Sabina Sasu – 260803977

Assignment 4, Cartpole LQR

To balance the cartpole, I decided to use LQR since the code was already mostly written with some values missing, meaning the A and B matrices, some tuning on Q and R and finally the control calculation using the resulting K matrix.

1. *Getting A and B*: This was the most tedious part as I was convinced my math was wrong as explained in the next step. The partial derivation steps are omitted as they wouldn't fit in the one-page report limit. I verified the derivations using online calculators like Symbolab.

$$x = \begin{bmatrix} x \\ \dot{x} \\ \dot{\theta} \\ \theta \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dfrac{2ml\dot{\theta}^2 \sin\theta + 3mg\sin\theta\cos\theta + 4(u - b\dot{x})}{4(M+m) - 3m\cos^2\theta} \\ \dfrac{-3(ml\dot{\theta}^2 \sin\theta\cos\theta + 2((M+m)g\sin\theta + (u - b\dot{x})\cos\theta))}{l(4(M+m) - 3m\cos^2\theta)} \\ \dot{\theta} \end{bmatrix}$$

$$A = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{\theta}} & \frac{\partial \dot{x}}{\partial \theta} \\ \frac{\partial \ddot{x}}{\partial x} & & \cdots & \\ \frac{\partial \ddot{\theta}}{\partial x} & & & \\ \frac{\partial \dot{\theta}}{\partial x} & & & \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1.6 & 0 & 5.9 \\ 0 & -4.8 & 0 & 47.2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \frac{\partial \dot{x}}{\partial u} \\ \frac{\partial \ddot{x}}{\partial u} \\ \frac{\partial \ddot{\theta}}{\partial u} \\ \frac{\partial \dot{\theta}}{\partial u} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.6 \\ 4.8 \\ 0 \end{bmatrix}$$

2. *Computing the control*:
    a. I used the equation from the instructions but ended negating K: u=-$K(x-g)$. With positive K, the cart would bug out and spaz everywhere. With negated K, the cartpole balanced perfectly.
    b. To compute the control in the code, I loop through K, x and goal matrix and arrays, applying the equation to each iteration and summing them in the control variable, which is returned. I spent a lot of time redoing my math from the previous step because I initially thought g was the gravity constant :(
3. *Tuning*: I ended up not needing to tune much since it was balancing almost perfectly from the first run. I did end up changing one of the Q values to 100 and R to 1, as recommended by one of the online resources I found, although it didn't change much from the default values.
4. *Conclusion*: The cartpole can instantly balance without much tuning and once it is balanced, it completely stops moving. I am not sure that's normal as I was expecting the cart to move back and forth but the math seems fine. I spent a few hours doing the explained steps above, including doing some reading.