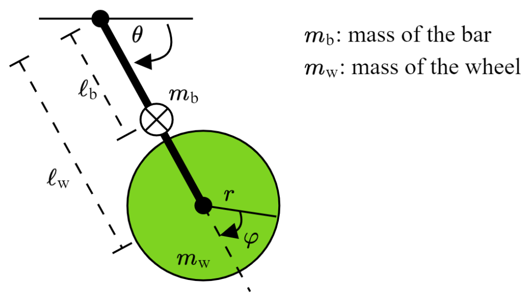
A Reaction Wheel Pendulum

Auralius Manurung
ME - Universitas Pertamina

11 September 2021

J_w: moment of inertia of the wheel at its center of mass
J_b: moment of inertia of the bar at its center of mass




```
> restart
> with(LinearAlgebra):
  with(Physics):
  with(plots):
  with(plottools):
  with(DEtools):
  with(Typesetting):
  interface(typesetting=extended):
  interface(showassumed=0):
> T__w:=1/2*J__w*(diff(varphi(t),t)+diff(theta(t),t))^2:
  V__w:=-m__w*g*l__w*cos(theta(t)):

  printf("\n");

  print(`Kinetic energy of the wheel:`);
  T:=simplify(T__w):
  print('T__m'=T__w);

  printf("\n");

  print(`Potential energy of the wheel:`);
  print('V__w'=V__w);

  printf("\n");
```

Kinetic energy of the wheel:

$$T_m = \frac{J_w \left(\dot{\varphi}(t) + \dot{\theta}(t)\right)^2}{2}$$

Potential energy of the wheel:

$$V_w = -m_w\, g\, l_w \cos(\theta(t))$$

```
> T__b:=1/2*(J__b+m__b*l__b^2+m__w*l__w^2)*diff(theta(t),t)^2:
  V__b:=-m__b*g*l__b*cos(theta(t)):

  printf("\n");

  print(`Kinetic energy of the bar:`);
  print('T__b'=T__b);

  printf("\n");

  print(`Potential energy of the bar:`);
  print('V__b'=V__b);
```

Kinetic energy of the bar:

$$T_b = \frac{\left(m_b l_b^{\,2} + m_w l_w^{\,2} + J_b\right) \dot{\theta}(t)^2}{2}$$

Potential energy of the bar:

$$V_b = -m_b g\, l_b \cos(\theta(t))$$

```
> T:=T__b+T__w:
  V:=V__b+V__w:
  L:=T-V:
  print('L'=L);

  eq1:=simplify(<diff(L,theta(t)), diff(L,varphi(t))>):
  eq2:=simplify(<diff(L,diff(theta(t),t)), diff(L,diff(varphi(t),t))>):
  eq3:=simplify(<diff(eq2[1],t), diff(eq2[2],t)>):

  print('diff(L,x)'=eq1);
  print('diff(L,diff(x,t))'=eq2);
  print('diff(diff(L,diff(x,t)),t)'=simplify(eq3));

  printf("\n");
  print(`Thus,`);
  print('diff(diff(L,diff(x,t)),t)-diff(L,x)'=<0, tau(t)>);
  eq4:=simplify({(eq3[1]-eq1[1]=0), (eq3[2]-eq1[2]=tau(t))}):
  print(<eq4[1],eq4[2]>);
  eq5:=simplify(solve(eq4,{diff(varphi(t),t,t), diff(theta(t),t,t)})):
  print(<eq5[1],eq5[2]>);
```

$$L = \frac{\left(m_b l_b^{\,2} + m_w l_w^{\,2} + J_b\right) \dot{\theta}(t)^2}{2} + \frac{J_w \left(\dot{\varphi}(t) + \dot{\theta}(t)\right)^2}{2} + m_b g\, l_b \cos(\theta(t)) + m_w g\, l_w \cos(\theta(t))$$

$$\frac{\partial}{\partial x} L = \begin{bmatrix} -\sin(\theta(t))\, g \left(l_b m_b + l_w m_w\right) \\ 0 \end{bmatrix}$$

$$\frac{\partial}{\partial \frac{\partial}{\partial t} x} L = \begin{bmatrix} \left(m_b l_b^2 + m_w l_w^2 + J_b + J_w\right) \dot{\theta}(t) + J_w \dot{\varphi}(t) \\ J_w \left(\dot{\varphi}(t) + \dot{\theta}(t)\right) \end{bmatrix}$$

$$\frac{\partial^2}{\partial \frac{\partial}{\partial t} x \partial t} L = \begin{bmatrix} \left(m_b l_b^2 + m_w l_w^2 + J_b + J_w\right) \ddot{\theta}(t) + J_w \ddot{\varphi}(t) \\ J_w \left(\ddot{\varphi}(t) + \ddot{\theta}(t)\right) \end{bmatrix}$$

Thus,

$$\frac{\partial^2}{\partial \frac{\partial}{\partial t} x \partial t} L - \frac{\partial}{\partial x} L = \begin{bmatrix} 0 \\ \tau(t) \end{bmatrix}$$

$$\begin{bmatrix} J_w \left(\ddot{\varphi}(t) + \ddot{\theta}(t)\right) = \tau(t) \\ \left(m_b l_b^2 + m_w l_w^2 + J_b + J_w\right) \ddot{\theta}(t) + J_w \ddot{\varphi}(t) + \sin\left(\theta(t)\right) g \left(l_b m_b + l_w m_w\right) = 0 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\theta}(t) = \dfrac{-\sin\left(\theta(t)\right) g \left(l_b m_b + l_w m_w\right) - \tau(t)}{m_b l_b^2 + m_w l_w^2 + J_b} \\ \ddot{\varphi}(t) = \dfrac{g J_w \left(l_b m_b + l_w m_w\right) \sin\left(\theta(t)\right) + \tau(t) \left(m_b l_b^2 + m_w l_w^2 + J_b + J_w\right)}{\left(m_b l_b^2 + m_w l_w^2 + J_b\right) J_w} \end{bmatrix}$$

```
> m__w:=0.15:
  m__b:=0.25:
  J__b:=1.98e-3:
  J__w:=7.81e-5:
  l__b:=0.12:
  l__w:=0.150:
  r:=0.02:
  g:=9.8:
  tau(t):=0.1*(Heaviside(t)-Heaviside(t-0.01)) - 0.2*(Heaviside(t-2)-Heaviside(t-2.01)) :
```

```
> ode:=eval(eq5):
```

```
> sol:=dsolve({ode[1], ode[2], theta(0)=0, D(theta)(0)=0, varphi(0)=0, D(varphi)(0)=0},
  numeric, method=rkf45, output=listprocedure):
  theta_array:=eval(theta(t),sol):
  varphi_array:=eval(varphi(t),sol):
```

```
> anim1 := animate(plot,[[0,l__w*sin(theta_array(t))], [0,-l__w*cos(theta_array(t))],
  color=red,thickness=5], t=0..5,scaling=constrained,frames=100):
  anim2 := animate(plot,[[l__w*sin(theta_array(t)),l__w*sin(theta_array(t))+r*sin
  (theta_array(t)+varphi_array(t))], [-l__w*cos(theta_array(t)),-l__w*cos(theta_array(t))
  -r*cos(theta_array(t)+varphi_array(t))], color=blue,thickness=5], t=0..5, scaling=
  constrained,frames=100):
```

```
anim3 := animate(circle,[[l__w*sin(theta_array(t)), -l__w*cos(theta_array(t))],r,
color=blue], t=0..5,scaling=constrained,frames=100):
> h:=display(anim1,anim2,anim3);
```