

```

import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

# -----
# Parameters
# -----
M = 1.0 # Mass of cart [kg]
m = 0.1 # Mass of pendulum [kg]
L = 0.5 # Length of pendulum [m]
g = 9.81 # Gravity [m/s^2]

# -----
# Input force u(t)
# -----
def u(t):
    return 0.0
    # Example: return 1.0 # constant force of 1 N

# -----
# State-space model
# x1 = y
# x2 = y_dot
# x3 = theta
# x4 = theta_dot
# -----

```

```

def inverted_pendulum(t, x):
    x1, x2, x3, x4 = x

    dx1 = x2
    dx2 = -(m * g / M) * x3 + (1 / M) * u(t)
    dx3 = x4
    dx4 = (g / L) * x3 - (1 / (M * L)) * u(t)

    return [dx1, dx2, dx3, dx4]

# -----
# Initial conditions
# -----
y0 = 0.0    # initial cart position [m]
y_dot0 = 0.0 # initial cart velocity [m/s]
theta0 = 0.1 # initial pendulum angle [rad]
theta_dot0 = 0.0 # initial angular velocity [rad/s]

x0 = [y0, y_dot0, theta0, theta_dot0]

# -----
# Simulation time
# -----
t_start = 0
t_end = 1.0
t_eval = np.linspace(t_start, t_end, 500)

```

```
# -----  
# Solve differential equations  
# -----  
solution = solve_ivp(  
    inverted_pendulum,  
    [t_start, t_end],  
    x0,  
    t_eval=t_eval  
)  
  
# -----  
# Results  
# -----  
t = solution.t  
y = solution.y[0]  
y_dot = solution.y[1]  
theta = solution.y[2]  
theta_dot = solution.y[3]  
  
# -----  
# Plots  
# -----  
plt.figure(figsize=(10, 6))  
  
plt.plot(t, y, label="Cart position y(t)")
```

```
plt.plot(t, theta, label="Pendulum angle theta(t)")
```

```
plt.xlabel("Time [s]")
```

```
plt.ylabel("Response")
```

```
plt.title("Inverted Pendulum Simulation")
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```