos gravitatorios se desprecian en los modelos incrementales, y se consideran desde el punto de vista de control como perturbaciones mantenidas a la entrada.
- La magnitud de estas perturbaciones decrece a medida que las relaciones de las reductoras son mayores (los términos gravitatorios no se magnifican con las reductoras por las reductoras).
- Posibilidad de compensarlos con un *Feedforward*.

## ◦ Consideraciones sobre los términos de fricción:

- Se desprecian en la linealización:  $F(q, \dot{q}) \approx 0$
- Posibilidad de compensarlos con Feedforward o estructura interna con Feedback.

# Linealización de modelos

## ◦ Ejemplo: Robot RR vertical:



$${}^{i-1}\mathbf{A}_i = \mathbf{Rotz}(\theta_i) \mathbf{T}(0,0,d_i) \mathbf{T}(a_i,0,0) \mathbf{Rotx}(\alpha_i)$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

### Eslabones

$$m_1 = m_2 = 3 \text{ Kg} \quad l_1 = l_2 = 1 \text{ m}$$
$$I_1 = I_2 = 0.2536 \text{ Kg.m}^2 \quad l_{c1} = l_{c2} = 0.5 \text{ m}$$

### Motores

$$J_{m1} = J_{m2} = 0.025 \text{ Kg.m}^2$$
$$B_{m1} = B_{m2} = 3.6 \times 10^{-6} \text{ Nm/(rad/s)}$$
$$R_1 = R_2 = 25 \text{ vs} \quad R_1 = R_2 = 1$$

# Linealización del modelo

## ◦ Ejemplo: Robot plano RR en vertical

$$K_t R I_m = (M(q) + J_m R^2) \ddot{q} + (C(q, \dot{q}) + B_m R^2) \dot{q} + G(q) + F(\dot{q})$$

**Brazo mecánico:**

$$M(q) = \begin{bmatrix} 5.0072 + 3.0 \cos(q_2) & 1.0036 + 1.5 \cos(q_2) \\ 1.0036 + 1.5 \cos(q_2) & 1.0036 \end{bmatrix}$$

$$G(q) = g \begin{bmatrix} 4.5 \cos(q_1) + 1.5 \cos(q_1 + q_2) \\ 1.5 \cos(q_1 + q_2) \end{bmatrix}$$

$$C(q, \dot{q}) = 1.5 \sin(q_2) \begin{bmatrix} -2\dot{q}_2 & -\dot{q}_2 \\ \dot{q}_1 & 0 \end{bmatrix}$$

**Actuadores:**

$$K_t R = \begin{bmatrix} 250 & 0 \\ 0 & 250 \end{bmatrix}$$

$$R = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix}$$

$$J_m R^2 = \begin{bmatrix} 15.625 & 0 \\ 0 & 15.625 \end{bmatrix}$$

$$B_m R^2 = \begin{bmatrix} 0.0022 & 0 \\ 0 & 0.0022 \end{bmatrix}$$

$$K_t = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

# Linealización del modelo

## ◦ Ejemplo: Robot plano RR en vertical

$$K_t R I_m = (M(q) + J_m R^2) \ddot{q} + (C(q, \dot{q}) + B_m R^2) \dot{q} + G(q) + F(\dot{q})$$

$$R_1 = R_2 = 25$$

$$\begin{bmatrix} 250 & 0 \\ 0 & 250 \end{bmatrix} \begin{bmatrix} I_{m1} \\ I_{m2} \end{bmatrix} = \begin{bmatrix} 20.6322 + 3.0 \cos(q_2) & 1.0036 + 1.5 \cos(q_2) \\ 1.0036 + 1.5 \cos(q_2) & 16.6286 \end{bmatrix} * \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\ + \begin{bmatrix} -1.5 \sin(q_2) (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) + 0.0022 \dot{q}_1 \\ -1.5 \sin(q_2) \dot{q}_1^2 + 0.0022 \dot{q}_2 \end{bmatrix} + \begin{bmatrix} 44.1 \cos(q_2) + 14.7 \cos(q_1 + q_2) \\ 14.7 \cos(q_1 + q_2) \end{bmatrix}$$

$$R_1 = R_2 = 1$$

$$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} I_{m1} \\ I_{m2} \end{bmatrix} = \begin{bmatrix} 5.0322 + 3.0 \cos(q_2) & 1.0036 + 1.5 \cos(q_2) \\ 1.0036 + 1.5 \cos(q_2) & 1.0286 \end{bmatrix} * \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\ + \begin{bmatrix} -1.5 \sin(q_2) (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) + 3.6 \times 10^{-6} \dot{q}_1 \\ -1.5 \sin(q_2) \dot{q}_1^2 + 3.6 \times 10^{-6} \dot{q}_2 \end{bmatrix} + \begin{bmatrix} 44.1 \cos(q_2) + 14.7 \cos(q_1 + q_2) \\ 14.7 \cos(q_1 + q_2) \end{bmatrix}$$

# Linealización del modelo

- Ejemplo: **Robot plano RR en vertical**

## Modelos lineales (en torno a velocidades bajas):

$$\frac{q_1(s)}{I_{m1}(s)} \approx \frac{250}{(23.6322s + 0.0022)s}$$
$$\frac{q_2(s)}{I_{m2}(s)} \approx \frac{250}{(16.6286 + 0.0022)s}$$

$$R_1 = R_2 = 25$$

**APROXIMACIÓN  
POR SISTEMAS DE  
SEGUNDO ORDEN Y  
DE TIPO 1**

$$\frac{q_1(s)}{I_{m1}(s)} \approx \frac{10}{(8.0322s + 3.6 \times 10^{-6})s}$$
$$\frac{q_2(s)}{I_{m2}(s)} \approx \frac{10}{(1.0286s + 3.6 \times 10^{-6})s}$$

$$R_1 = R_2 = 1$$

## Resumen “Symbolic Toolbox”

- `syms`  
`>> syms q1 q2 qn real`
- `PI = sym('pi');`  
`>> sin(PI), sin(pi)`
- `pretty();`  
`>> x=sin(q1)^2+cos(q1)^2`  
`>> pretty(x)`
- `eval();`  
`>> q1 = pi/4;`  
`>> x`  
`>> eval(x)`
- `simplify();`  
`>> x=sin(qn)^2+cos(qn)^2`  
`>> y=simplify(x)`
- `simple();`  
`>> y=simple(x)`
- `vpa();`  
`>> z=x/3`  
`>> r=vpa(z,4)`
- `collect();`  
`>> x=q2*cos(qn)+q2*sin(qn)`  
`>> y = collect(x,q2)`
- `diff();`  
`>> y=diff(x,q2)`  
`>> z=diff(x,qn)`
- `>> help symbolic`