Modelado dinámico de un tiovivo (2GL, Euler Lagrange)

Este código funciónó con Matlab **`R2022b`**

*Image from jana_lsb0 at Pixabay , Pixabay License 2022 (free even for commercial use)

Presentaciones en vídeo:

https://personales.upv.es/asala/YT/V/tiovEL.html

https://personales.upv.es/asala/YT/V/tiovEL2.html

https://personales.upv.es/asala/YT/V/tiovELsim.html

Objetivos: Comprender el modelado físico de un tiovivo como la figura, y su simulación. En concreto, abordaremos el caso 2 grados de libertad con ecuaciones Euler-Lagrange de la dinámica.

Tabla de Contenidos


```
syms phi theta real %grados de libertad para describir el movimiento
syms R_0 L M g I_0 real
```

```
assume(M>0), assume(g>0), assume(I_0>0), assume(L>0), assume(R_0>0)
%ayuda para symbolic toolbox
```

Cinemática

```
R=R_0+L*sin(theta); %Distancia masa al eje de rotación central
x=R*cos(phi);
y=R*sin(phi);
z=L*(1-cos(theta)); %z=0 es el equilibrio vertical inferior cuando
no se mueve
r=[x;y;z]
```

r =

$$\begin{pmatrix} \cos(\phi)\,(R_0 + L\sin(\theta)) \\ \sin(\phi)\,(R_0 + L\sin(\theta)) \\ -L\,(\cos(\theta) - 1) \end{pmatrix}$$

```
syms omega_0 omega_1 real %vels ang. dot_phi dot_theta
syms alpha_0 alpha_1 real %acels. ddot_phi ddot_theta
q=[phi; theta]; v_q=[omega_0;omega_1]; a_q=[alpha_0;alpha_1];
```

$$v(q,\dot{q}) = \frac{dr}{dt} = \frac{\partial r}{\partial q}\cdot\frac{dq}{dt} = \frac{\partial r}{\partial q}\cdot\dot{q}$$

```
veloc = jacobian(r,q)*v_q %dr/dt
```

veloc =

$$\begin{pmatrix} L\,\omega_1\cos(\phi)\cos(\theta) - \omega_0\sin(\phi)\,(R_0 + L\sin(\theta)) \\ \omega_0\cos(\phi)\,(R_0 + L\sin(\theta)) + L\,\omega_1\cos(\theta)\sin(\phi) \\ L\,\omega_1\sin(\theta) \end{pmatrix}$$

Si quisiéramos calcular la aceleración de la masa suspendida en función de posición, velocidad y aceleraciones angulares, saldría:

$$a(q,\dot{q},\ddot{q}) = \frac{d^2r}{dt^2} = \frac{dv}{dt} = \frac{\partial v}{\partial q}\cdot\dot{q} + \frac{\partial v}{\partial \dot{q}}\cdot\ddot{q} = \begin{pmatrix} \frac{\partial v}{\partial q} & \frac{\partial v}{\partial \dot{q}} \end{pmatrix}\cdot\begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix}$$

```
acel = simplify( jacobian(veloc,[q; v_q])*[v_q; a_q] ) %d2r/dt^2
```

acel =

$$\begin{pmatrix} L\,\alpha_1\cos(\phi)\cos(\theta) - \omega_0\,(\omega_0\cos(\phi)\,\sigma_1 + L\,\omega_1\cos(\theta)\sin(\phi)) - \alpha_0\sin(\phi)\,\sigma_1 - \omega_1\,(L\,\omega_0\cos(\theta)\sin(\phi) + L\,\omega \\ \omega_1\,(L\,\omega_0\cos(\phi)\cos(\theta) - L\,\omega_1\sin(\phi)\sin(\theta)) - \omega_0\,(\omega_0\sin(\phi)\,\sigma_1 - L\,\omega_1\cos(\phi)\cos(\theta)) + \alpha_0\cos(\phi)\,\sigma_1 + L \\ L\,\left(\cos(\theta)\,{\omega_1}^2 + \alpha_1\sin(\theta)\right) \end{pmatrix}$$

where

$$\sigma_1 = R_0 + L\sin(\theta)$$

***Nota:** esta aceleración NO es necesaria para obtener las ecuaciones con la metodología Euler-Lagrange.

Energías y ecs. movimiento Euler-Lagrange

```
T=simplify( 0.5*M*(veloc'*veloc) + 0.5*I_0*omega_0^2, 100) %kinetic energy
```

T =

$$\frac{M\,L^2\,\omega_0{}^2\,\sin(\theta)^2}{2}+\frac{M\,L^2\,\omega_1{}^2}{2}+M\,L\,R_0\,\omega_0{}^2\,\sin(\theta)+\frac{M\,R_0{}^2\,\omega_0{}^2}{2}+\frac{I_0\,\omega_0{}^2}{2}$$

```
V=M*g*z %potential energy
```

V = $-L\,M\,g\,(\cos(\theta)-1)$

```
Lagrangian=T-V;
```

Calculemos $p(q,\dot{q})=\dfrac{\partial L}{\partial \dot{q}}$:

```
p = simplify( jacobian(Lagrangian,v_q) ) %momentum generalizado
```

p = $\left(\omega_0\,\left(M\,L^2\,\sin(\theta)^2+2\,M\,L\,R_0\,\sin(\theta)+M\,R_0{}^2+I_0\right)\quad L^2\,M\,\omega_1\right)$

Las ecuaciones Euler Lagrange para cada coordenada q_i son:

$$\frac{dp_i}{dt}-\frac{\partial L}{\partial q_i}=\tau_i$$

donde, aplicando regla de la cadena:

$$\frac{dp_i}{dt}=\frac{\partial p_i}{\partial q}\cdot\dot{q}+\frac{\partial p_i}{\partial \dot{q}}\cdot\ddot{q}=\begin{pmatrix}\frac{\partial p_i}{\partial q} & \frac{\partial p_i}{\partial \dot{q}}\end{pmatrix}\cdot\begin{pmatrix}\dot{q}\\ \ddot{q}\end{pmatrix}$$

```
dpdt = simplify(jacobian(p,[q;v_q])*[v_q;a_q]);
syms tau real %par motor en tiovivo
%pongo en "columna" cada ec. movimiento q_i
EcsMovimiento = (dpdt - jacobian(Lagrangian,q)'==[tau;0]) %Euler-Lagrange
```

EcsMovimiento =

$$\begin{pmatrix}\alpha_0\,\left(M\,L^2\,\sin(\theta)^2+2\,M\,L\,R_0\,\sin(\theta)+M\,R_0{}^2+I_0\right)+L\,M\,\omega_0\,\omega_1\,\left(2\,R_0\cos(\theta)+2\,L\cos(\theta)\sin(\theta)\right)=\tau\\ -M\cos(\theta)\sin(\theta)\,L^2\,\omega_0{}^2+M\,\alpha_1\,L^2-M\,R_0\cos(\theta)\,L\,\omega_0{}^2+M\,g\sin(\theta)\,L=0\end{pmatrix}$$

Que podemos reescribir como $M(q)\ddot{q}=\tau-C(q,\dot{q})\dot{q}+G(q)$

$$(I_0 + M(R_0 + L\sin\theta)^2)\cdot\alpha_0 = \tau - 2M(R_0 + L\sin\theta)L\cos\theta\cdot\omega_0\omega_1 \text{ %plataforma}$$

$$ML^2\cdot\alpha_1 = -MgL\sin\theta + M\omega_0^2(R_0 + L\sin\theta)L\cos\theta \quad \text{%masa suspendida}$$

```
MatrizDeMasa = simplify(jacobian(lhs(EcsMovimiento),a_q))
```

```
MatrizDeMasa =
```

$$\begin{pmatrix} M\,L^2\sin(\theta)^2 + 2\,M\,L\,R_0\sin(\theta) + M\,{R_0}^2 + I_0 & 0 \\ 0 & L^2\,M \end{pmatrix}$$

Casos particulares

1.- Longitud suspendida cero

```
simplify(subs(EcsMovimiento,L,0))
```

```
ans =
```

$$\begin{pmatrix} \alpha_0\,\left(M\,{R_0}^2 + I_0\right) = \tau \\ \text{symtrue} \end{pmatrix}$$

queda un modelo de 1GL con inercia equivalente $I_0 + MR_0^2$.

2.- R0=0, I0=0, tau=0, péndulo articulado esférico

```
simplify(subs(EcsMovimiento,{R_0,I_0,tau},{0,0,0}))
```

```
ans =
```

$$\begin{pmatrix} \frac{\theta}{\pi} \in \mathbb{Z} \vee \alpha_0\sin(\theta) + 2\,\omega_0\,\omega_1\cos(\theta) = 0 \\ L\,{\omega_0}^2\sin(2\,\theta) = 2\,L\,\alpha_1 + 2\,g\sin(\theta) \end{pmatrix}$$

3.- Oscilación columpio con velocidad tiovivo "forzada"

Es el modelo visto en otros vídeos:

$$ML^2\alpha_1 = -MgL\sin\theta + M\omega_0^2(R_0 + L\sin\theta)L\cos\theta,$$

par causado por peso y fuerza centrífuga, mirando sólo la segunda ecuación de movimiento. Si fijamos $\omega_0(t)$ forzándolo a valer la trayectoria de velocidad de referencia que se desee, la primera ecuación de movimiento daría, sustituyendo $\alpha_0 = \frac{d\omega_0}{dt}$, el par en función del tiempo necesario para mantener el perfil de velocidad deseado.

Este tipo de cálculo está detras de metodologías de "par calculado" en control dinámico de robots.

Forma Normalizada (ecs. en variables de estado)

```
Estado=[phi; theta; omega_0; omega_1];
acels=solve(EcsMovimiento,a_q); %despejar aceleraciones
EcEstado4=simplify([omega_0;omega_1;acels.alpha_0;acels.alpha_1])
```

```
EcEstado4 =
```

$$\begin{pmatrix} \omega_0 \\ \omega_1 \\ -\dfrac{2\,M\,\omega_0\,\omega_1\cos(\theta)\sin(\theta)\,L^2 + 2\,M\,R_0\,\omega_0\,\omega_1\cos(\theta)\,L - \tau}{M\,L^2\sin(\theta)^2 + 2\,M\,L\,R_0\sin(\theta) + M\,{R_0}^2 + I_0} \\ \dfrac{R_0\,{\omega_0}^2\cos(\theta) - g\sin(\theta) + L\,{\omega_0}^2\cos(\theta)\sin(\theta)}{L} \end{pmatrix}$$