Report By:

Lab Partner:

Lab TA:

Section:

# Final Project Prelab \_\_\_/15

You don’t need to use this template. The first part of this template contains a set of examples, and then the questions are at the end. The examples are designed to demonstrate how to determine if the open-loop and closed-loop systems are stable, how to determine the controllability of the system, and how to design the feedback gains to obtain the desired closed-loop system.

## Open-Loop Stability Example not graded

Assume we have the following system:



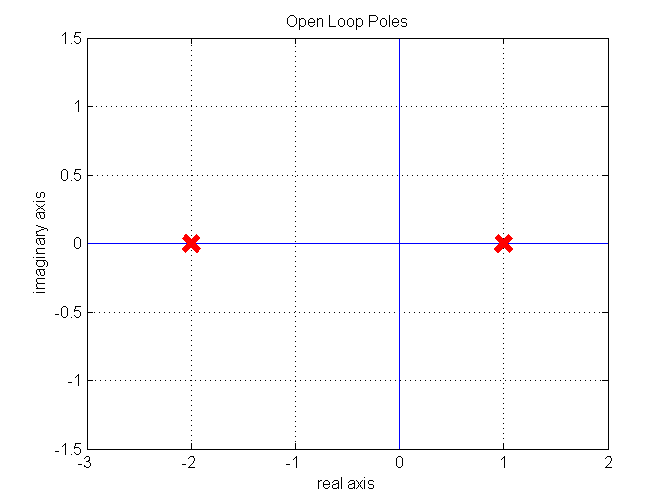
The first thing we would like to check is whether or not the open-loop system is stable. To do this, we first assume that **u** = **0** (since we’re considering open-loop). The system then simplifies to:





As we already know from our studies of transfer functions, we need to look at the poles of the system to determine if the system is stable or not. To find the poles of a system that is written in the form  where **M** is a matrix, find the eigenvalues of the **M** matrix. In our case, this can be done in Matlab by typing:

eig(A)

We find that the eigenvalues (and therefore the poles of the open-loop system) are 1 and –2. Since one of the poles is in the right half plane, the open-loop system is unstable. 

## Controllability Example not graded

Now, we would like to see if we could stabilize the closed-loop system by setting our control **u** = –**Kx** for some matrix **K**. First, we’ll need to check if the system is controllable by building the controllability matrix. In general, the controllability matrix is:



Since we only have 2 states, our controllability matrix is:



If this matrix is full rank (the rows are linearly independent), then the system is controllable. These same steps can be done in Matlab as well, by doing the following:

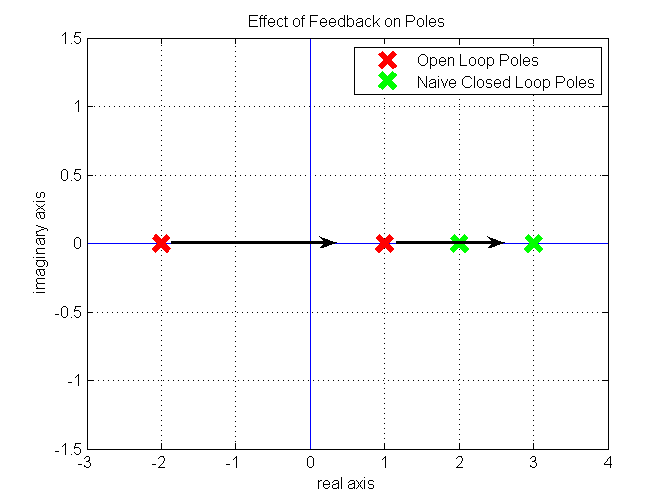
R = ctrb(A,B)

rank(R)

We find that the rank is 2. Since **R** is a 2-by-2 matrix, **R** is full rank. Thus, the system is completely controllable.

## Closed-loop Stability Example not graded

Because the system is completely controllable, then there exists a matrix **K** such that applying **u** = –**Kx** as our input will stabilize the system. Let’s naively pick **K** = [1 –6], and see if it works by plugging it back in to our original equation:





This matrix’s eigenvalues are 3 and 2. So, our control moved the poles even further right from 1 and –2 to 3 and 2. Clearly, we need to be careful with our choice of **K**. This leads us to an important point: we cannot conclude that our closed-loop system is stable just because it is completely controllable. We have to also make sure that we have chosen the correct **K** matrix.

Fortunately, Matlab gives us the ability to calculate a **K** matrix that will stabilize the system. In fact, not only can we stabilize the system, we can even choose what the closed-loop poles will be! Let’s say that we want the closed loop poles to be at –2 + *i* and –2 – *i*. We can then use the Matlab command *place* to tell us what our matrix **K** needs to be.

poles = [-2+i, -2-i];

K = place(A,B,poles)

We find that **K** needs to be the vector [5 3]. What this actually means is that the control **u** that will make the closed-loop poles be at –2 + *i* and –2 – *i* is:



However, we should double-check to see if this choice of **u** really does yield a stable closed-loop system. Just as we did above, let’s plug in **u** = –**Kx**:



Just as before, we need to find the eigenvalues of this matrix to determine if our new system is stable or not. We’ll do this in Matlab by typing:

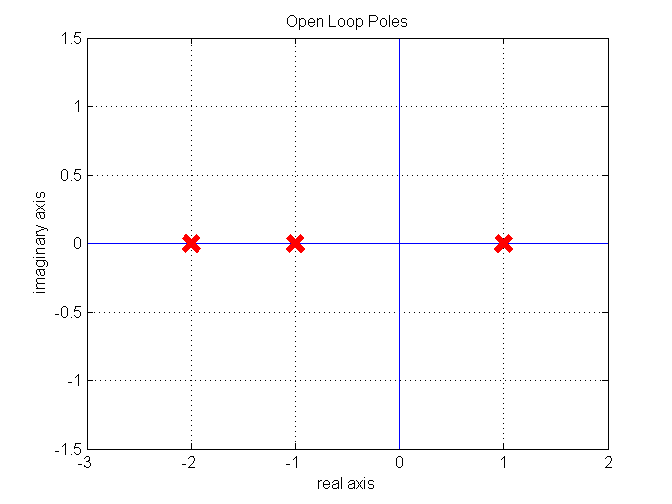
eig(A-B\*K)

Sure enough, the eigenvalues are –2 + i and –2 – i, as we desired, and the closed-loop system is stable.

## closedlooppolesUncontrollable System Example not graded

Let’s do another example. Assume:





First, let’s check to see if this open-loop system is stable by checking the eigenvalues of A:

eig(A)

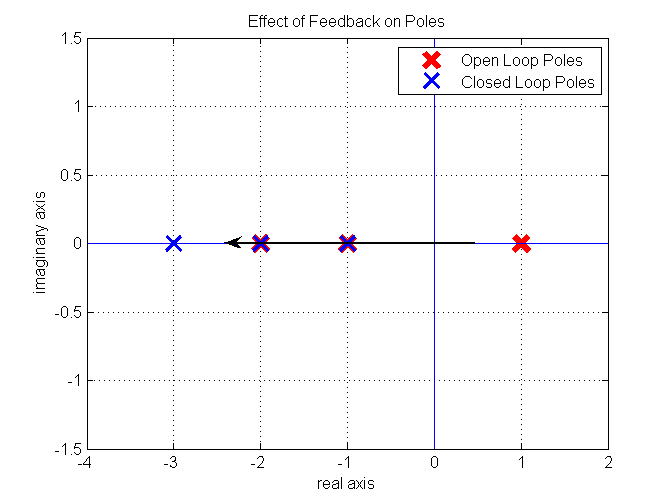
The eigenvalues of A are -2, -1, and 1. As shown at right, one of the poles is in the right half plane, and so the open-loop system is unstable. Therefore, we should check if the system is controllable. We can build our **R** matrix as above to get:



Since **R** has rank 1, it is not full rank. Because it is not full rank, the system is not completely controllable. However, this does not necessarily mean that the system is not *stabilizable*. For example, if we take K = [0 0 4], then our closed loop system becomes:



This matrix has eigenvalues -3, -2, and -1, which are all in the left half-plane. Therefore, our closed loop system is stable.



These last two examples show us the relationship between controllability and stabilizability. If our system is controllable, then we are guaranteed to be able to find a K matrix that will make our closed-loop system stable. If our system is not controllable, we are not guaranteed to be able to find a K matrix that stabilizes our closed loop system, but such a K matrix might still exist.

# Questions \_\_/15

For these problems, you may use Matlab for any calculations that you like. However, if you do use Matlab, turn in your Matlab code or print out the command window where you performed the calculations. Assume we are using the following system:



## (A). Determining Open-loop Stability \_\_/4

Find the poles of the open-loop system. Is the open-loop system stable?

## (B). Determining Controllability \_\_/4

Find the controllability matrix R of this system. What is its rank? Is this system completely controllable?

## (C). Designing Feedback Gains \_\_/3

Using the place command, find the K matrix that makes the closed loop poles be -2, -19, and -38.

## (D). Verifying Closed-loop Stability \_\_/4

By substituting in your answer for K from the previous question, verify that our closed-loop system does have the desired poles, and is stable.