Cody Pappa

MAE 144 HW1

1. https://github.com/CodyPappa/Classes

2. A. I set up my polynomials a, b, and f and called RR_ Diophantine to find the solution to the Diophantine equation with the lowest order for Y. My residual error is basically 0, therefore the code worked to produce a solution.
   B. Looking at my TF D, it currently has 5 zeros and 3 poles, which makes it not proper. Adding k=5 poles at s=-20, makes my new TF D1 proper, with 5 zeros and 5 poles

3. My code takes the input transfer function, h, and if requested, omegabar and a strictly-causal flag in the form of 1 or 0. The code assumes a strictly-causal output, so it sets sc=1. If a semi-causal is requested, set sc=0. Omegabar is assumed to be 0, but may be any exact frequency or interest to focus the gain of the system. In relation to the provided MATLAB code, my version gives the freedom of setting the two options above for increased usefulness in different applications. The downside to my code, however, is that is does not function if the transfer function has a pole at zero and the inputted omegabar is 0. This is due to the gain matching at omegabar, which is not useful when the original gain is infinity.
   The test code with the provided values satisfies the above requirements, so it creates an error. I also included another code that does show that my code and the matlab version have the same output, following the example given in the Brain Douglas video that was linked.