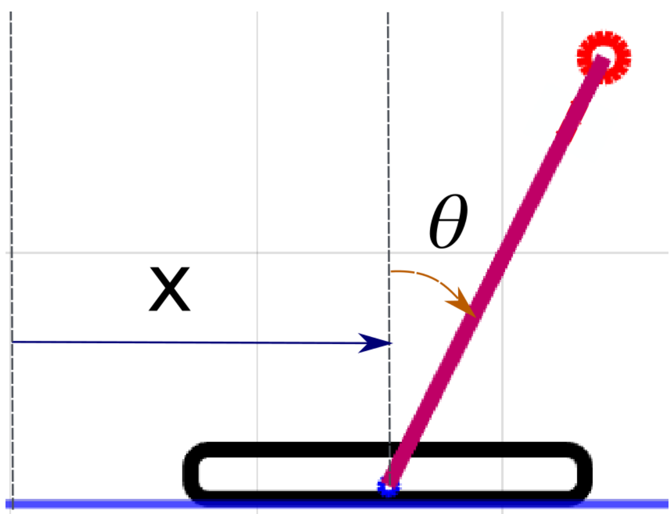
Modelado de sistema bola-carrito con aceleración carrito forzada (reducción de dinámica, sistema reonómico)

Presentación en vídeo: http://personales.upv.es/asala/YT/V/cpforced.html

Este código ejecutó sin errores en Matlab `R2021a`

Objetivo: Modelar las ecuaciones dinámicas de cuarto orden de un sistema bola-carrito de 2 grados de libertad, si uno de los grados pasa a ser forzado a seguir un movimiento $f(t)$ prefijado.

Tabla de Contenidos



Cinemática

```
syms x theta real %señales (funciones del tiempo)
syms L M m g b_x real %parámetros constantes del sistema
%L: distancia carro-bola
%M: masa carro
%m: masa bola
%g: cte. gravedad
%b_x: rozamiento deslizamiento x
pos_bola=[x+L*sin(theta); L*cos(theta)]; %theta=0, equilibrio superior (inestable), pos
```

Como los sistemas mecánicos suelen ser sistemas algebraico-diferenciales de "índice superior", debemos derivar dos veces las ecuaciones que relacionan las "posiciones" para obtener relaciones que involucren a velocidades y aceleraciones.

```
syms v_x v_theta a_x a_theta real %velocidades y aceleraciones
vel_bola=jacobian(pos_bola,[theta x])*[v_theta; v_x] %Regla de la cadena
```

vel_bola =

$$\begin{pmatrix} v_x + L\, v_\theta \cos(\theta) \\ -L\, v_\theta \sin(\theta) \end{pmatrix}$$

```
%acel_bola=jacobian(vel_bola,[theta x v_theta v_x])*[v_theta; v_x; a_theta; a_x] %Regla
```

Preliminares: dinámica con 2GL (objetivo de otros vídeos)

Ecuaciones de la Dinámica mediante la formulación Euler-Lagrange

*No consideraremos inercia rotacional de la masa que cuelga del péndulo.

```
T=0.5*m*(vel_bola'*vel_bola) + 0.5*M*v_x^2 %E. cinética
```

T =

$$\frac{M\, {v_x}^2}{2} + \frac{m\, \left({(v_x + L\, v_\theta \cos(\theta))}^2 + L^2\, {v_\theta}^2 \sin(\theta)^2\right)}{2}$$

```
V=m*g*pos_bola(2) %E. potencial
```

V = $L\, g\, m \cos(\theta)$

```
Lagrangiano=simplify(T-V);
```

Estas son las ecuaciones Euler Lagrange: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q$

Con Jacobian, haremos los 2 grados de libertad simultáneamente en Matlab.

Denominaremos $p := \frac{\partial L}{\partial \dot{q}}$

```
p=jacobian(Lagrangiano,[v_x;v_theta]); %momentum generalizado
```

Fuerzas externas y rozamiento conforman el término Q no conservativo. Supondremos sólamente rozamiento carro-vía en el deslizamiento horizontal.

```
syms F real
Q=[F-b_x*v_x; 0];
```

Con lo que las ecuaciones del movimiento finales quedan: $\frac{dp}{dt} - \frac{\partial L}{\partial q} = Q$

```
EcsMovimientoEL = ...
    jacobian(p,[x;theta; v_x;v_theta])*[v_x;v_theta; a_x;a_theta] ...
                    - jacobian(Lagrangiano,[x;theta])'==Q
```

EcsMovimientoEL =

$$\begin{pmatrix} -L\,m\sin(\theta)\,{v_\theta}^2 + a_x\,(M+m) + L\,a_\theta\,m\cos(\theta) = F - b_x\,v_x \\ L^2\,a_\theta\,m + L\,a_x\,m\cos(\theta) - L\,g\,m\sin(\theta) = 0 \end{pmatrix}$$

Despejamos aceleraciones para poder simular.

```
sol1=solve(EcsMovimientoEL,a_x,a_theta);
a_xEL=simplify(sol1.a_x)
```

a_xEL =

$$\frac{L\,m\sin(\theta)\,{v_\theta}^2 + F - b_x\,v_x - g\,m\cos(\theta)\sin(\theta)}{-m\cos(\theta)^2 + M + m}$$

```
a_thetaEL=simplify(sol1.a_theta)
```

a_thetaEL =

$$\frac{-L\,m\cos(\theta)\sin(\theta)\,{v_\theta}^2 - F\cos(\theta) + b_x\,v_x\cos(\theta) + g\,m\sin(\theta) + M\,g\sin(\theta)}{L\,(-m\cos(\theta)^2 + M + m)}$$

Dinámica con posición x (+ velocidad, aceleración) del carro forzada

Método 1: fuerza/par calculado, reducción de la dinámica

Si calculamos la fuerza necesaria para mover el carro con una cierta trayectoria

```
syms t real
caso_generico=false;
if caso_generico
    syms x_forzada(t) %función del tiempo genérica
else
    x_forzada(t)=t^2+4*t;
end
v_x_forzada=diff(x_forzada) %no se hace jacobian, porque sólo depende de 1 vble (t).
```

v_x_forzada(t) = $2\,t+4$

```
a_x_forzada=diff(v_x_forzada)
```

a_x_forzada(t) = 2

```
solForzada=solve(EcsMovimientoEL,F,a_theta)
```

```
solForzada = struct with fields:
          F: [1×1 sym]
    a_theta: [1×1 sym]
```

```
F_forzada=simplify(solForzada.F)
```

F_forzada = $-L\, m \sin(\theta)\, v_\theta^2 - a_x\, m \cos(\theta)^2 + g\, m \sin(\theta) \cos(\theta) + M\, a_x + a_x\, m + b_x\, v_x$

```
a_th_forzada=simplify(solForzada.a_theta)
```

a_th_forzada =

$$-\frac{a_x \cos(\theta) - g \sin(\theta)}{L}$$

Y sustituiríamos las posiciones, velocidades y aceleraciones prefijadas:

```
subs(F_forzada,{v_x,a_x},{v_x_forzada,a_x_forzada})
```

ans = $-L\, m \sin(\theta)\, v_\theta^2 - 2\, m \cos(\theta)^2 + g\, m \sin(\theta) \cos(\theta) + 2\, M + 2\, m + b_x\, (2\, t + 4)$

```
subs(a_th_forzada,a_x,a_x_forzada)
```

ans =

$$-\frac{2 \cos(\theta) - g \sin(\theta)}{L}$$

La idea de calcular fuerzas para que determinados elementos de un sistema mecánico tengan aceleraciones prefijadas se puede entender como solución de un problema de control; de hecho, en robótica, se denomina "método de par calculado" (porque los actuadores son motores que giran, pero la idea básica es la aquí presentada).

Método 2: Euler-Lagrange con Lagrangiano explícitamente dependiente del tiempo (sistema reonómico)

No es necesario calcular las ecuaciones 2GL y luego la F_forzada.

Euler-Lagrange también funciona con un Lagrangiano dependiente del tiempo:

```
x=x_forzada %para poder copiar-pegar lo de arriba donde la pos. del carro era "x".
```

x(t) = $t^2 + 4\, t$

La cinemática de posiciones y velocidades es:

```
pos_bola=[x+L*sin(theta); L*cos(theta)] %theta=0, equilibrio superior (inestable), posi
```

pos_bola(t) =

$$\begin{pmatrix} t^2+4\,t+L\sin(\theta) \\ L\cos(\theta) \end{pmatrix}$$

```
vel_bola=jacobian(pos_bola,[theta t])*[v_theta; 1] %Regla de la cadena, derivada total.
```

vel_bola(t) =

$$\begin{pmatrix} 2\,t+L\,v_\theta\cos(\theta)+4 \\ -L\,v_\theta\sin(\theta) \end{pmatrix}$$

```
assume(diff(x_forzada(t), t),'Real') % si no, Matlab se "raya" por si resultase ser un
```

La energía cinética (bueno, la que depende del grado de libertad)

```
T=0.5*m*(vel_bola'*vel_bola)% +0.5*M*diff(x)^2 %no hace falta, porque ese grado de libe
```

T(t) =

$$\frac{m\,\left((2\,t+L\,v_\theta\cos(\theta)+4)^2+L^2\,{v_\theta}^2\sin(\theta)^2\right)}{2}$$

La energía potencial es:

```
%pos_bola(2) No está correcto
tmp=pos_bola(t);%necesario para que no se "raye" el symbolic toolbox, tmp ya no es una
V=m*g*tmp(2)
```

V = $L\,g\,m\cos(\theta)$

```
Lagr_forzado=simplify(T-V)
```

Lagr_forzado(t) =

$$\frac{m\,\left((2\,t+L\,v_\theta\cos(\theta)+4)^2+L^2\,{v_\theta}^2\sin(\theta)^2\right)}{2}-L\,g\,m\cos(\theta)$$

Ahora ya hacemos Euler-Lagrange con el grado de libertad theta, como si nada hubiese cambiado:

```
p_forz=diff(Lagr_forzado,v_theta) %momentum generalizado
```

p_forz(t) =

$$\frac{m\,\left(2\,L\cos(\theta)\,(2\,t+L\,v_\theta\cos(\theta)+4)+2\,L^2\,v_\theta\sin(\theta)^2\right)}{2}$$

```
d_p_forz_dt = ...
   jacobian(p_forz,[theta v_theta t])*[v_theta; a_theta; 1];%Regla de la cadena, deriva
```

Y estas son las ecuaciones Euler-Lagrange del movimiento forzado: $\dfrac{dp}{dt}-\dfrac{\partial L}{\partial q}=Q$

```
ecmov_forz= simplify(d_p_forz_dt-diff(Lagr_forzado,theta) ==0) %ecuación del movimiento
```

ecmov_forz(t) = $L = 0 \vee 2\cos(\theta) + L\,a_\theta = g\sin(\theta) \vee m = 0$

```
a_th_forzEL=simplify(solve(ecmov_forz,a_theta))
```

a_th_forzEL =

$$-\frac{2\cos(\theta) - g\sin(\theta)}{L}$$

La ventaja de esta última manipulación es que sólo tenemos que manejar UNA única ecuación del movimiento, evitando tener que calcular las ecuaciones de todos los grados de libertad, y luego "las fuerzas que consiguen forzar una aceleración prefijada dada". También, como en todo Euler-Lagrange, podemos plantear el Lagrangiano sólamente con la cinemática de las velocidades. A cambio, no calculamos esas fuerzas de control que causarían el movimiento prefijado.