

Ejemplo01-6-2Shi-es

April 25, 2025

1 Ejemplo de perfil de leva con movimiento armónico simple

1.1 Seguidor alternante de cara plana

Una leva de placa con seguidor de movimiento alternativo y cara plana debe subir 2 pulg con movimiento armónico simple en 180° de rotación de la leva, y retornar con movimiento armónico simple en los 180° restantes. El radio del círculo primario será de 1.5 pulg y la leva girará en sentido contrario de las manecillas del reloj. Constrúyase el diagrama de desplazamientos y el perfil de la leva, dándole al vástago del seguidor una excentricidad de 0.75 pulg, en la dirección que reduce el esfuerzo de flexión en el seguidor durante la subida.

1.2 Librerías

Se ha de utilizar la librería `DiskCamMechanismLibrary`, la cual se puede encontrar en este [link](#), así como `matplotlib` y `numpy`.

```
[4]: from DiskCamMechanismLibrary import PDCamFlatFaceFollower
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation
import matplotlib.animation as animation

import matplotlib as mpl
mpl.rcParams['figure.dpi'] = 300
```

1.3 Movimiento armónico simple

Las ecuaciones de movimiento armónico simple para subida y descenso del seguidor se muestran a continuación:

$$\begin{aligned}y &= \frac{L}{2} (1 - \cos \theta) \\y' &= \frac{L}{2} \sin \theta \\y'' &= \frac{L}{2} \cos \theta\end{aligned}$$

donde L es el desplazamiento máximo que alcanza el seguidor y θ es la posición angular de la leva.

Se agrega el siguiente código de python para calcular, desplazamiento, velocidad y aceleración del seguidor.

```
[6]: def MovArmonicoSimple(th,L):  
    y = 0.5*L*(1-np.cos(th))  
    yp = 0.5*L*np.sin(th)  
    ypp = 0.5*L*np.cos(th)  
    return y,yp,ypp
```

1.4 Datos del problema:

$$\begin{aligned}L &= 2 \text{ pulg} \\ r_{\text{primario}} &= 1.5 \text{ pulg} \\ \epsilon &= 0.75 \text{ pulg}\end{aligned}$$

```
[8]: # %% Datos del problema  
L=2  
Rbase=1.5 #radio primario  
Rbroca=3/16 # Radio de la broca (centro de la leva)  
excentricidad = 0.75  
# posicion angular del seguidor en radianes  
posAngularSeguidor = np.pi/2  
# barrido angular de cero a 2pi radianes  
theta = np.linspace(0,1,500)*2*np.pi  
# calcular desplazamiento, velocidad, aceleracion  
y,yp,ypp = MovArmonicoSimple(theta,L)  
  
CamData={'theta':theta,  
        'y':y,  
        'yp':yp,  
        'ypp':ypp,  
        'Rbase':Rbase,  
        'Rhole':Rbroca,  
        'epsilon':excentricidad,  
        'FollowerAng':posAngularSeguidor,  
        'Followerwidth': 4/16,  
        'turn_direction':'anti-clockwise',  
        }
```

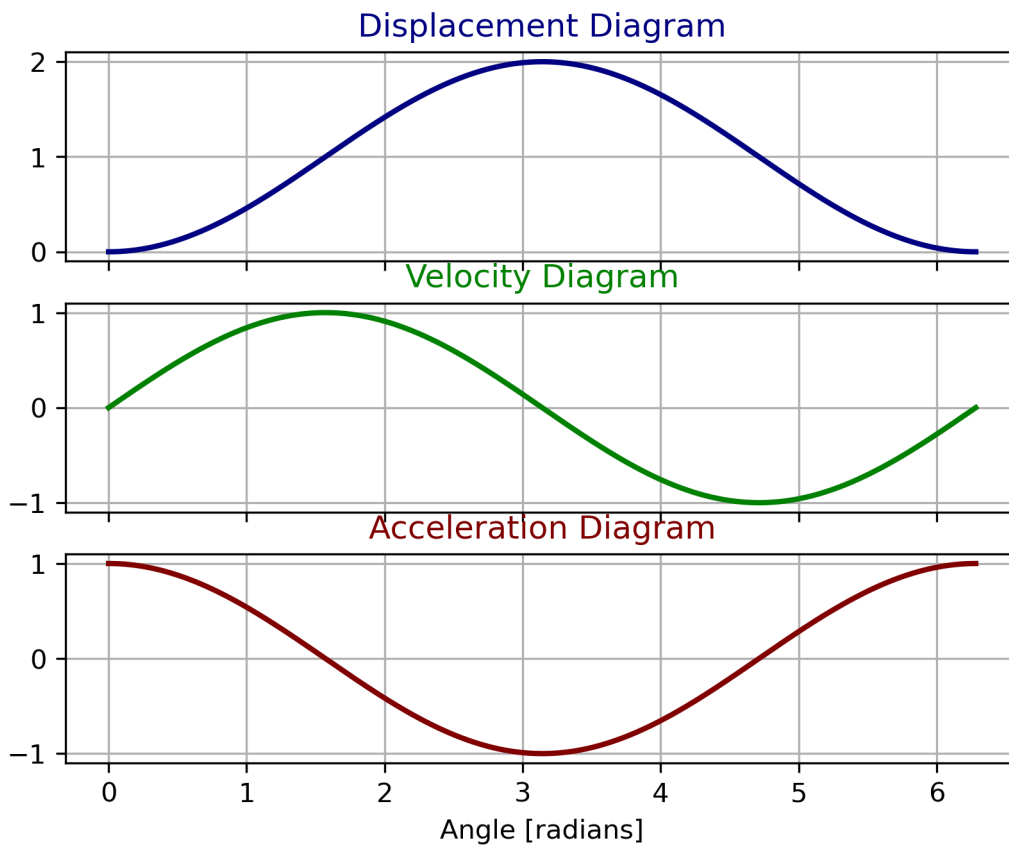
1.5 Calcular el perfil de la Leva

```
[10]: Leva=PDCamFlatFaceFollower(**CamData)
```

1.6 Diagrama de movimiento

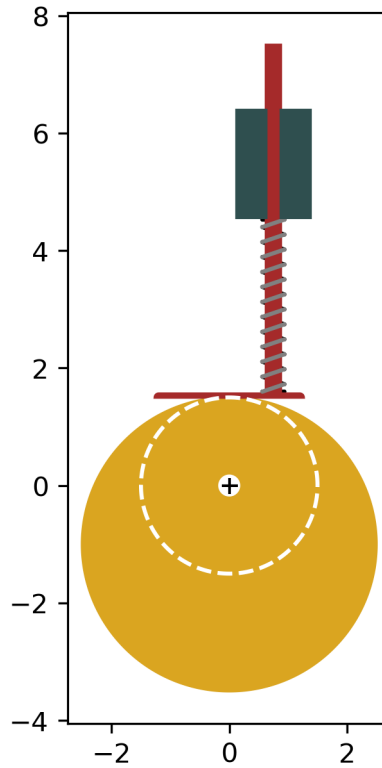
```
[12]: figMD=plt.figure()  
Leva.PlotMotionDiagram(figMD)
```

```
[12]: array([<Axes: title={'center': 'Displacement Diagram'}>,  
            <Axes: title={'center': 'Velocity Diagram'}>,  
            <Axes: title={'center': 'Acceleration Diagram'}, xlabel='Angle  
[radians]'>],  
          dtype=object)
```



1.7 Graficar el perfil de la leva

```
[14]: figPCam=plt.figure()  
Leva.PlotCamFlatFollower(figPCam)
```



Los datos de las coordenadas del perfil se encuentran en los atributos `Leva.Xp` y `Leva.Yp`:

```
[16]: print(Leva.Xp[0:10]) # Just a few data
      print(Leva.Yp[0:10]) # Just a few data
```

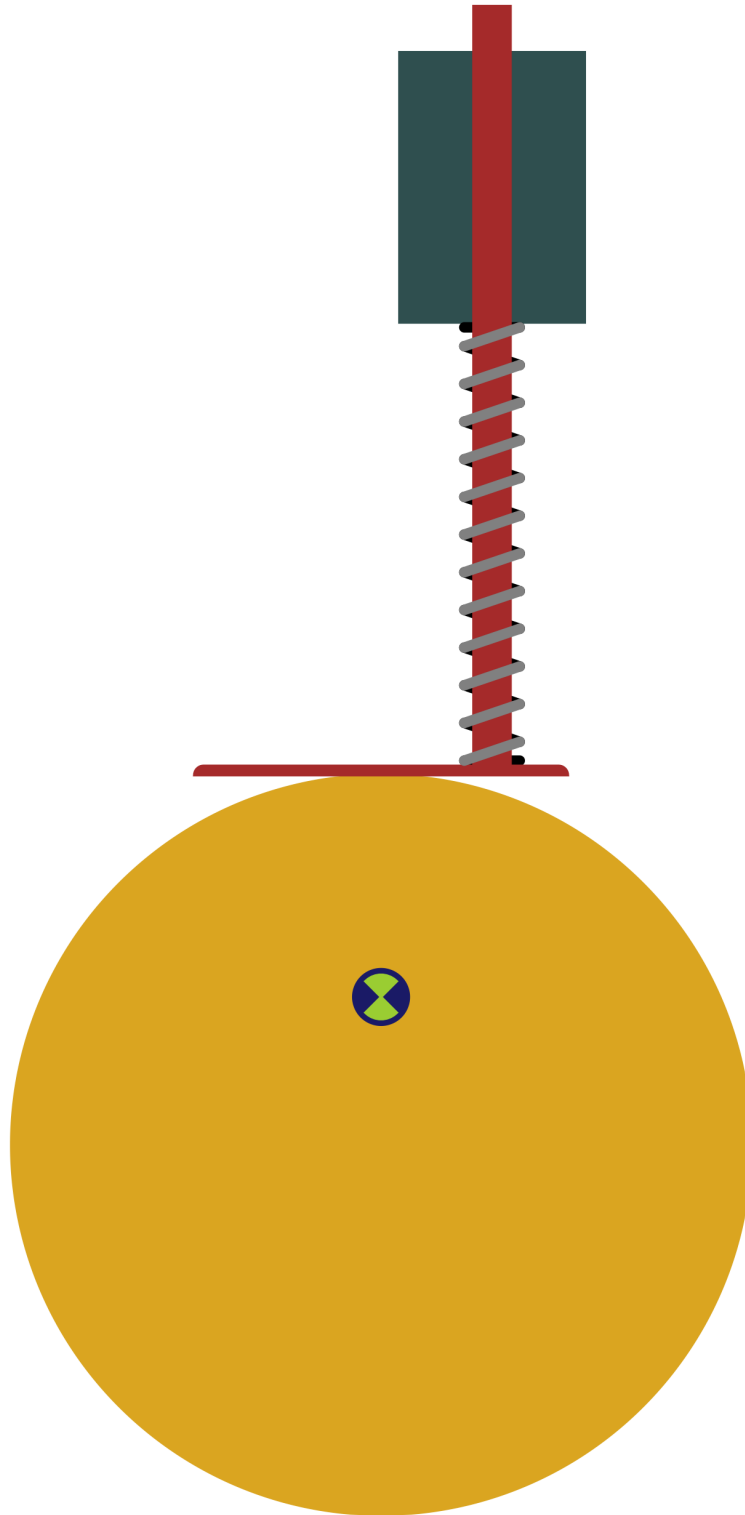
```
[9.18485099e-17 3.14780525e-02 6.29511143e-02 9.44141955e-02
 1.25862308e-01 1.57290465e-01 1.88693685e-01 2.20066988e-01
 2.51405401e-01 2.82703955e-01]
[1.5          1.49980182 1.49920731 1.49821656 1.49682973 1.49504704
 1.49286877 1.49029527 1.48732694 1.48396427]
```

Los datos de la curva de paso se encuentran en los atributos `Leva.Xr` y `Leva.Yr`:

1.8 Animación de la leva

```
[19]: fig, ax=plt.subplots()
      ax.set_axis_off()
      init_func=Leva.initAnim(ax),
      dpi=100
      width = 1920/dpi
      height = 1080/dpi
      fig.set_size_inches(width,height)
```

```
anim3 = FuncAnimation(fig, Leva, frames=np.arange(1000),  
                      interval=100, blit=False)  
plt.show()
```



1.9 Guardar la animación de la leva en un archivo

```
[ ]: writer = animation.writers['ffmpeg'](fps=30)
anim3.save('Leva01-6-2.mp4',writer=writer,dpi=dpi)
```

1.10 Código completo

```
[ ]: ### Librerias
from DiskCamMechanismLibrary import PDCamFlatFaceFollower
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation
import matplotlib.animation as animation
ver="c"
# %% Movimiento armonico simple
def MovArmonicoSimple(th,L):
    y = 0.5*L*(1-np.cos(th))
    yp = 0.5*L*np.sin(th)
    ypp = 0.5*L*np.cos(th)
    return y,yp,ypp

# %% Datos del problema
L=2
Rbase=1.5 #radio primario
Rbroca=3/16 # Radio de la broca (centro de la leva)
excentricidad = 0.75
# posicion angular del seguidor en radianes
posAngularSeguidor = np.pi/2
# barrido angular de cero a 2pi radianes
theta = np.linspace(0,1,500)*2*np.pi
# calcular desplazamiento, velocidad, aceleracion
y,yp,ypp = MovArmonicoSimple(theta,L)

# Agrupar datos en diccionario, para otros parametros consultar la
↪documentacion de DiskCamMechanismLibrary
CamData={'theta':theta,
        'y':y,
        'yp':yp,
        'ypp':ypp,
        'Rbase':Rbase,
        'Rhole':Rbroca,
        'epsilon':excentricidad,
        'FollowerAng':posAngularSeguidor,
        'Followerwidth': 4/16,
```

```

        'turn_direction':'anti-clockwise',
    }

    ### Calcular el perfil de la Leva
    Leva=PDCamFlatFaceFollower(**CamData)

    ### Diagrama de movimiento
    figMD=plt.figure()
    Leva.PlotMotionDiagram(figMD)

    ### Graficar el perfil de la leva
    figPCam=plt.figure()
    Leva.PlotCamFlatFollower(figPCam)

    ### Animación de la leva
    fig, ax=plt.subplots()
    ax.set_axis_off()
    init_func=Leva.initAnim(ax),
    dpi=100
    width = 1920/dpi
    hight = 1080/dpi
    fig.set_size_inches(width,hight)

    anim3 = FuncAnimation(fig, Leva, frames=np.arange(1000),
                          interval=100, blit=False)
    plt.show()

    ### Guardar la animación de la leva en un archivo
    writer = animation.writers['ffmpeg'](fps=30)
    anim3.save('Leva01-6-2.mp4',writer=writer,dpi=dpi)

```