1 Task 1

My name is *Anton Brisilin*, and my email is *a.brisilin@innopolis.university*, so my generated variant is **F**.

2 Task 2

2.1 Write equations of motion of the system in manipulator form

Source system of equations:

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

Manipulator form:

$$\begin{cases} M(q)\ddot{q} + n(q,\dot{q}) = Bu \\ u = F \\ q = \begin{bmatrix} x \\ \theta \end{bmatrix} \end{cases} \tag{2}$$

My goal is to transform (1) to the form of (2). To do it I can transform first equation of (1) to the form of

$$\begin{bmatrix} M+m & -mlcos(\theta) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + (mlsin(\theta)\dot{\theta}^2) = [1]u \tag{3}$$

This already is motion equation in standart (manipulator) form. But for this equation to describe motion of our system it should also consider second equation of system (1). Therefore, our equation (3) becomes

$$\begin{bmatrix} M+m & -mlcos(\theta) \\ -cos(\theta) & l \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} mlsin(\theta)\dot{\theta}^2 \\ -gsin\theta \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{4}$$

And with substitution of numbers from variant **F**:

$$\begin{bmatrix} 11.6 & -7.298 * cos(\theta) \\ -cos(\theta) & 0.89 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 7.298 * sin(\theta)\dot{\theta}^2 \\ -9.81 * sin\theta \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{5}$$

2.2 Write dynamics of the system in control affine non-linear form

Control affine nonlinear form:

$$\begin{cases} \dot{z} = f(z) + g(z)u \\ u = F \\ z = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T \end{cases} \tag{6}$$

It's pretty obvious that one can easily express $\dot{x}$ and $\dot{\theta}$ through z:

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F \tag{7}$$

So the goal is to express $\ddot{x}$ and $\ddot{\theta}$ through $x, \dot{x}, \theta$ and $\dot{\theta}$. From the second equation of system (1):

$$\ddot{x} = \frac{l\ddot{\theta} - gsin(\theta)}{cos(\theta)} \tag{8}$$

Substituting (8) to the first equation of (1), we can express $\ddot{\theta}$:

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

$$\ddot{\theta} = \frac{(M+m)(l\ddot{\theta} - gsin(\theta))}{mlcos^2(\theta)} + tg(\theta)\dot{\theta}^2 - \frac{F}{mlcos(\theta)}$$

$$\ddot{\theta} = \frac{(M+m)l\ddot{\theta}}{mlcos^2(\theta)} - \frac{(M+m)gsin(\theta)}{mlcos^2(\theta)} + tg(\theta)\dot{\theta}^2 - \frac{F}{mlcos(\theta)}$$

$$\ddot{\theta}\left(\frac{mlcos^2(\theta) - (M+m)l}{mlcos^2(\theta)}\right) = -\frac{(M+m)gsin(\theta)}{mlcos^2(\theta)} + tg(\theta)\dot{\theta}^2 - \frac{F}{mlcos(\theta)}$$

$$\ddot{\theta} = \frac{-\frac{(M+m)gsin(\theta)}{mlcos^2(\theta)} + tg(\theta)\dot{\theta}^2 - \frac{F}{mlcos(\theta)}}{\frac{mlcos^2(\theta) - (M+m)l}{mlcos^2(\theta)}}$$

$$\ddot{\theta} = \frac{-(M+m)gsin(\theta) + mlcos(\theta)sin(\theta)\dot{\theta}^2 - Fcos(\theta)}{mlcos^2(\theta) - (M+m)l}$$

$$\ddot{\theta} = \frac{(M+m)gsin(\theta) - mlcos(\theta)sin(\theta)\dot{\theta}^2}{l((M+m) - mcos^2(\theta))} + \frac{Fcos(\theta)}{l((M+m) - mcos^2(\theta))} \tag{9}$$

Now, substituting (9) to (8):

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

$$\ddot{x} = \frac{-mlsin(\theta)\dot{\theta}^2 + mgsin(\theta)cos(\theta)}{(M+m) - mcos^2(\theta)} + \frac{F}{(M+m) - mcos^2(\theta)} \tag{10}$$

Hence, our system (1) in control affine nonlinear form can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \frac{-mlsin(\theta)\dot{\theta}^2+mgsin(\theta)cos(\theta)}{(M+m)-mcos^2(\theta)} \\ \frac{(M+m)gsin(\theta)-mlcos(\theta)sin(\theta)\dot{\theta}^2}{l((M+m)-mcos^2(\theta))} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{(M+m)-mcos^2(\theta)} \\ \frac{cos(\theta)}{(M+m)-mcos^2(\theta)} \end{bmatrix} u \tag{11}$$

And with substitution of numbers from variant **F**:

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \frac{-7.298sin(\theta)\dot{\theta}^2+80.442sin(\theta)cos(\theta)}{11.6-8.2cos^2(\theta)} \\ \frac{113.796sin(\theta)-7.298cos(\theta)sin(\theta)\dot{\theta}^2}{10.324-7.298cos^2(\theta)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{11.6-8.2cos^2(\theta)} \\ \frac{cos(\theta)}{11.6-8.2cos^2(\theta)} \end{bmatrix} u \tag{12}$$

2.3 Linearize nonlinear dynamics of the systems around equilibrium point $\vec{z} = [0\ 0\ 0\ 0]^T$

General form of linearized equation:

$$\delta\dot{z} = A\delta z + B\delta u \tag{13}$$

Let's begin with checking, whether $\vec{z} = [0\ 0\ 0\ 0]^T$ is really an equilibrium point:

$$\begin{cases} \dot{x} = 0 \\ \dot{\theta} = 0 \\ \ddot{x} = 0 + \frac{1}{11.6-8.2cos^2(\theta)}u_e = 0 \\ \ddot{\theta} = 0 + \frac{cos(\theta)}{11.6-8.2cos^2(\theta)}u_e = 0 \end{cases} \tag{14}$$

It's seen that last two equations are true, iff $u_e = 0$. If our system is in form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} f_1(z,u) \\ f_2(z,u) \\ f_3(z,u) \\ f_4(z,u) \end{bmatrix} \tag{15}$$

Then, to linearize it, we should found such matrices A and B, that

$$\begin{cases} A = \begin{bmatrix} \frac{\partial f_1}{\partial z_1}|_{z_0,u_0} & \frac{\partial f_1}{\partial z_2}|_{z_0,u_0} & \frac{\partial f_1}{\partial z_3}|_{z_0,u_0} & \frac{\partial f_1}{\partial z_4}|_{z_0,u_0} \\ \frac{\partial f_2}{\partial z_1}|_{z_0,u_0} & \frac{\partial f_2}{\partial z_2}|_{z_0,u_0} & \frac{\partial f_2}{\partial z_3}|_{z_0,u_0} & \frac{\partial f_2}{\partial z_4}|_{z_0,u_0} \\ \frac{\partial f_3}{\partial z_1}|_{z_0,u_0} & \frac{\partial f_3}{\partial z_2}|_{z_0,u_0} & \frac{\partial f_3}{\partial z_3}|_{z_0,u_0} & \frac{\partial f_3}{\partial z_4}|_{z_0,u_0} \\ \frac{\partial f_4}{\partial z_1}|_{z_0,u_0} & \frac{\partial f_4}{\partial z_2}|_{z_0,u_0} & \frac{\partial f_4}{\partial z_3}|_{z_0,u_0} & \frac{\partial f_4}{\partial z_4}|_{z_0,u_0} \end{bmatrix} \\ B = \begin{bmatrix} \frac{\partial f_1}{\partial u}|_{z_0,u_0} \\ \frac{\partial f_2}{\partial u}|_{z_0,u_0} \\ \frac{\partial f_3}{\partial u}|_{z_0,u_0} \\ \frac{\partial f_4}{\partial u}|_{z_0,u_0} \end{bmatrix} \end{cases} \tag{16}$$

In our case, functions $f_1, ...f_4$ can be taken from previous part of the task. Now let's compute the matrices, considering

$$\begin{cases} z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} \\ f_1 = z_3 \\ f_2 = z_4 \\ f_3 = \frac{-7.298sin(z_2)z_4^2+80.442sin(z_2)cos(z_2)}{11.6-8.2cos^2(z_2)} + \frac{u}{11.6-8.2cos^2(z_2)} \\ f_4 = \frac{113.796sin(z_2)-7.298cos(z_2)sin(z_2)z_4^2}{10.324-7.298cos^2(z_2)} + \frac{cos(z_2)u}{11.6-8.2cos^2(z_2)} \end{cases} \tag{17}$$

And let's select one output among all state variables - I have chosen position (x). Here is obtained state-space model:

$$\begin{cases} A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 23.66 & 0 & 0 \\ 0 & 37.6 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ 0.294 \\ 0.294 \end{bmatrix} \\ \\ C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} D = \begin{bmatrix} 0 \end{bmatrix} \end{cases} \tag{18}$$

All the derivatives were computed with Matlab. Scripts for differentiation can be found in `Task2/diff.m`

2.4 Stability checking

I will check stability with looking to eigenvalues of A, obtained in (18). If all of them have negative real parts, then system is stable. Using `eig()` from `numpy` I obtained:

$$eigenvalues = \begin{bmatrix} 0 \\ 0 \\ 6.13 \\ -6.13 \end{bmatrix}$$

It can be seen, that there is a positive-valued eigenvalue of A, so the linearized system is not stable.

2.5 Controllability checking

For the linear system to be controllable, the next condition should hold:
If $\dot{x} = Ax + B$, then rank of $\Gamma = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix}$ should be equal to the dimension of A (in my case 4). Calculation of matrix Γ can be found in `Task2/contr.m`.

$$\Gamma = \begin{bmatrix} 0 & 0.2940 & 0 & 6.9560 \\ 0 & 0.2940 & 0 & 11.0544 \\ 0.2940 & 0 & 6.9560 & 0 \\ 0.2940 & 0 & 11.0544 & 0 \end{bmatrix} \tag{19}$$

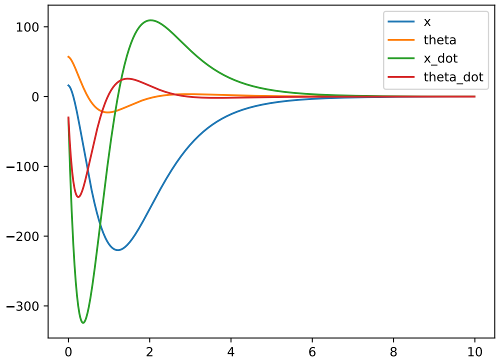
It is visible, that rank of this matrix is 4, which is also confirmed by Matlab. Hence, the system is controllable.

2.6 Design a state feedback controller

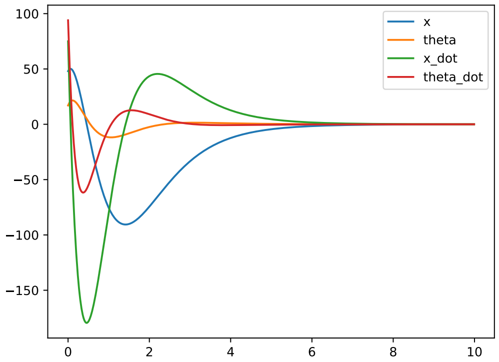
According to Wikipedia, state feedback controller is the controller designed using pole-placement method.
The task now is to determine where desired poles should be, and then design such controller that will force the actual poles be at the location of desired ones.
For now, poles of unstabilized system is equal to the eigenvalues of system, that were found in 2.4.
Let's set all poles be negative to make the system stable. I have chosen poles $P = [-1, -2 \pm i, -3]$.
As the Python solution I found `scipy.signal.place_poles()` function, that finds gain matrix K, that can be used in closed-loop system model:

$$\begin{cases} \dot{x} = Ax + Bu \\ u = -Kx \end{cases} \tag{20}$$

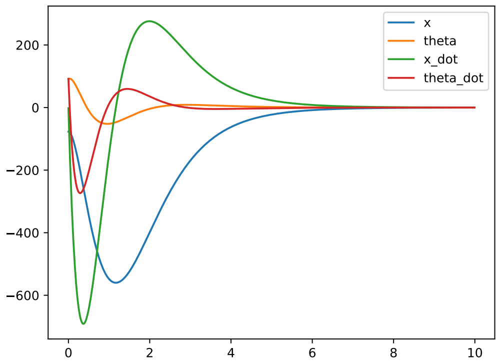
Here, are the step responses of system with 4 different initial condition, chosen in random.



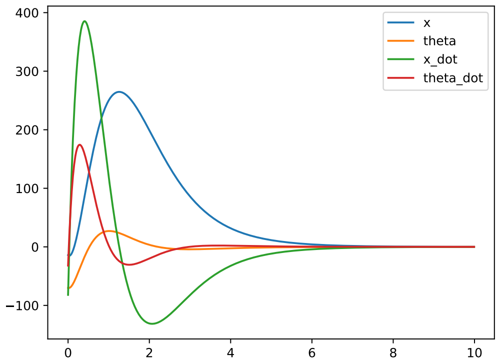
System response with initial vector [16 57 -32 -30]



System response with initial vector [48 17 75 94]

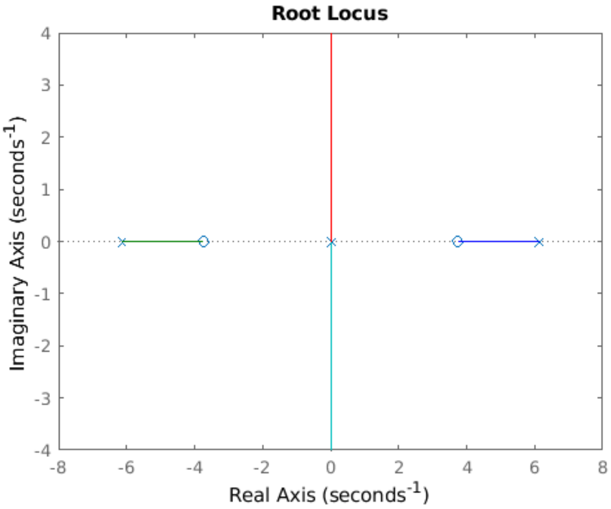


System response with initial vector [-77 91 -2 92]

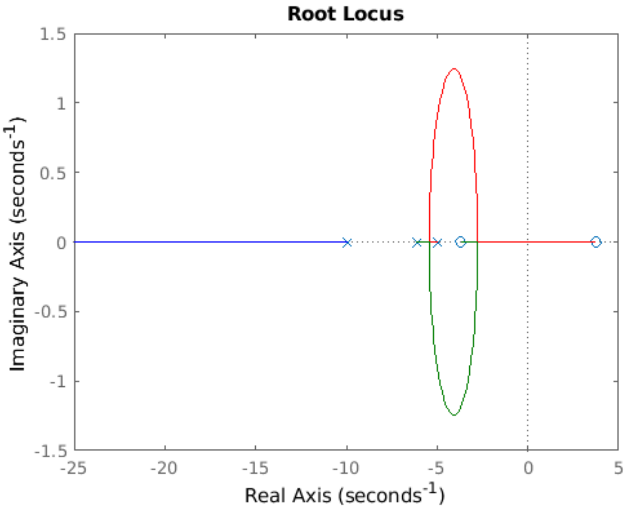


System response with initial vector [-14 -70 -82 -32]

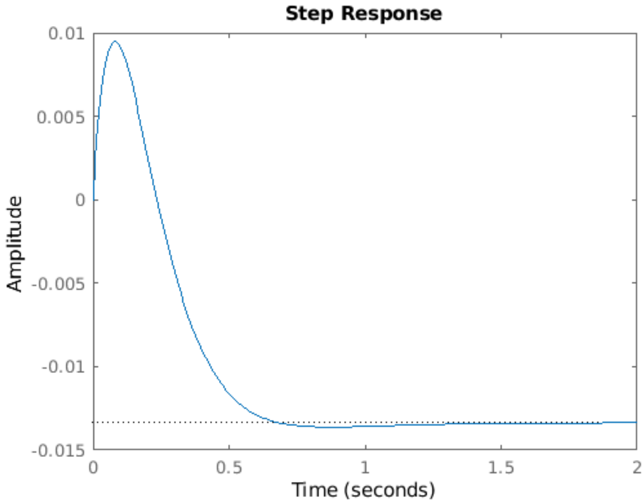
As it can be seen, all of them are going to stable state, with settling time about 5 seconds, but has large overshoot.
The Python code of solution is located at `Task2/pole_placement.py`.
In Matlab I used `rlocus()` function to view Root-Locus plot of the system. Unfortunately, `controlSystemDesigner` does not work properly on my version of Matlab, and shows wrong step response. So I looked at the plot, and added and removed poles, controlling my step response. Initially, my Root-Locus plot was looking like this:



As it can be seen, there is positive pole, so i added zero at this point to eliminate its influence. Then, I added two zeroes at the origin, and two poles at left half-plane, to make number of poles greater than number of zeroes, as written there. Now, my obtained Root-Locus plot is:



And step response of the system, is here:



As it can be seen, output of the system is pretty stable, with steady-state error around $2e-2$. Full code of the solution can be found in `Task2/design.m`.

3 Used software.

- Python 3.8.1
- Matlab R2018b 9.5.0
- draw.io

All software was run under Manjaro Linux with 5.4.18-rt kernel