1. Código

A continuación se presenta el código elaborado para ambos problemas. El desarrollo se realizó empleando *Jupyter* con *Sagemath kernel*. El código se encuentra disponible en `https://github.com/der-coder/SystemModeling/blob/master/Jupyter/Exam01.ipynb`. Se recomienda visualizarlo en `https://nbviewer.jupyter.org/`

1.1. Problema 1

```
m_car = 3 # Kilograms
m_pendulum = 1 # Kilograms
l_pendulum = 1.5 # Meters
k_ip = 0.5 # Newton/meter
g = 9.81 # meter/second^2
f_spring = 0
f_theta = 0
t = [0,10] # Evaluation time
x0_ip = np.array([1.0,1.0,n(pi*3/4),0]) # Initial conditions

def invertedPendulum(t, x):
    dx = np.zeros((len(x))) # Create the matrix to store the values

    pos = x[0]  # Define state variables
    d_pos = x[1]
    theta = x[2]
    d_theta = x[3]

    # Define matrix of the syste
    # Such that A d2_q + b = forces

    A = np.array([[m_pendulum + m_car,
    0.5*m_pendulum*mt.cos(theta)],
    [0.5*m_pendulum*l_pendulum*mt.cos(theta),
    m_pendulum*(l_pendulum^2)] ])

    b = np.array([[-0.5*m_pendulum*(d_theta^2)*mt.sin(theta)],
    [-0.5*m_pendulum*l_pendulum*d_theta*d_pos*mt.sin(theta)
```

```
    +
    0.5*m_pendulum*d_pos*l_pendulum*d_theta*mt.sin(theta)
    - m_pendulum*g*l_pendulum*mt.sin(theta)]])

    forces = np.array([[f_spring],[f_theta]])

    x_sol = np.matmul(np.linalg.inv(A),forces-b)

    dx[0]= d_pos
    dx[1] = x_sol[0]
    dx[2]= d_theta
    dx[3] = x_sol[1]

    return dx

f_spring= 0
f_theta = 0
resultsInvertedPendulum =
integrate.solve_ivp(invertedPendulum,[0,10],x0_ip,  max_step=0.05)

f_spring= 1
f_theta = 0
resultsInvertedPendulum2 =
integrate.solve_ivp(invertedPendulum,[0,10],x0_ip,  max_step=0.05)

xs = np.transpose(resultsInvertedPendulum.y)
ts = np.transpose(resultsInvertedPendulum.t)


plt.rc('text', usetex=True)
plt.rc('font', family='serif')

plt.figure(num=1,figsize=(15,10))
plt.plot(ts, xs[:,0],"--k",ts,xs[:,2],"k", linewidth=6)
plt.xlim(0, 12)
plt.legend([u'$x$',u'$\\theta$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
```

```
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $x$ y $\\theta$",fontsize=80)
plt.tight_layout()

plt.figure(num=2,figsize=(15,10))
plt.plot(ts, xs[:,1],"--k",ts,xs[:,3],"k", linewidth=5)
plt.xlim(0, 15)
plt.legend([u'$\dot{x}$',u'$\dot{\\theta}$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $\dot{x}$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()


plt.figure(num=3,figsize=(15,10))
plt.plot(xs[:,0],xs[:,1],"k", linewidth=5)
plt.xlabel(u'$x$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{x}$", fontsize=40)
plt.title(u"Diagrama fase de $x$ y $\dot{x}$",fontsize=80)
plt.tight_layout()

plt.figure(num=4,figsize=(15,10))
plt.plot(xs[:,2],xs[:,3],"k", linewidth=5)
plt.xlabel(u'$\\theta$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{\\theta}$", fontsize=40)
plt.title(u"Diagrama fase de $\\theta$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()

xs2 = np.transpose(resultsInvertedPendulum2.y)
ts2 = np.transpose(resultsInvertedPendulum2.t)

plt.figure(num=11,figsize=(15,10))
plt.plot(ts2, xs2[:,0],"--k",ts2,xs2[:,2],"k", linewidth=5)
```

```
plt.xlim(0, 12)
plt.legend([u'$x$',u'$\\theta$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $x$ y $\\theta$",fontsize=80)
plt.tight_layout()

plt.figure(num=12,figsize=(15,10))
plt.plot(ts2, xs2[:,1],"--k",ts2,xs2[:,3],"k", linewidth=5)
plt.xlim(0, 15)
plt.legend([u'$\dot{x}$',u'$\dot{\\theta}$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $\dot{x}$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()


plt.figure(num=13,figsize=(15,10))
plt.plot(xs2[:,0],xs2[:,1],"k", linewidth=5)
plt.xlabel(u'$x$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{x}$", fontsize=40)
plt.title(u"Diagrama fase de $x$ y $\dot{x}$",fontsize=80)
plt.tight_layout()

plt.figure(num=14,figsize=(15,10))
plt.plot(xs2[:,2],xs2[:,3],"k", linewidth=5)
plt.xlabel(u'$\\theta$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{\\theta}$", fontsize=40)
plt.title(u"Diagrama fase de $\\theta$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()
```

1.2. Problema 2

```
m = 1 # Kilograms
l = 1.5 # Meters
k = 0.5 # Newton/meter
# Gravity is already defined in Problem 1
f_1 = 0
f_2 = 0
# Evaluation time is already defined in Problem 1
x0_sp = np.array([1.0,1.0,1.0,1.0]) # Initial conditions

def simplePendulum(t, x):
    dx = np.zeros((len(x))) # Create the matrix to store the values

    pos = x[0]  # Define state variables
    d_pos = x[1]
    theta = x[2]
    d_theta = x[3]

    # Define matrix of the syste
    # Such that A d2_q + b = forces

    A = np.array([[m, m*l*mt.cos(theta)],
                  [l*mt.cos(theta),m*(l^2)] ])

    b = np.array([[-m*l*(d_theta^2)*mt.sin(theta)],
        [-l*(d_pos^2)*mt.sin(theta)
        +m*d_pos*d_theta*l*mt.sin(theta)]])

    forces = np.array([[f_1],[f_2]])

    x_sol = np.matmul(np.linalg.inv(A),forces-b)

    dx[0]= d_pos
    dx[1] = x_sol[0]
    dx[2]= d_theta
    dx[3] = x_sol[1]
```

```
    return dx

f_1= 0
f_2 = 0
resultssimplePendulum =
integrate.solve_ivp(simplePendulum,[0,10],x0_sp,  max_step=0.05)

f_1 = 1
f_2 = 0
resultssimplePendulum2 =
integrate.solve_ivp(simplePendulum,[0,10],x0_sp,  max_step=0.05)

xs3 = np.transpose(resultssimplePendulum.y)
ts3 = np.transpose(resultssimplePendulum.t)

plt.figure(num=21,figsize=(15,10))
plt.plot(ts3, xs3[:,0],"--k",ts3,xs3[:,2],"k", linewidth=5)
plt.xlim(0, 12)
plt.legend([u'$x$',u'$\\theta$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $x$ y $\\theta$",fontsize=80)
plt.tight_layout()

plt.figure(num=22,figsize=(15,10))
plt.plot(ts3, xs3[:,1],"--k",ts3,xs3[:,3],"k", linewidth=5)
plt.xlim(0, 15)
plt.legend([u'$\dot{x}$',u'$\dot{\\theta}$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $\dot{x}$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()
```

```
plt.figure(num=23,figsize=(15,10))
plt.plot(xs3[:,0],xs3[:,1],"k", linewidth=5)
plt.xlabel(u'$x$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{x}$", fontsize=40)
plt.title(u"Diagrama fase de $x$ y $\dot{x}$",fontsize=80)
plt.tight_layout()

plt.figure(num=24,figsize=(15,10))
plt.plot(xs3[:,2],xs3[:,3],"k", linewidth=5)
plt.xlabel(u'$\\theta$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{\\theta}$", fontsize=40)
plt.title(u"Diagrama fase de $\\theta$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()

xs4 = np.transpose(resultssimplePendulum2.y)
ts4 = np.transpose(resultssimplePendulum2.t)

plt.figure(num=31,figsize=(15,10))
plt.plot(ts4, xs4[:,0],"--k",ts4,xs4[:,2],"k", linewidth=5)
plt.xlim(0, 12)
plt.legend([u'$x$',u'$\\theta$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
plt.title(u"Comparaci\\'on de $x$ y $\\theta$",fontsize=80)
plt.tight_layout()

plt.figure(num=32,figsize=(15,10))
plt.plot(ts4, xs4[:,1],"--k",ts4,xs4[:,3],"k", linewidth=5)
plt.xlim(0, 15)
plt.legend([u'$\dot{x}$',u'$\dot{\\theta}$'], loc=5,fontsize=60,
frameon=False)
plt.xlabel(u'Tiempo', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"Soluci\\'on", fontsize=40)
```

```
plt.title(u"Comparaci\\'on de $\dot{x}$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()


plt.figure(num=33,figsize=(15,10))
plt.plot(xs4[:,0],xs4[:,1],"k", linewidth=5)
plt.xlabel(u'$x$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{x}$", fontsize=40)
plt.title(u"Diagrama fase de $x$ y $\dot{x}$",fontsize=80)
plt.tight_layout()

plt.figure(num=34,figsize=(15,10))
plt.plot(xs4[:,2],xs4[:,3],"k", linewidth=5)
plt.xlabel(u'$\\theta$', fontsize=40)
plt.tick_params(labelsize='30')
plt.ylabel(u"$\dot{\\theta}$", fontsize=40)
plt.title(u"Diagrama fase de $\\theta$ y $\dot{\\theta}$",fontsize=80)
plt.tight_layout()
```