Bola en barra "catapulta": ecuaciones del movimiento (2GL, 1GL forzado)

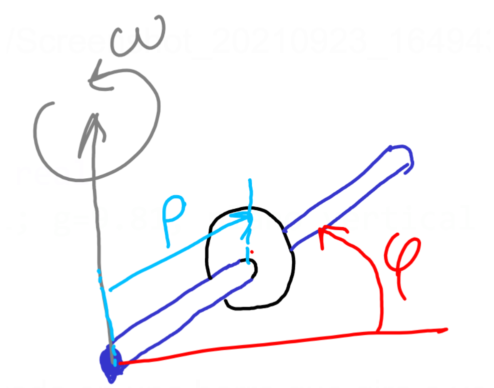
Presentaciones en vídeo:

http://personales.upv.es/asala/YT/V/ballbar2GL.html , http://personales.upv.es/asala/YT/V/ballbar1GL.html .

Este código funcionó correctamente con Matlab **`R2021b`**

Objetivo: obtener las ecuaciones de movimiento de un sistema "bola agujereada ensartada en una barra que gira", con Euler-Lagrange.

Tabla de Contenidos



***Nota:** no vamos a considerar energía potencial de la barra. Suponemos simétrica alrededor del eje de rotación, de modo que su CdG no se mueva. Si no, deberíamos incluir la masa de la barra y la posición de su CdG en la energía potencial en el modelo 2GL. Esto NO es relevante en el caso 1GL con la velocidad de la barra forzada, obviamente, dado que no importa el detalle de peso, inercia, etc. de la barra en el movimiento de la bola ensartada si "alguien" hace las fuerzas necesarias para que gire con la velocidad prefijada ω que queramos.

```
syms rho t real
syms M g omega phi_0 real
%g=0 %plano horizontal; g=9.81, plano vertical
```

Analicemos una bola ensartada en una barra que gira a velocidad constante (1GL, el otro GL sería "forzado").

Cinematica forzada

```
x=rho*cos(omega*t+phi_0); y=rho*sin(omega*t+phi_0); %bola en barra sin sujección radial
syms v_rho a_rho real %velocidades y aceleraciones
r=[x;y] %vector de posición
```

r =

$$\begin{pmatrix} \rho\cos(\phi_0+\omega\,t) \\ \rho\sin(\phi_0+\omega\,t) \end{pmatrix}$$

El único grado de libertad será $q \equiv \rho$.

```
q=rho; v_q=v_rho; a_q=a_rho;
radial_vector=diff(r,q)
```

radial_vector =

$$\begin{pmatrix} \cos(\phi_0+\omega\,t) \\ \sin(\phi_0+\omega\,t) \end{pmatrix}$$

Implementemos cálculo de velocidad $\dfrac{dr}{dt} = \dfrac{\partial r}{\partial q}\dot{q} + \dfrac{\partial r}{\partial t} = [\dfrac{\partial r}{\partial q} \;\; \dfrac{\partial r}{\partial t}] \cdot \begin{pmatrix} \dot{q} \\ 1 \end{pmatrix}$

```
veloc=jacobian(r,[q; t])*[v_q; 1]
```

veloc =

$$\begin{pmatrix} v_\rho\cos(\phi_0+\omega\,t) - \omega\,\rho\sin(\phi_0+\omega\,t) \\ v_\rho\sin(\phi_0+\omega\,t) + \omega\,\rho\cos(\phi_0+\omega\,t) \end{pmatrix}$$

```
veloc_radial=simplify(radial_vector'*veloc)
```

veloc_radial = v_ρ

y, del mismo modo, la regla de la cadena dice que, con $v(q(t),\dot{q}(t),t)$, su derivada temporal es:

$$\frac{dv}{dt} = \frac{\partial v}{\partial q}\dot{q} + \frac{\partial v}{\partial \dot{q}}\ddot{q} + \frac{\partial v}{\partial t}$$

```
acel=simplify(jacobian(veloc,[q; v_q; t])*[v_q; a_q; 1])
```

acel =

$$\begin{pmatrix} a_\rho\cos(\phi_0+\omega\,t) - \omega^2\,\rho\cos(\phi_0+\omega\,t) - 2\,\omega\,v_\rho\sin(\phi_0+\omega\,t) \\ a_\rho\sin(\phi_0+\omega\,t) - \omega^2\,\rho\sin(\phi_0+\omega\,t) + 2\,\omega\,v_\rho\cos(\phi_0+\omega\,t) \end{pmatrix}$$

```
acel_radial=simplify(radial_vector'*acel)
```

acel_radial = $a_\rho - \omega^2 \rho$

Newton: balance de fuerzas

Balance tangencial:

```
Gravedad_radial=radial_vector'*[0;-M*g]
```

Gravedad_radial = $-M\ g \sin(\phi_0 + \omega\ t)$

```
a_qNewton=simplify(solve(radial_vector'*acel==Gravedad_radial/M,a_q))
```

a_qNewton = $\omega^2 \rho - g \sin(\phi_0 + \omega\ t)$

Euler-Lagrange 1GL

```
T=simplify(0.5*M*veloc'*veloc) %cinética, no hace falta cinética de la barra fozada
```

T =

$$\frac{M\ (\omega^2 \rho^2 + {v_\rho}^2)}{2}$$

```
V=M*g*y %potencial
```

V = $M\ g\ \rho \sin(\phi_0 + \omega\ t)$

```
Lagrangian=T-V;
p=diff(Lagrangian,v_q)
```

p = $M\ v_\rho$

```
dpdt=jacobian(p,[q v_q t])*[v_q; a_q; 1]
```

dpdt = $M\ a_\rho$

```
EcmovEL1GL= dpdt - diff(Lagrangian,q) ==0
```

EcmovEL1GL = $M\ a_\rho + M\ g \sin(\phi_0 + \omega\ t) - M\ \omega^2 \rho = 0$

```
a_q_EulerLagrange=simplify(solve(EcmovEL1GL  , a_q))
```

a_q_EulerLagrange = $\omega^2 \rho - g \sin(\phi_0 + \omega\ t)$

Modelado como sistema 2GL con par externo

Cinemática

Barra de inercia "J" alrededor de eje de rotación.

Entonces ρ es un grado de libertad, y su derivada es la v_ρ; ϕ es el segundo grado de libertad, y su derivada es la ω:

```
syms phi J real
syms  a_phi real
x=rho*cos(phi); y=rho*sin(phi);
r=[x;y] %vector de posición
```

r =

$$\begin{pmatrix} \rho\cos(\phi) \\ \rho\sin(\phi) \end{pmatrix}$$

```
q=[rho;phi]; v_q=[v_rho;omega]; a_q=[a_rho;a_phi];
```

Regla de la cadena; dado $r(q(t))$, su derivada: $\frac{dr}{dt} = \frac{\partial r}{\partial q}\dot{q}$

```
veloc=jacobian(r,q)*v_q
```

veloc =

$$\begin{pmatrix} v_\rho\cos(\phi) - \omega\,\rho\sin(\phi) \\ v_\rho\sin(\phi) + \omega\,\rho\cos(\phi) \end{pmatrix}$$

Regla de la cadena; dado $v(q(t), \dot{q}(t))$, su derivada: $\frac{dv}{dt} = \frac{\partial v}{\partial q}\dot{q} + \frac{\partial v}{\partial \dot{q}}\ddot{q} = [\frac{\partial v}{\partial q}\ \ \frac{\partial v}{\partial \dot{q}}] \cdot \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}$

```
acel=simplify(jacobian(veloc,[q; v_q])*[v_q; a_q]) %no hace falta para Euler-Lagrange
```

acel =

$$\begin{pmatrix} -\rho\cos(\phi)\,\omega^2 - 2\,v_\rho\sin(\phi)\,\omega + a_\rho\cos(\phi) - a_\phi\,\rho\sin(\phi) \\ -\rho\sin(\phi)\,\omega^2 + 2\,v_\rho\cos(\phi)\,\omega + a_\rho\sin(\phi) + a_\phi\,\rho\cos(\phi) \end{pmatrix}$$

Energías y ecs. movimiento Euler-Lagrange

```
T=simplify(0.5*M*(veloc'*veloc)+0.5*J*omega^2) %cinética
```

T =

$$\frac{M\,\omega^2\,\rho^2}{2} + \frac{J\,\omega^2}{2} + \frac{M\,{v_\rho}^2}{2}$$

```
V=M*g*y %potencial [¡¡¡CDG barra centro rotación!!!!]
```

V = $M\,g\,\rho\sin(\phi)$

```
Lagrangian=T-V
```

```
Lagrangian =
```

$$\frac{M\,\omega^2\,\rho^2}{2}+\frac{J\,\omega^2}{2}-M\,g\,\sin(\phi)\,\rho+\frac{M\,{v_\rho}^2}{2}$$

```
p=jacobian(Lagrangian,v_q) %momento cinético generalizado
```

p = $\begin{pmatrix} M\,v_\rho & M\,\omega\,\rho^2+J\,\omega \end{pmatrix}$

Regla de la cadena, dado $p(q(t),\dot{q}(t))$, su derivada: $\frac{dp}{dt}=\frac{\partial p}{\partial q}\dot{q}+\frac{\partial p}{\partial \dot{q}}\ddot{q}=[\frac{\partial p}{\partial q}\ \ \frac{\partial p}{\partial \dot{q}}]\cdot\begin{bmatrix}\dot{q}\\ \ddot{q}\end{bmatrix}$

```
dpdt=jacobian(p,[q; v_q ])*[v_q;  a_q]
```

dpdt =

$$\begin{pmatrix} M\,a_\rho \\ a_\phi\,(M\,\rho^2+J)+2\,M\,\omega\,\rho\,v_\rho \end{pmatrix}$$

```
syms tau real %par sobre barra (externo), trabajo virtual tau*ðphi
```

Ecs. Euler Lagrange del movimiento $\frac{dp}{dt}-\frac{\partial L}{\partial q}=Q$

```
EcsMovimiento = dpdt - jacobian(Lagrangian,q)' == [0;tau]
```

EcsMovimiento =

$$\begin{pmatrix} -M\,\rho\,\omega^2+M\,a_\rho+M\,g\,\sin(\phi)=0 \\ a_\phi\,(M\,\rho^2+J)+2\,M\,\omega\,\rho\,v_\rho+M\,g\,\rho\cos(\phi)=\tau \end{pmatrix}$$

```
a_q_EulerLagrange2GL=solve( EcsMovimiento  , a_q);
a_q_2gl=simplify(a_q_EulerLagrange2GL.a_rho)
```

a_q_2gl = $\omega^2\,\rho-g\,\sin(\phi)$

```
a_phi_2gl=simplify(a_q_EulerLagrange2GL.a_phi)
```

a_phi_2gl =

$$-\frac{2\,M\,\omega\,\rho\,v_\rho-\tau+M\,g\,\rho\cos(\phi)}{M\,\rho^2+J}$$

Con lo cual, el par τ necesario para mantener el giro a velocidad constante de la barra guía es:

```
tau_rFL=simplify(solve(a_phi_2gl==0,tau))
```

Warning: Solutions are only valid under certain conditions. To include parameters and conditions in the solution, specify the 'ReturnConditions' value as 'true'.

tau_rFL = $M\,\rho\,(2\,\omega\,v_\rho+g\cos(\phi))$