Control Theory homework 4

Kamil Hayrullin
BS18-02

1 Introduction

My Innopolis mail is k.hayrullin@innopolis.ru
My variant is ”c”
Link to my GitHub repository is here

2 Consider classical benchmark system in control theory - inverted pendulum on a cart

$$(M+m)\ddot{x} - mlcos(\theta)\ddot{\theta} + mlsin(\theta)\dot{\theta}^2 = F$$

$$-cos(\theta)\ddot{x} + l\ddot{\theta} - gsin(\theta) = 0$$

$$g = 9.81, M = 15.1, m = 1.2, l = 0.35$$

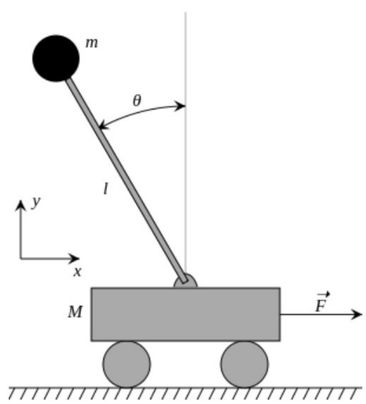


Figure 1: A schematic drawing of the inverted pendulum on a cart. The rod is considered massless. The mass of the cart and the point mass at the end of the rod are denoted by M and m. The rod has a length l.

(A) Write equations of motion of the system in manipulator form

$$M(q)\ddot{q} + n(q, \dot{q}) = Bu$$

where

$$u = F, q = \begin{bmatrix} x & \theta \end{bmatrix}^T$$

In general form:

$$a\ddot{x} + b\ddot{y} + c\dot{y} = F$$

$$d\ddot{x} + e\ddot{y} + gy = 0$$

$$\begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} c\dot{y} \\ gy \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

In our case:

$$(M + m)\ddot{x} - mlcos(\theta)\ddot{\theta} + mlsin(\theta)\dot{\theta}^2 = F$$

$$-cos(\theta)\ddot{x} + l\ddot{\theta} - gsin(\theta) = 0$$

$$\begin{bmatrix} M + m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} mlsin(\theta)\dot{\theta}^2 \\ -gsin(\theta) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

$$\begin{bmatrix} 16.3 & -0.42cos(\theta) \\ -cos(\theta) & 0.35 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0.42sin(\theta)\dot{\theta}^2 \\ -9.81sin(\theta) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

(B) Write dynamics of the system in control affine nonlinear form

$$\dot{z} = f(z) + g(z)u$$

where

$$z = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

In general form:

$$\begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} c\dot{y} \\ gy \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} F - \begin{bmatrix} c\dot{y} \\ gy \end{bmatrix} \right)$$

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ .. \\ .. \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ .. \\ .. \end{bmatrix} F$$

In our case:

$$\begin{bmatrix} M+m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} mlsin(\theta)\dot{\theta}^2 \\ -gsin(\theta) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} F$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} M+m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} F - \begin{bmatrix} mlsin(\theta)\dot{\theta}^2 \\ -gsin(\theta) \end{bmatrix} \right)$$

$$\begin{bmatrix} M+m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{m+M-mcos^2(\theta)} & -\frac{2mcos(\theta)}{cos(2\theta)m-m-2M} \\ \frac{cos(\theta)}{l(-mcos^2(\theta)+m+M)} & \frac{m+M}{l(-mcos^2(\theta)+m+M)} \end{bmatrix}$$

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \frac{1}{m+M-mcos^2(\theta)} * (-1)mlsin(\theta)\dot{\theta}^2 - \frac{2mcos(\theta)}{cos(2\theta)m-m-2M} * gsin(\theta) \\ -\frac{cos(\theta)}{l(-mcos^2(\theta)+m+M)} * mlsin(\theta)\dot{\theta}^2 + \frac{m+M}{l(-mcos^2(\theta)+m+M)} * gsin(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m+M-mcos^2(\theta)} \\ \frac{cos(\theta)}{l(-mcos^2(\theta)+m+M)} \end{bmatrix} F$$

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ -\frac{9.88848\dot{\theta}^2 sin^2(x)cos(x)}{(1.2cos(2x)-16.3)(16.3-1.2cos^2(x))} \\ \frac{sin(x)(\dot{\theta}^2 cos(x)-380.721)}{cos^2(x)-13.5833} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{16.3-1.2cos^2(\theta)} \\ \frac{cos(\theta)}{0.35(-1.2cos^2(\theta)+16.3)} \end{bmatrix} F$$

(C) Linearize nonlinear dynamics of the systems around equilibrium point

$$\delta\dot{z} = A\delta z + B\delta u$$
$$\tilde{z} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\Lambda(\theta) = \begin{bmatrix} M+m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix}$$

$$\Lambda(0) = \begin{bmatrix} M+m & -ml \\ -1 & l \end{bmatrix}$$

$$\Lambda^{-1}(0) = \frac{1}{Ml}\begin{bmatrix} l & ml \\ 1 & m+M \end{bmatrix} = \begin{bmatrix} \frac{1}{M} & \frac{m}{M} \\ \frac{1}{Ml} & \frac{m+M}{Ml} \end{bmatrix}$$

$$\frac{\delta f}{\delta x} = \begin{bmatrix} 0 & 0 \quad 1 \quad 0 \\ 0 & 0 \quad 0 \quad 1 \\ \Lambda^{-1}(0) * \begin{bmatrix} 0 & -mlcos(0)*0 & 0 & -2mlsin(0) \\ 0 & gcos(0) & 0 & 0 \end{bmatrix} & \end{bmatrix}$$

$$A = \frac{\delta f}{\delta x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{gm}{M} & 0 & 0 \\ 0 & \frac{g(M+m)}{Ml} & 0 & 0 \end{bmatrix} B = \frac{\delta f}{\delta u} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{M} \\ \frac{1}{Ml} \end{bmatrix}$$

$$A = \frac{\delta f}{\delta x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.7796 & 0 & 0 \\ 0 & 30.256 & 0 & 0 \end{bmatrix} B = \frac{\delta f}{\delta u} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{0.06622} \\ \frac{1}{0.1892} \end{bmatrix}$$

(D) Check stability of the linearized system using any method you like Lets find eigenvalues of A
A =

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.7796 & 0 & 0 \\ 0 & 30.256 & 0 & 0 \end{bmatrix}$$

$$\lambda_{1,2} = -+\frac{\sqrt{\frac{3782}{5}}}{5}$$

$$\lambda_3 = 0$$

not all eigenvalues less than 0, system is unstable

(E) Check if linearized system is controllable We check it using such code in matlab:

```
A = [0, 0, 1, 0; 0, 0, 0, 1; 0, 0.7796, 0, 0; 0, 30.256, 0, 0]
B = [0; 0; 1/0.06622; 1/0.1892]
sys = ctrb(A, B)
unco = length(A) - rank(sys)
```

The uncontrollable state (unco) indicates that sys does not have full rank. In our case system is controllable, because unco = 0.

```
A =

         0         0    1.0000         0
         0         0         0    1.0000
         0    0.7796         0         0
         0   30.2560         0         0


B =

         0
         0
   15.1012
    5.2854


sys =

         0   15.1012         0    4.1205
         0    5.2854         0  159.9154
   15.1012         0    4.1205         0
    5.2854         0  159.9154         0


unco =

     0
```

(F) (for the controllable system) design state feedback controller for linearized system using pole placement method. Assess the performance of the controller for variety of initial conditions. Justify the choice of initial conditions. Solve the task by two ways: using root-locus and with python.

Using python we have:

```python
from scipy import signal
import numpy as np
A = np.array([[0, 0, 1, 0],
              [0, 0, 0, 1],
              [0, 0.7796, 0, 0],
              [0, 30.256, 0, 0]])
B = np.array([[0],[0],
              [1/0.06622],
              [1/0.1892]])
p = np.array([-1, -2, -3, -4])
K = signal.place_poles(A, B, p)
K.gain_matrix
```

```
array([[-0.05300579, 12.49788031, -0.11042873,  2.20751065]])
```

using matlab we have:

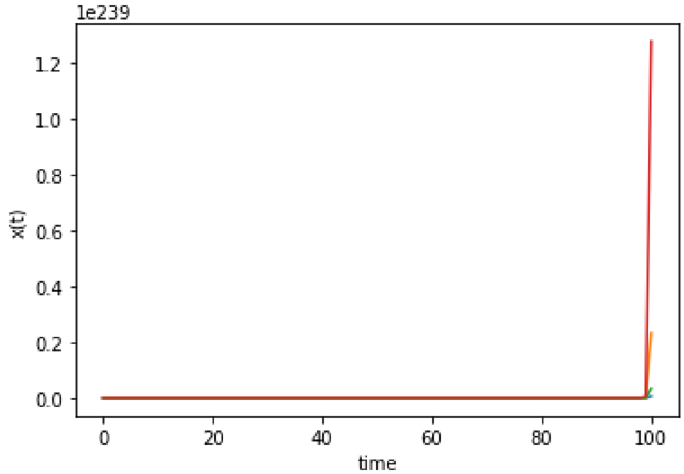
```matlab
A = [0, 0, 1, 0; 0, 0, 0, 1; 0, 0.7796, 0, 0; 0, 30.256, 0, 0]
B = [0; 0; 1/0.06622; 1/0.1892]
p = [-1, -2, -3, -4]
K = place(A,B,p)
```

```
Command Window

K =

   -0.0530   12.4979   -0.1104    2.2075
```

(G) Design linear quadratic regulator for linearized system. Assess the performance of the controller for variety of initial conditions
Link to Google Colab is here

```python
from math import sin
import numpy as np
import scipy.linalg
from scipy.integrate import odeint
import matplotlib.pyplot as plt

time = np.linspace(0, 100, 100)
Q = np.eye(4)*0.1
R = 0.01
B = np.array([[0], [0], [1/0.06622], [1/0.1892]])
A = np.array([[0, 0, 1, 0], [0, 0, 0, 1], [0, 0.7796, 0, 0], [0, 30.256, 0, 0]])


def lqr(A,B,Q,R):
    X = np.matrix(scipy.linalg.solve_discrete_are(A, B, Q, R))
    K = np.matrix(scipy.linalg.inv(B.T*X*B+R)*(B.T*X*A))
    return K

def LTV_LQR(x, t):
    global A, B, Q, R
    K = lqr(A, B, Q, R)
    T = (A - B*K).dot(x)
    return T[0, 0], T[0, 1], T[0, 2], T[0, 3]

def LTV(x, t):
    return A.dot(x)

print("LTV without control")
x0 = np.random.rand(4)
solution_LTV = odeint(LTV, x0, time)

plt.plot(time, solution_LTV)
plt.xlabel('time')
plt.ylabel('x(t)')
plt.show()

print("LTV with LQR control")
solution_LQR = odeint(LTV_LQR, x0, time)
plt.plot(time, solution_LQR)
plt.xlabel('time')
plt.ylabel('x(t)')
plt.show()
```

LTV without control



LTV with LQR control