

**EE-370L: Control Systems**

**Lab #3: Basics of   
Control System in MATLAB and Simulink**

**Professor: Jula P.H.D**

**Joshua Coleman**

**Dean Dennis**

**Micheal Atia**

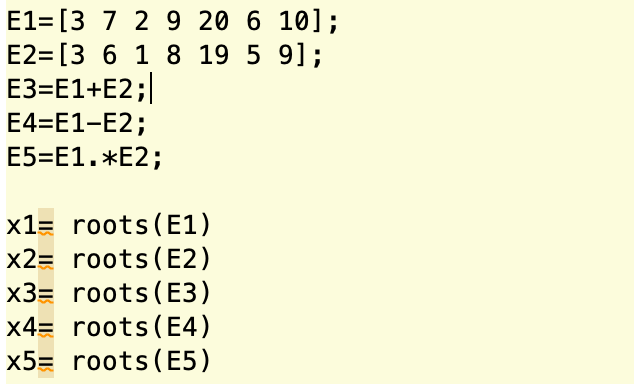
**Abstract:**

The Purpose of Lab 3 was to familiarize students with MATLAB and Simulink capabilities in regards to transfer functions and closed loop systems. MATLAB’s capabilities were demonstrated through the following applications : determining roots of polynomials, determining polynomial coefficients, converting between expanded form and zero-pole-gain (zpk) form and handling transfer function operations, working with partial fraction decomposition and inverse Laplace operations as well as plotting these types of functions. In regards to Simulink, the following functionalities were showcased: plotting with simulink, Simulink’s Linear System Analyzer which was used to plot plot and analyze various parameters such as impulse response, rise time, percent overshoot, etc., and block routing with closed loop systems containing negative feedback components.

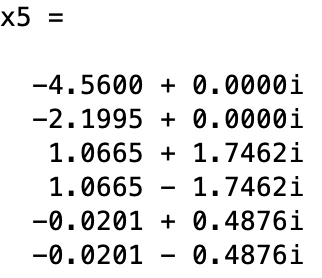
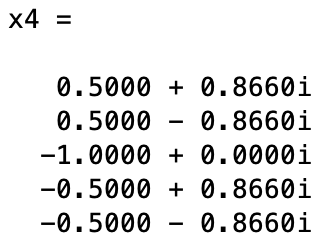
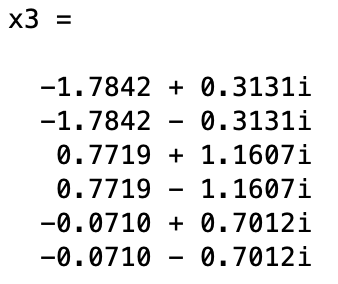
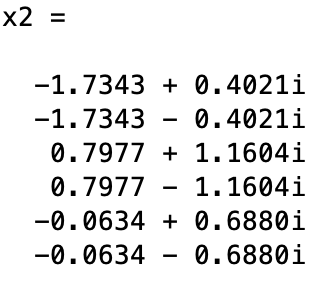
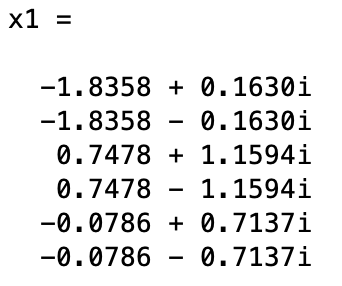
**Data:**

2.1.1 Roots and polynomial operations

Code:



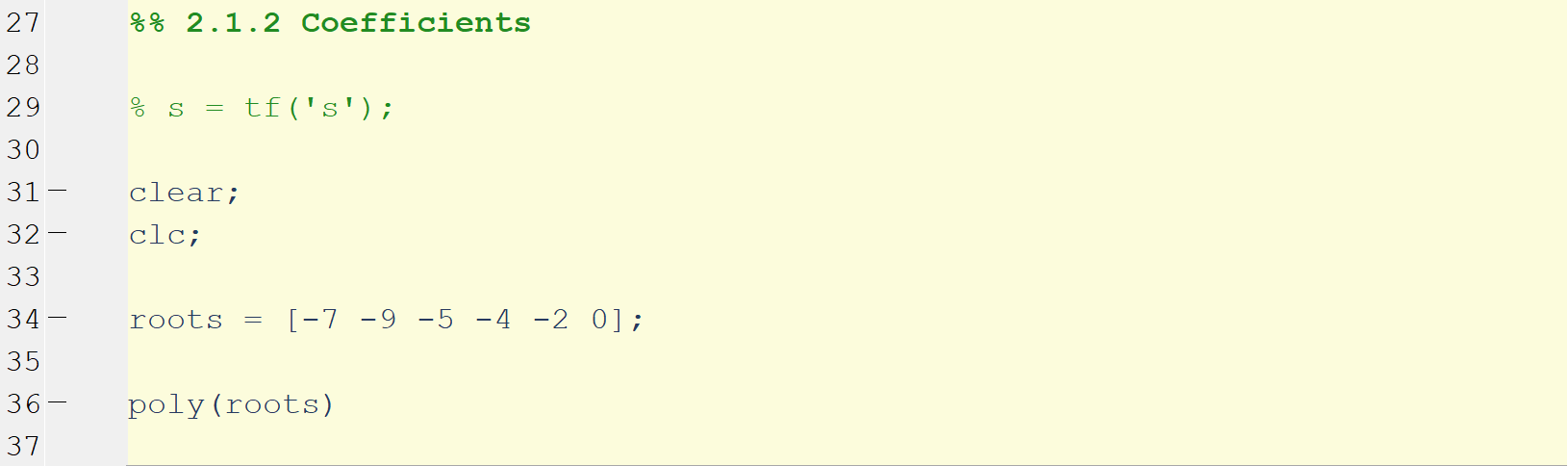
Output:



Discussion: Having the code written as a vector, matlab is able to differentiate the magnitude of each variable in descending order from left to right. The addition, subtraction and multiplication are done so via the code and since the vectors are the same magnitude we get the appropriate response.

2.1.2 Coefficients

Code:



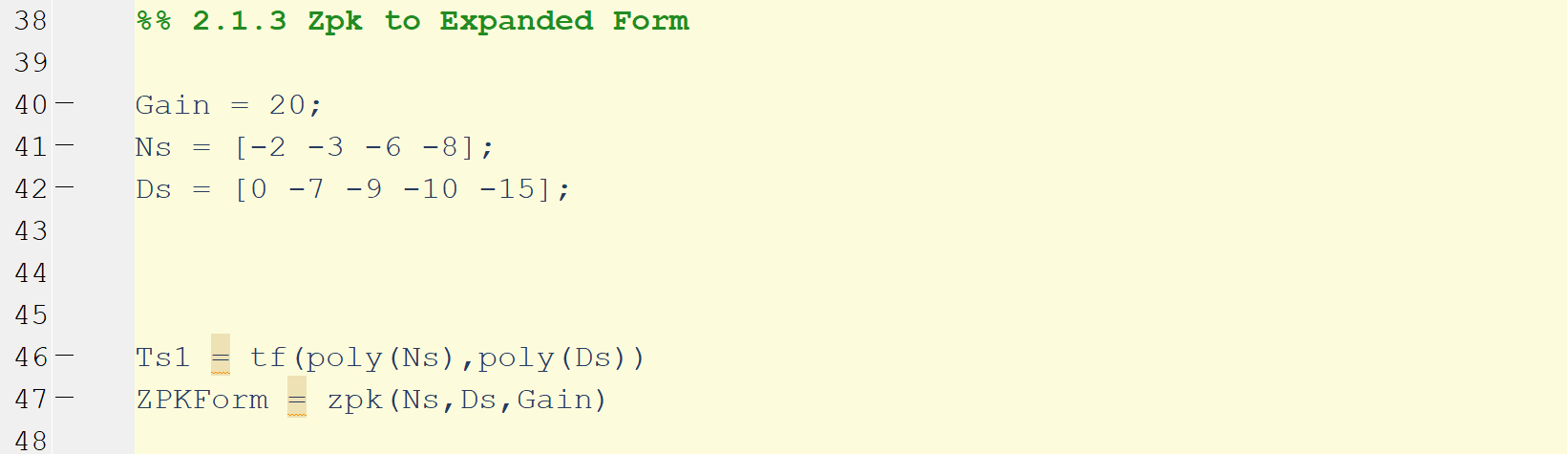
Output:



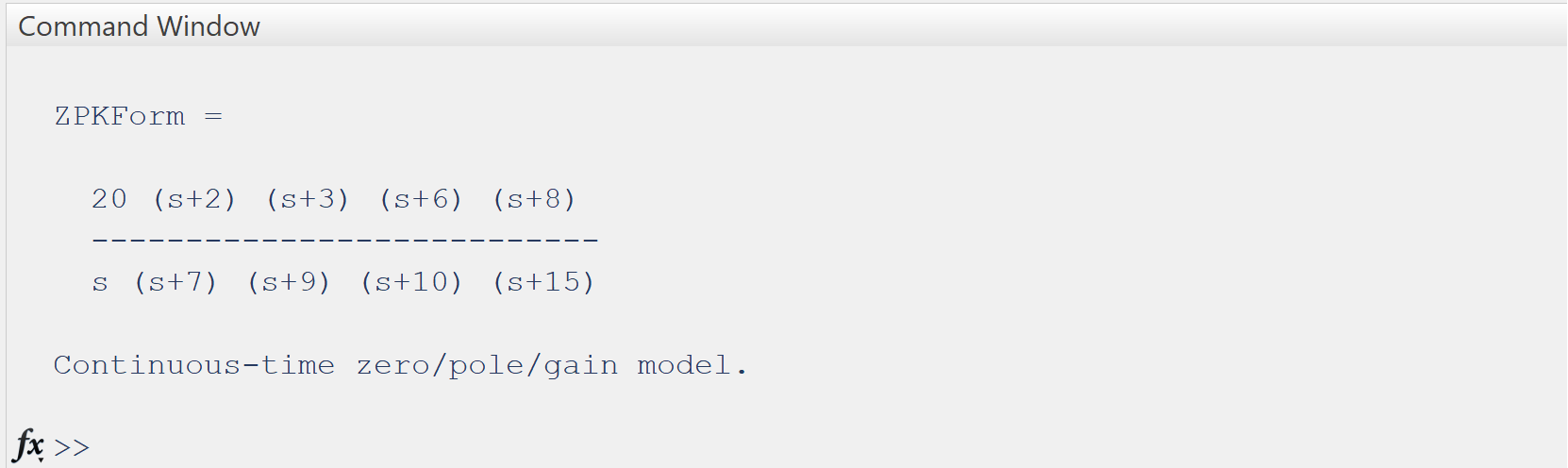
Discussion: Essentially, we create a vector with a set of variables in congruence with the “zeros” of the function given and use the *poly()* function to get the proper results requested.

2.1.3 Zpk to expanded form

Code:



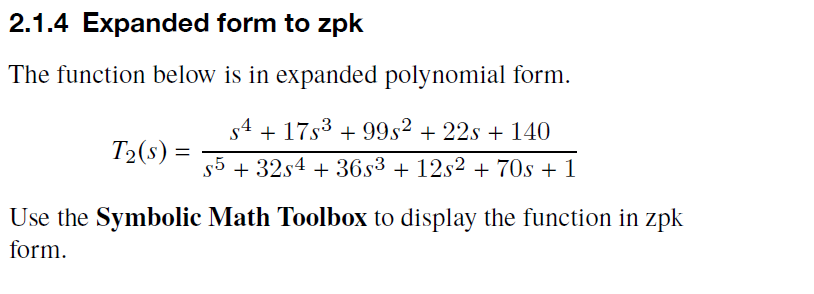
Output:



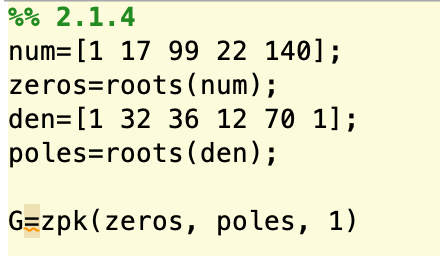
Discussion: Having set a vector and value for the zeros, poles, and gain, we can use the function *zpk(x,y,z)* to get the proper output on the command window. Where x,y,z is the zero, pole, and gain variables in that order.

2.1.4 Expanded form to zpk

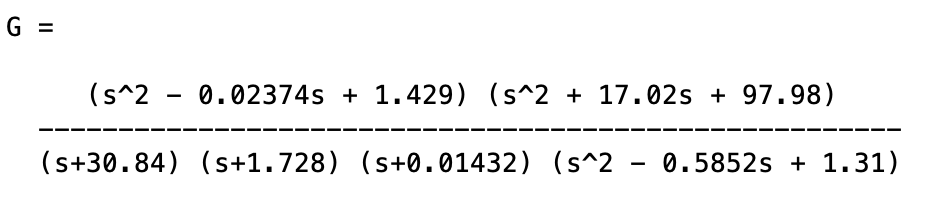
Prompt:



Code:

**

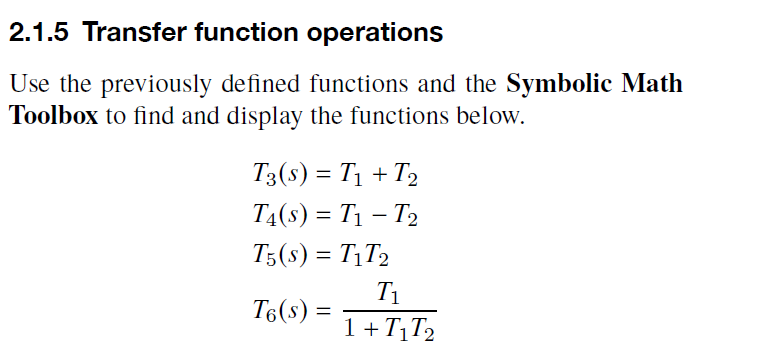
Output:

**

Discussion: We are to take the transfer function and expand the polynomials in the numerator and denominator and display it in zpk form. To do so, we get the roots of the numerator and denominator separately. Use those as the zeroes and poles respectively and then put them into the *zpk()* function giving us the desired output.

2.1.5 Transfer function operations

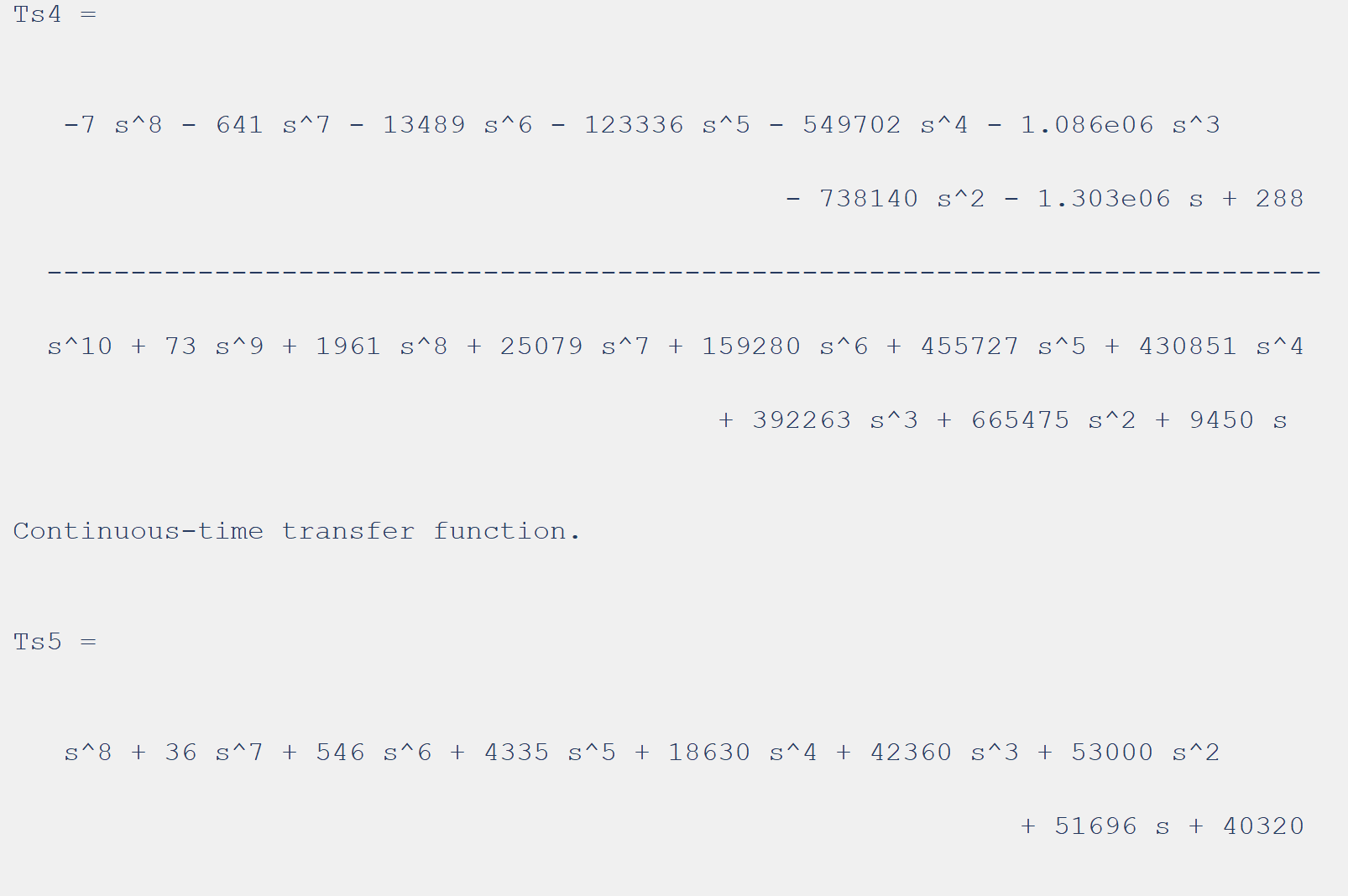
Prompt:

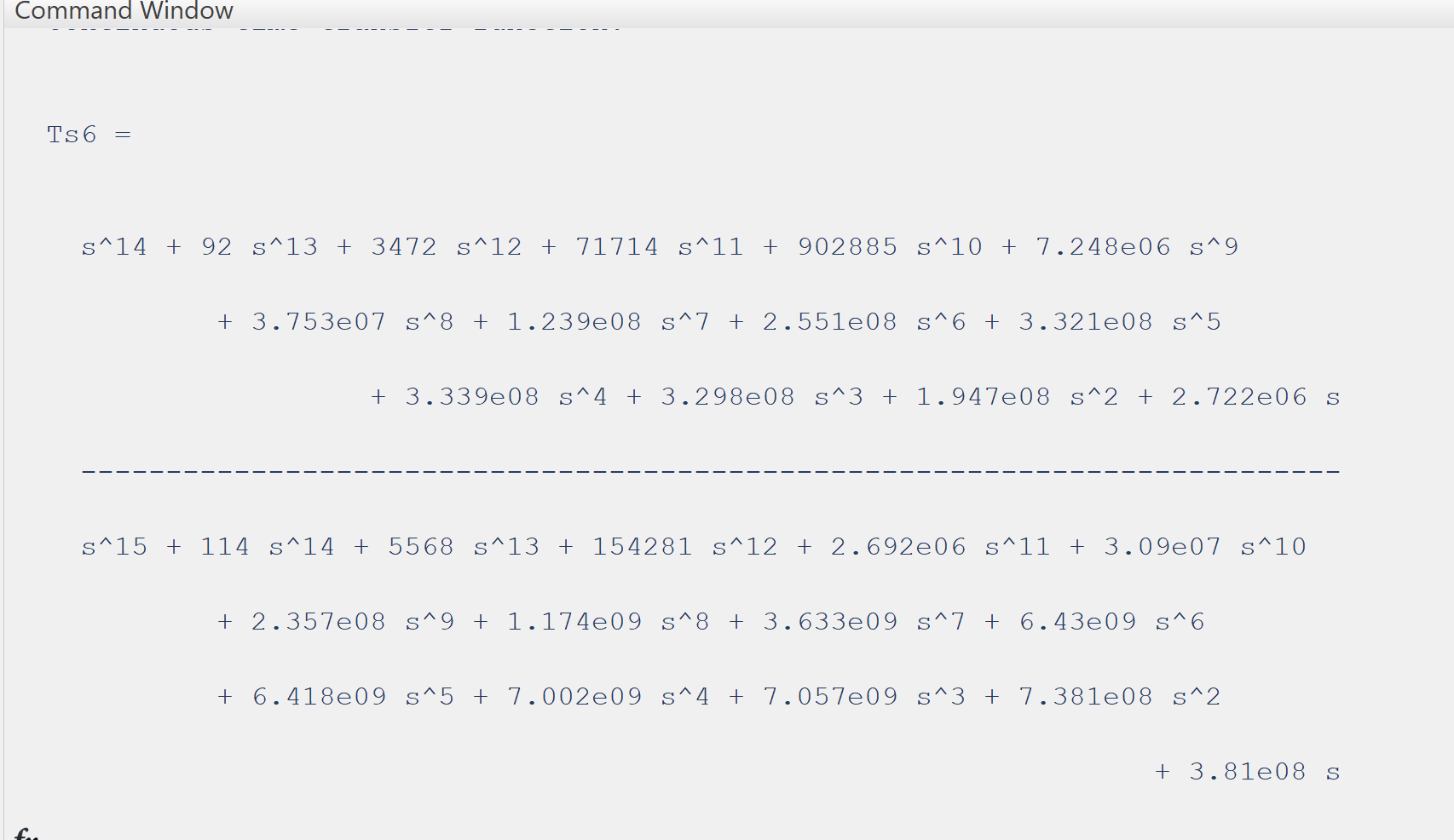


Code:



Output:

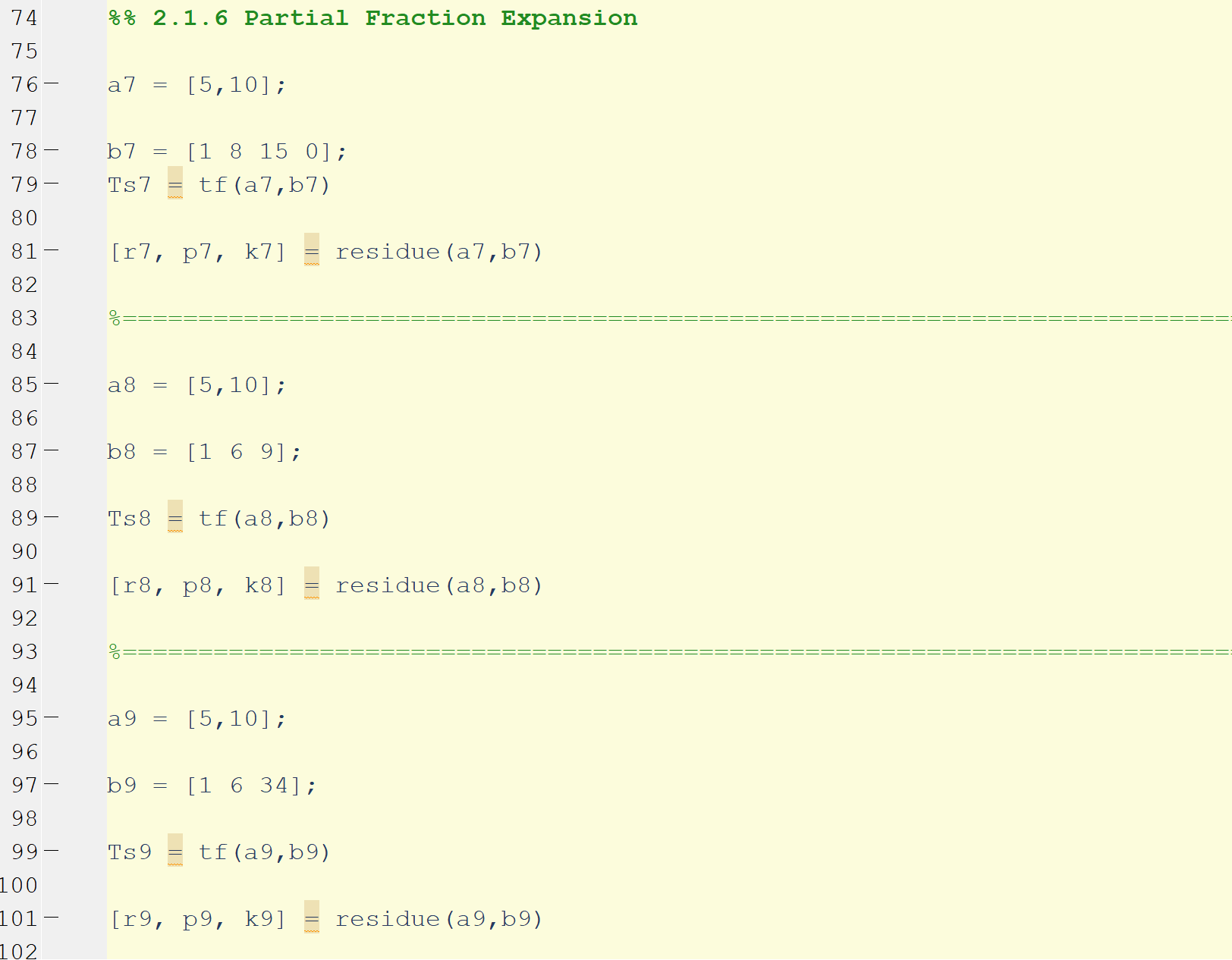




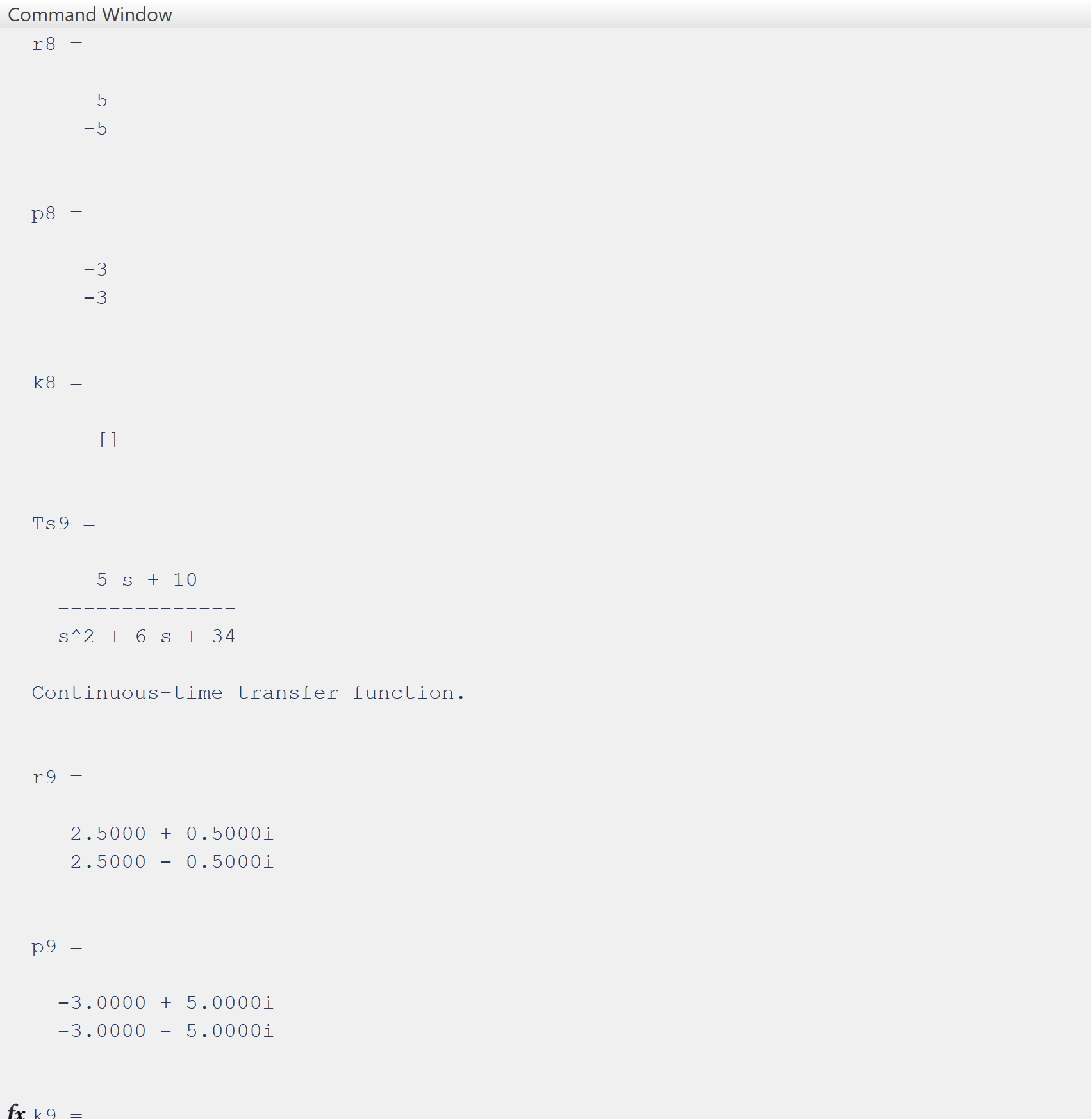
Discussion: From all the previous part, we take the transfer functions and add, subtract, multiply, and divide. The code is self explanatory.

2.1.6 Partial fraction expansion

Code:

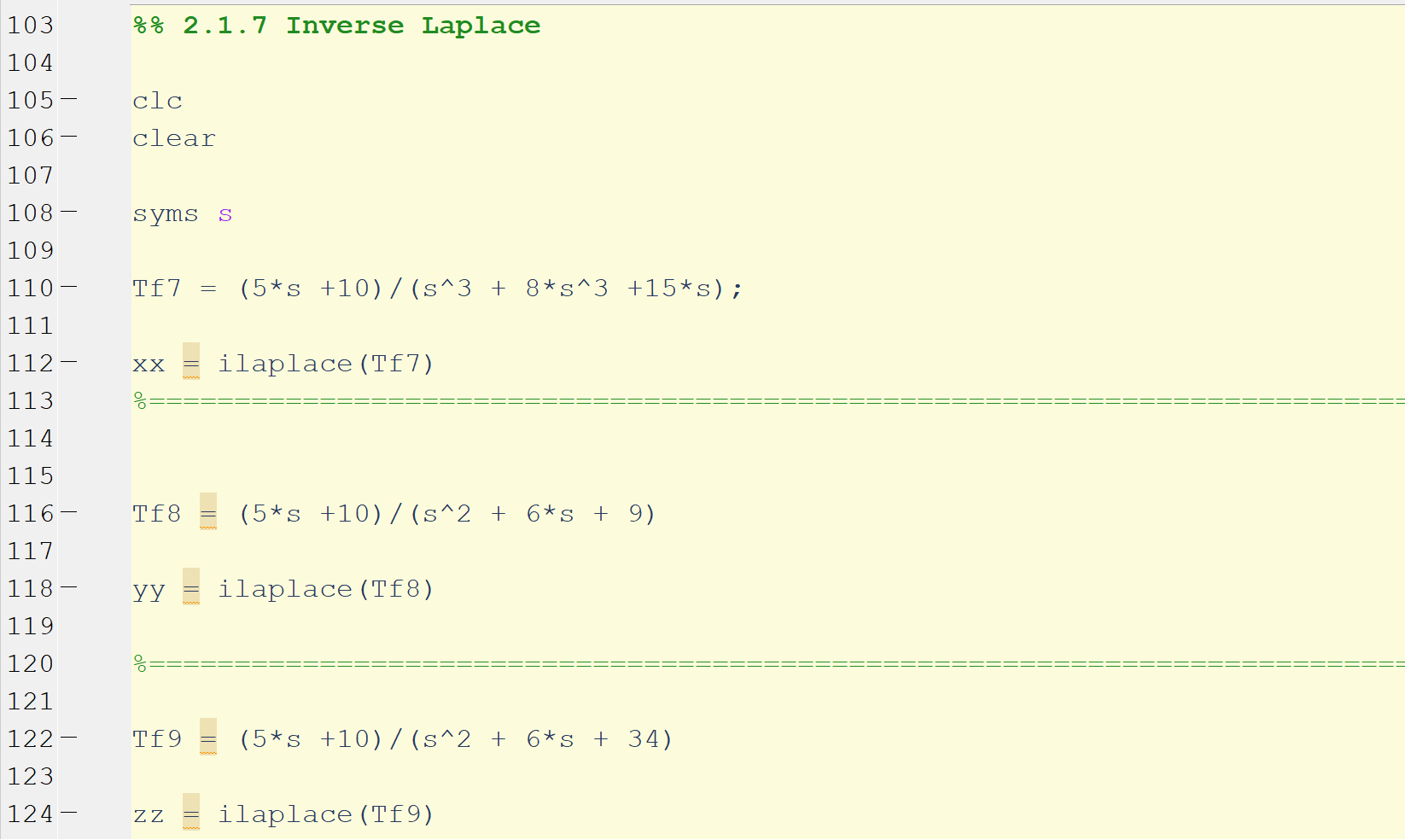


Output:

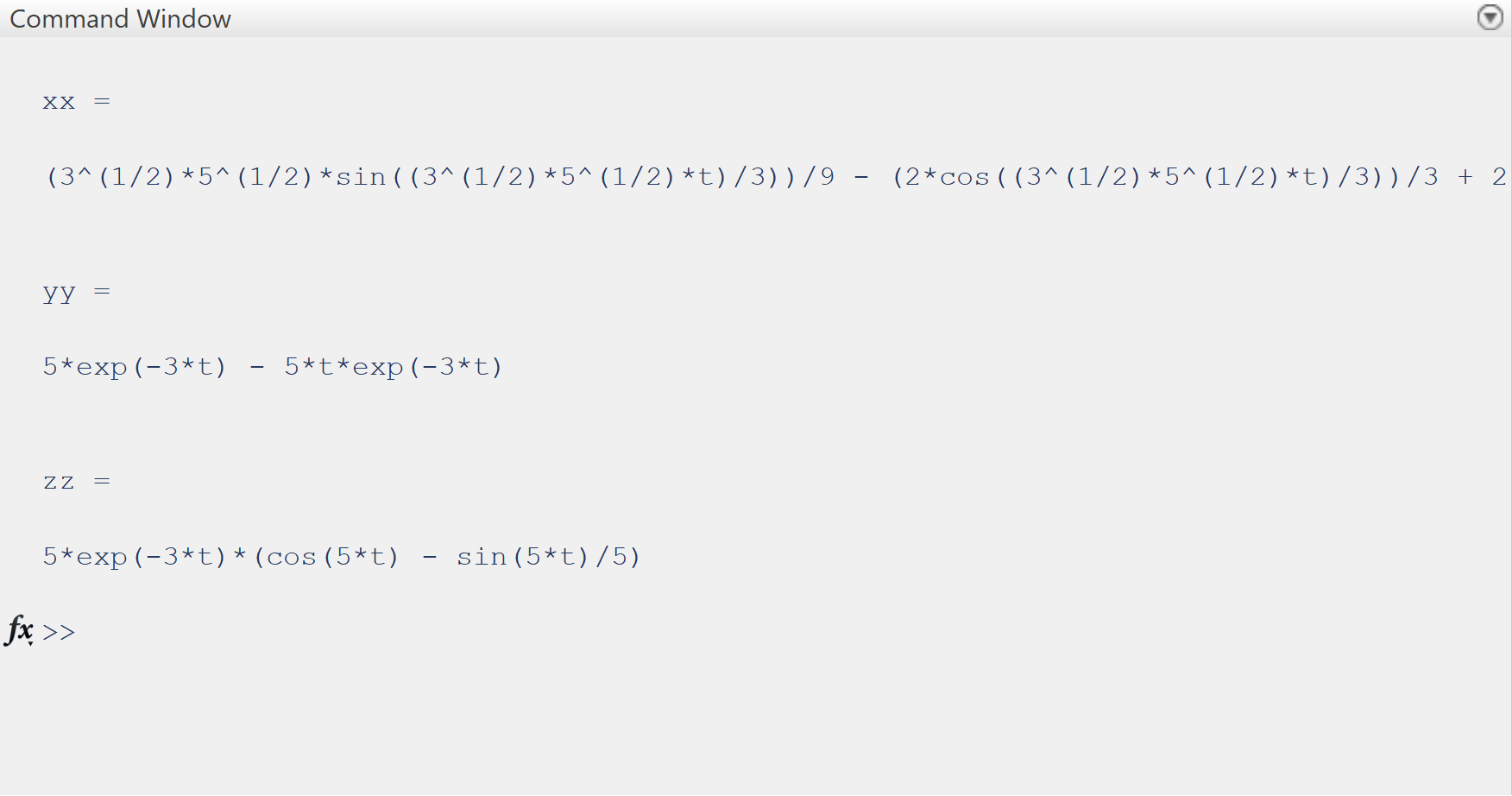


2.1.7 Inverse laplace

Code:

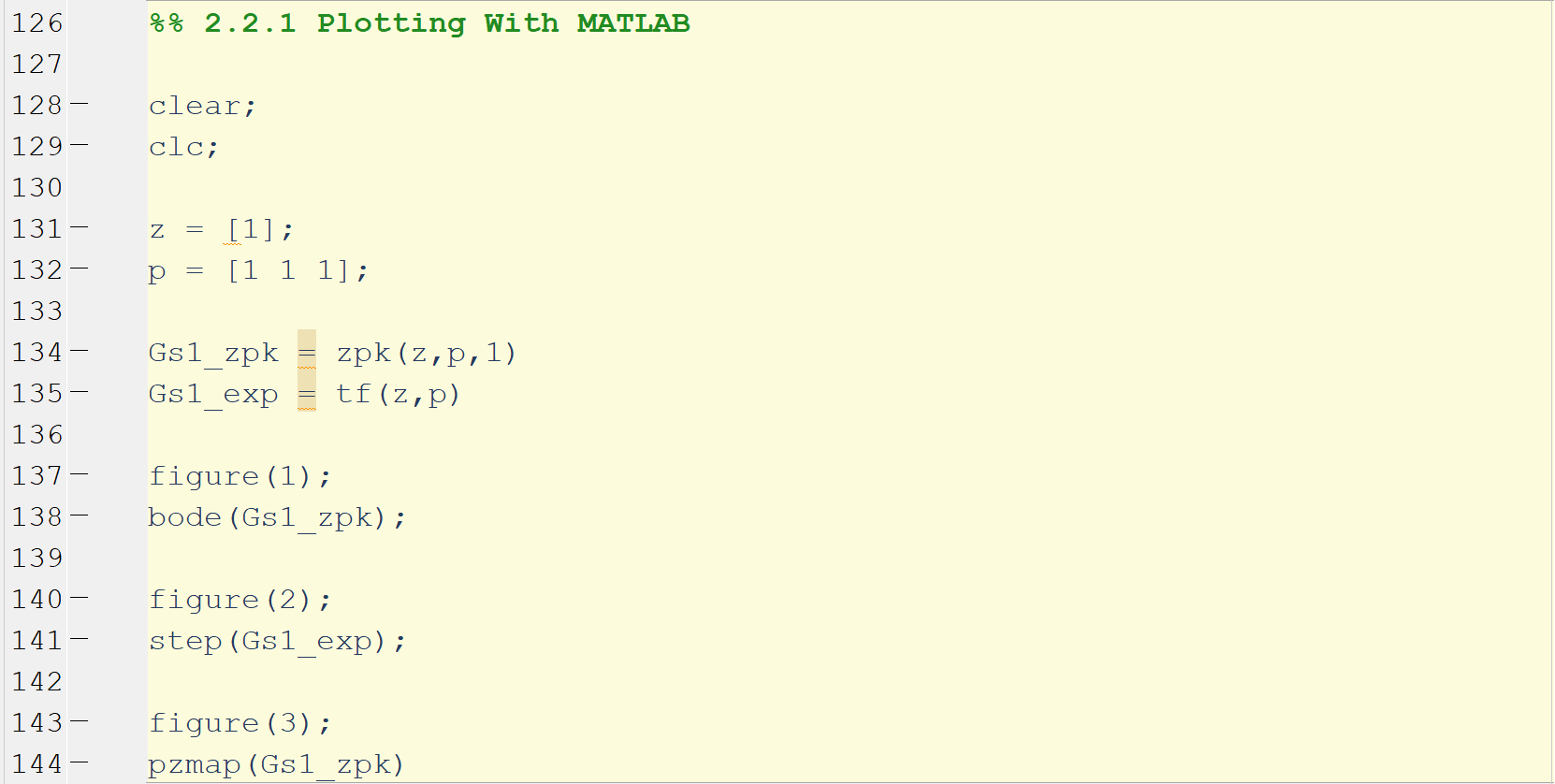


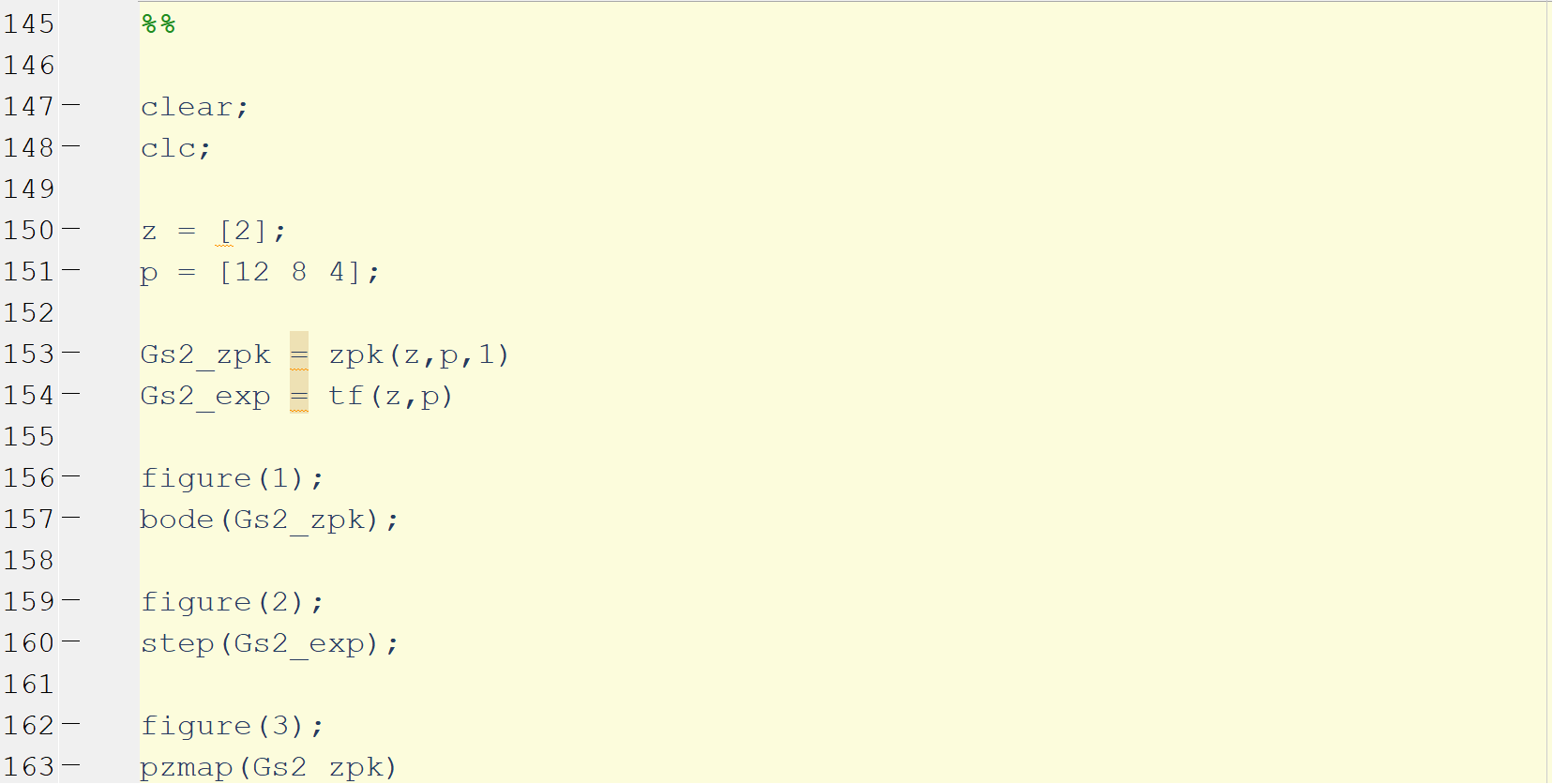
Output:

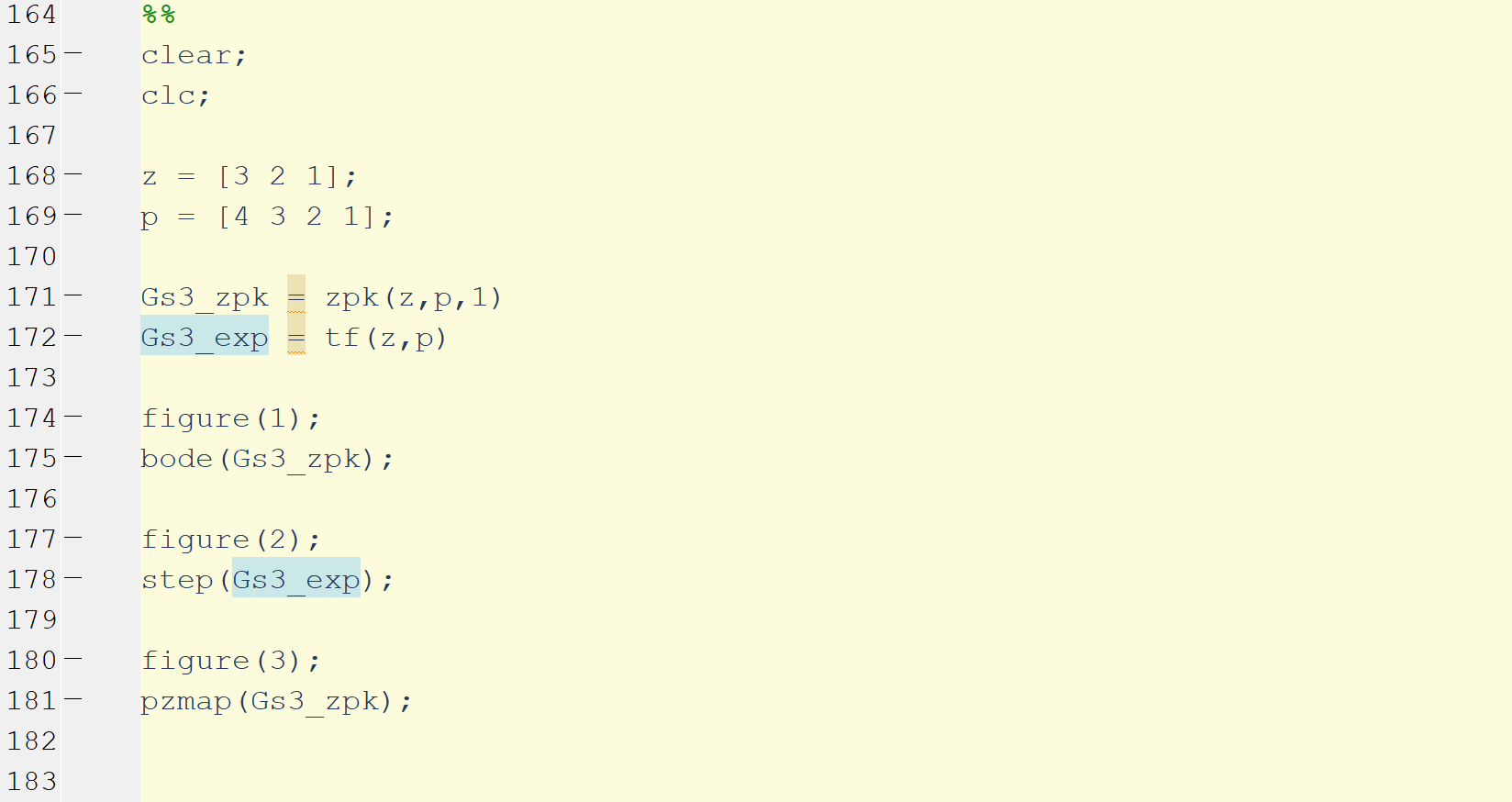


2.2.1 Plotting with MATLAB

Code:



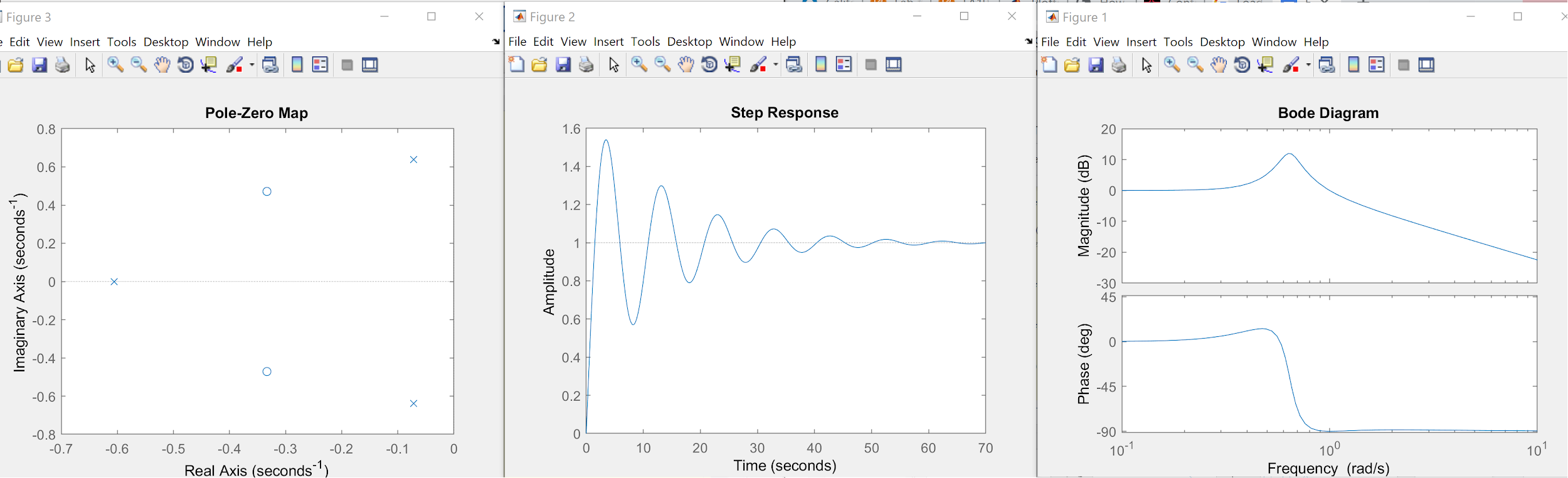




Output:





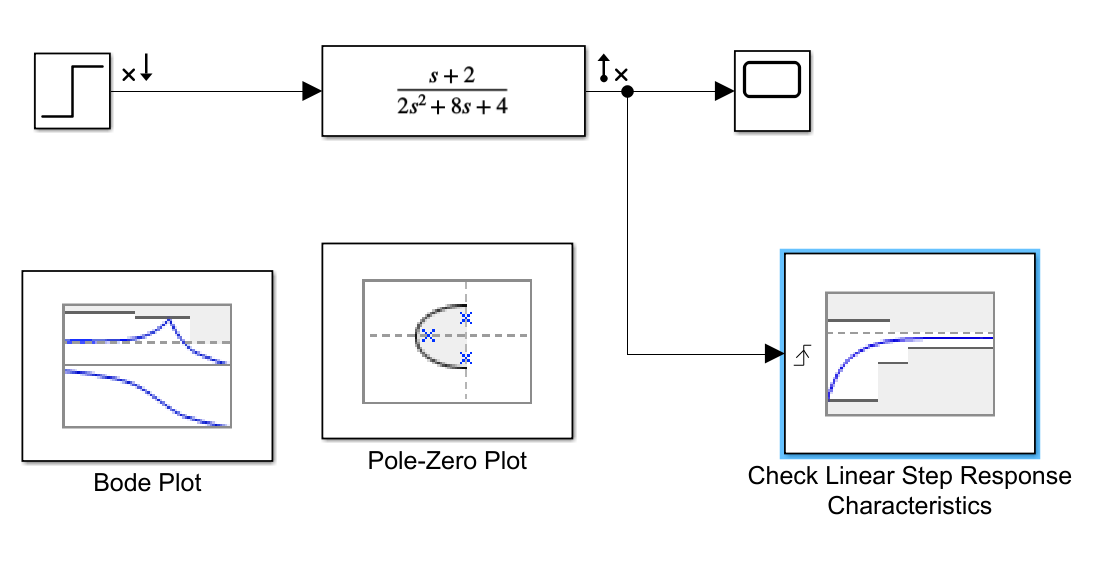


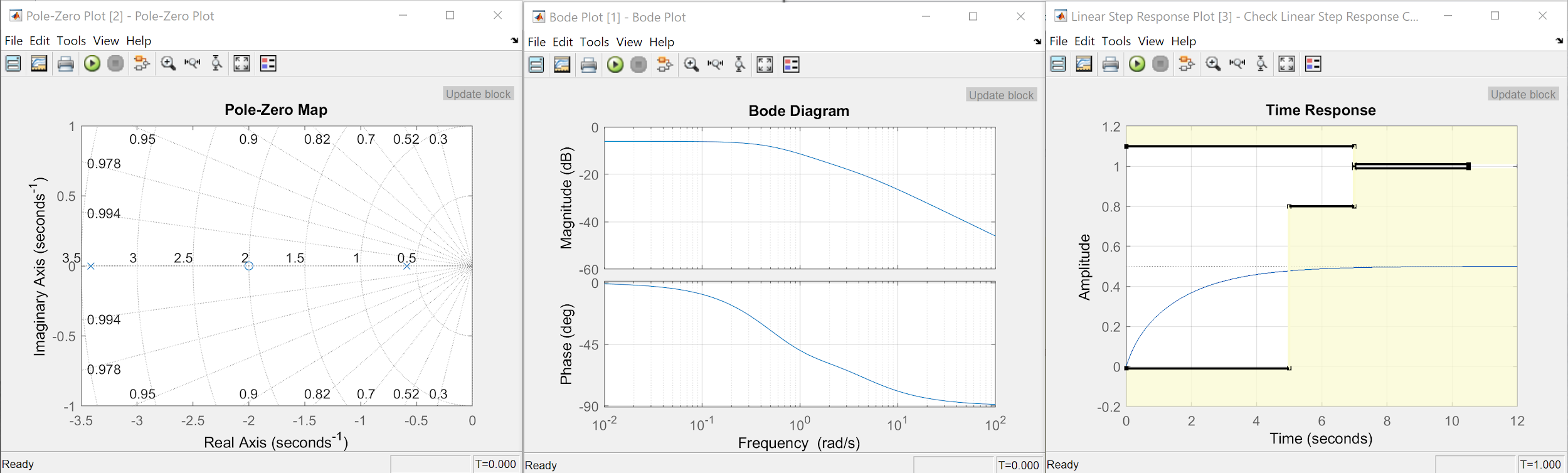
Discussion:



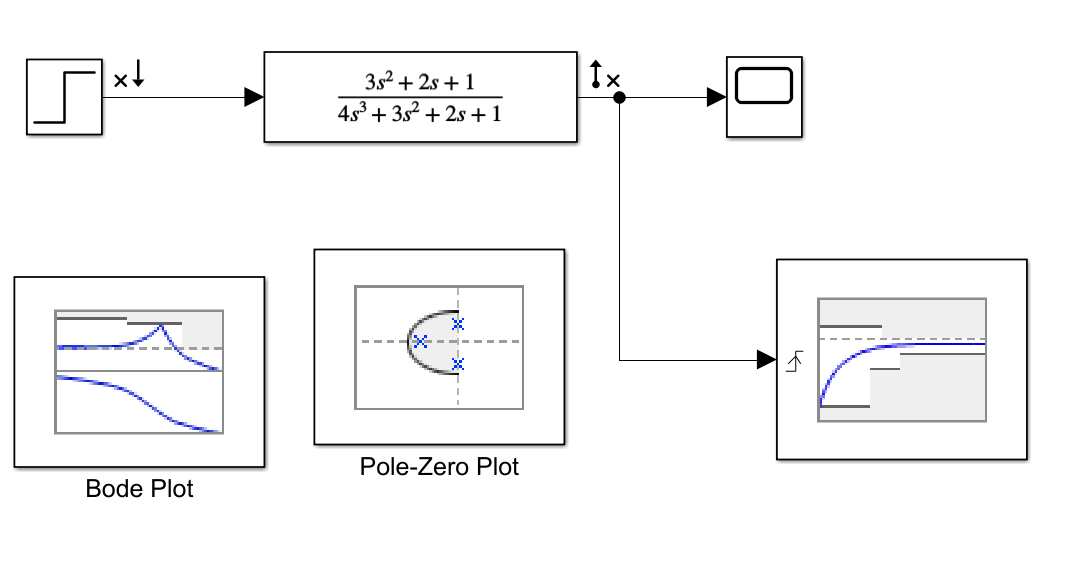


2.2.2 Simulink Gs1





2.2.2 Simulink Gs2



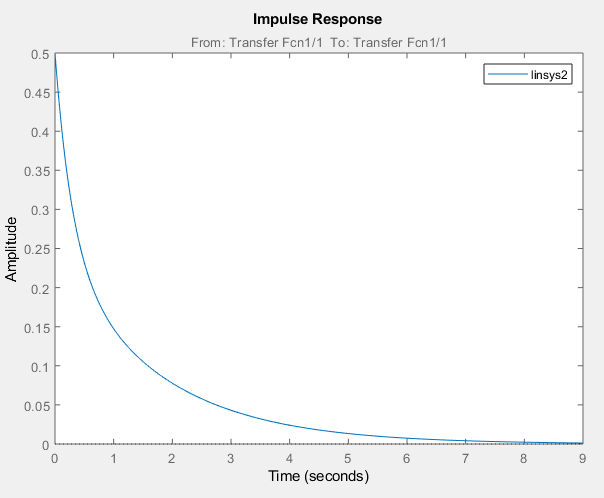


2.2.2 Simulink Gs3

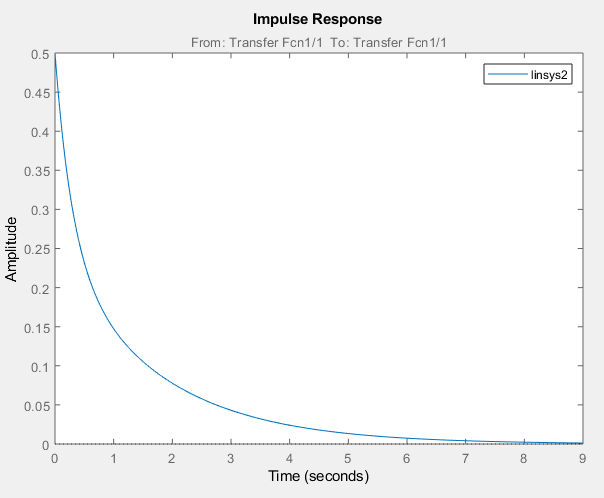
**2.3 Linear System Analyzer**

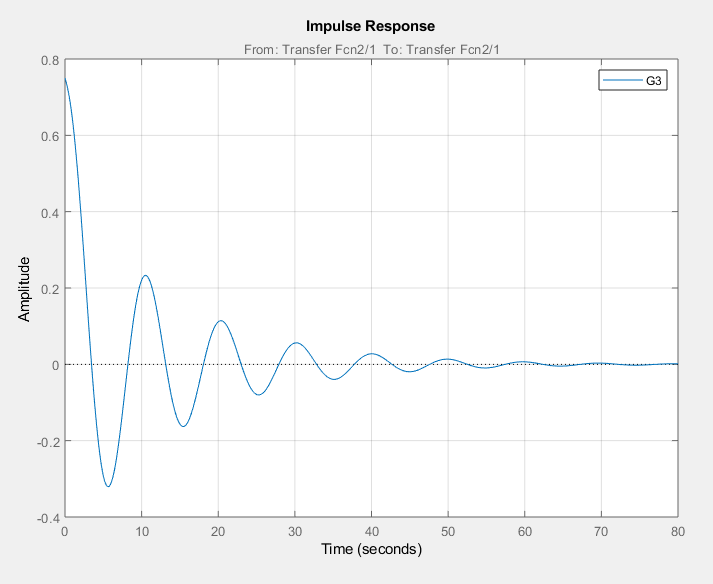
**Discussion**: Using the Linear System Analyzer output the impulse response, nyquist diagram of each transfer function G1, G2 and G3 and find the peak time, settling time, rise time and % overshoot.

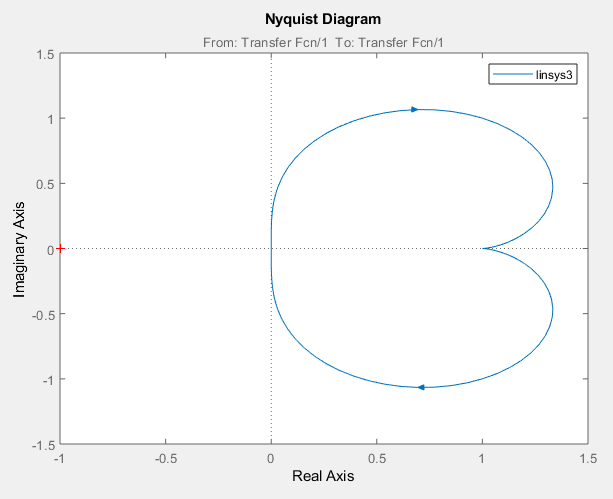
**Figure 2.3.1 Impulse Response for G1**

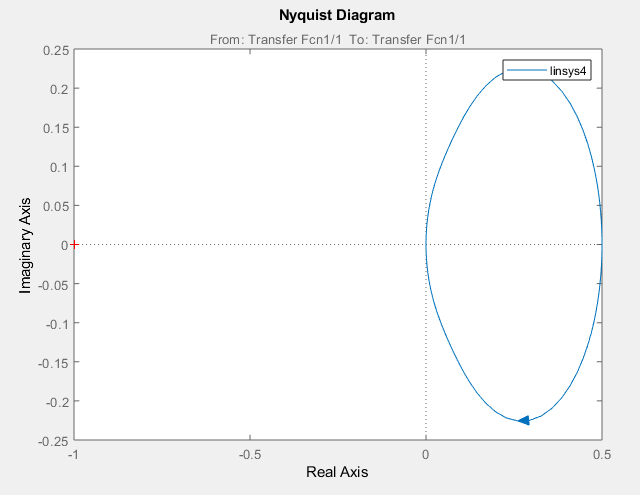


**Figure 2.3.2 Impulse Response for G2**

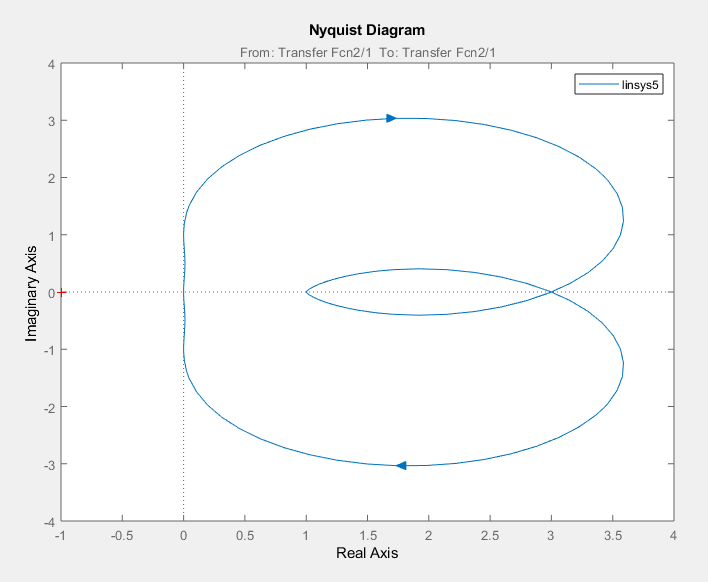


**Figure 2.3.3 Impulse Response for G3**

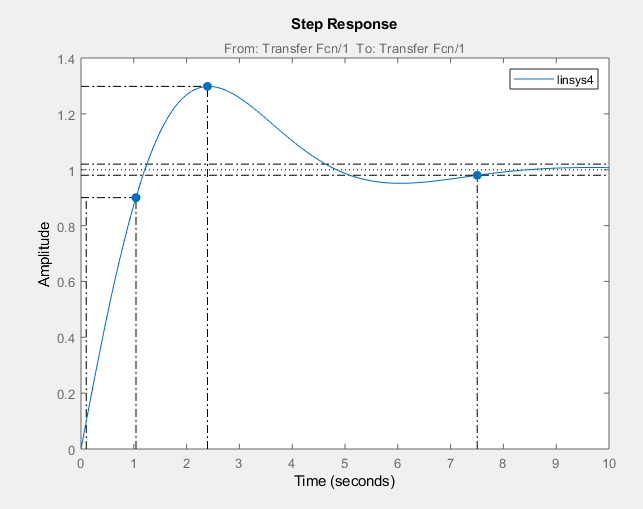
**Figure 2.3.4 Nyquist Diagram for G1**

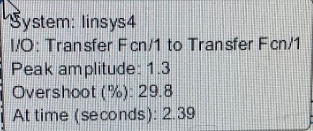
**Figure 2.3.5 Nyquist Diagram for G2**

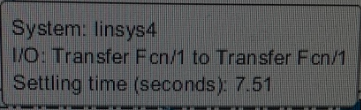
**Figure 2.3.6 Nyquist Diagram for G3**



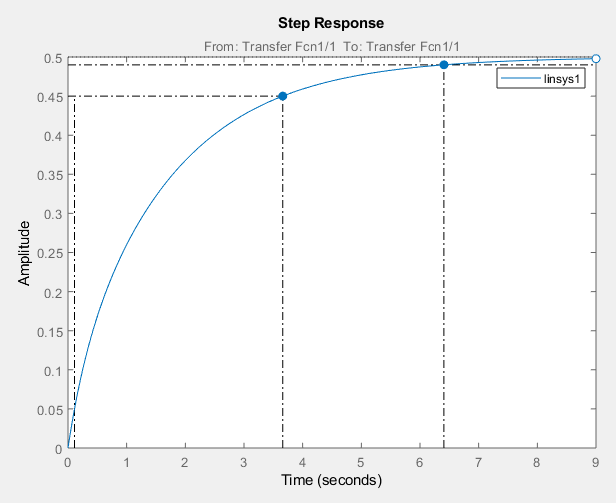
**Figure 2.3.7 Step Response for G1 with rise time, peak time, settling time and %overshoot.**

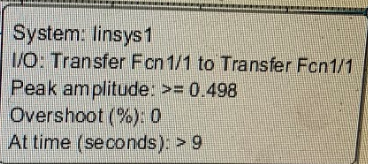
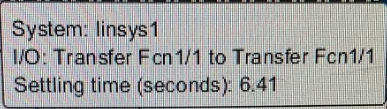
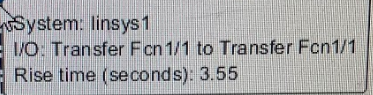




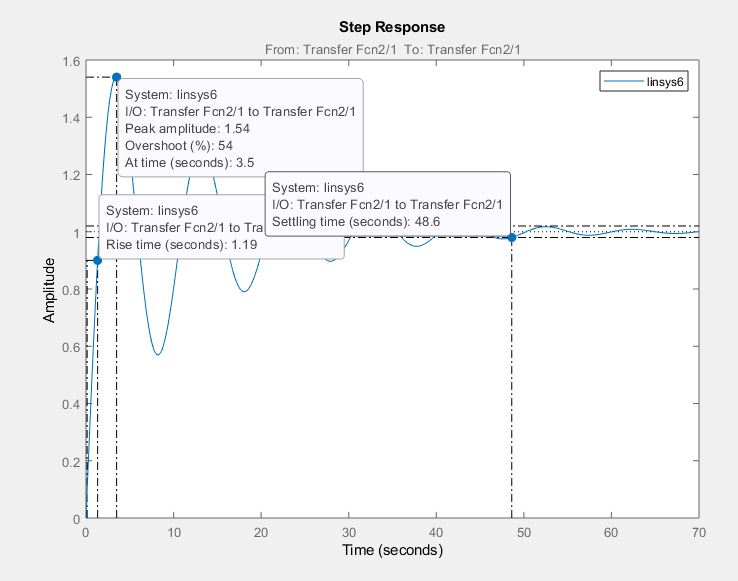


**Figure 2.3.8 Step Response for G2 with rise time, settling time, peak time and % overshoot**





**Figure 2.3.9 Step Response for G3 w/ %overshoot, rise time and settling time and peak time.**

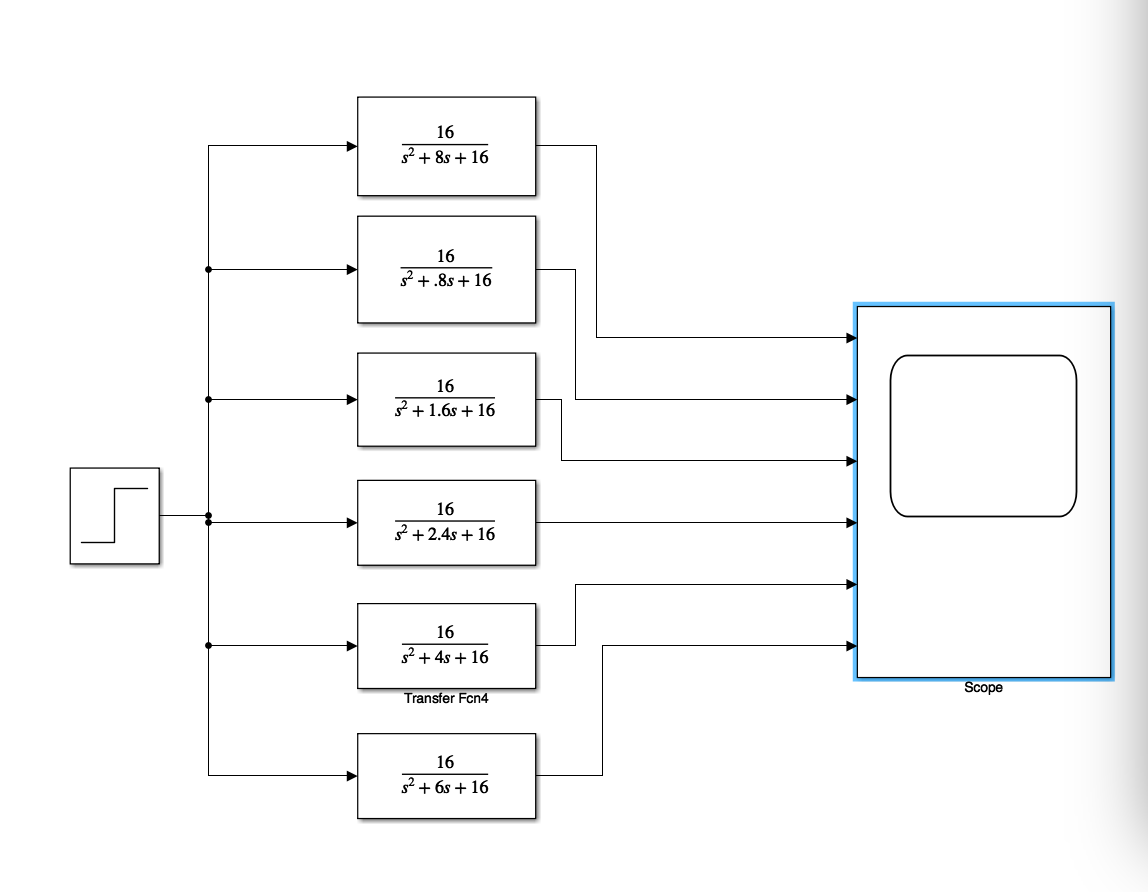


**2.4 Harmonic Oscillation**

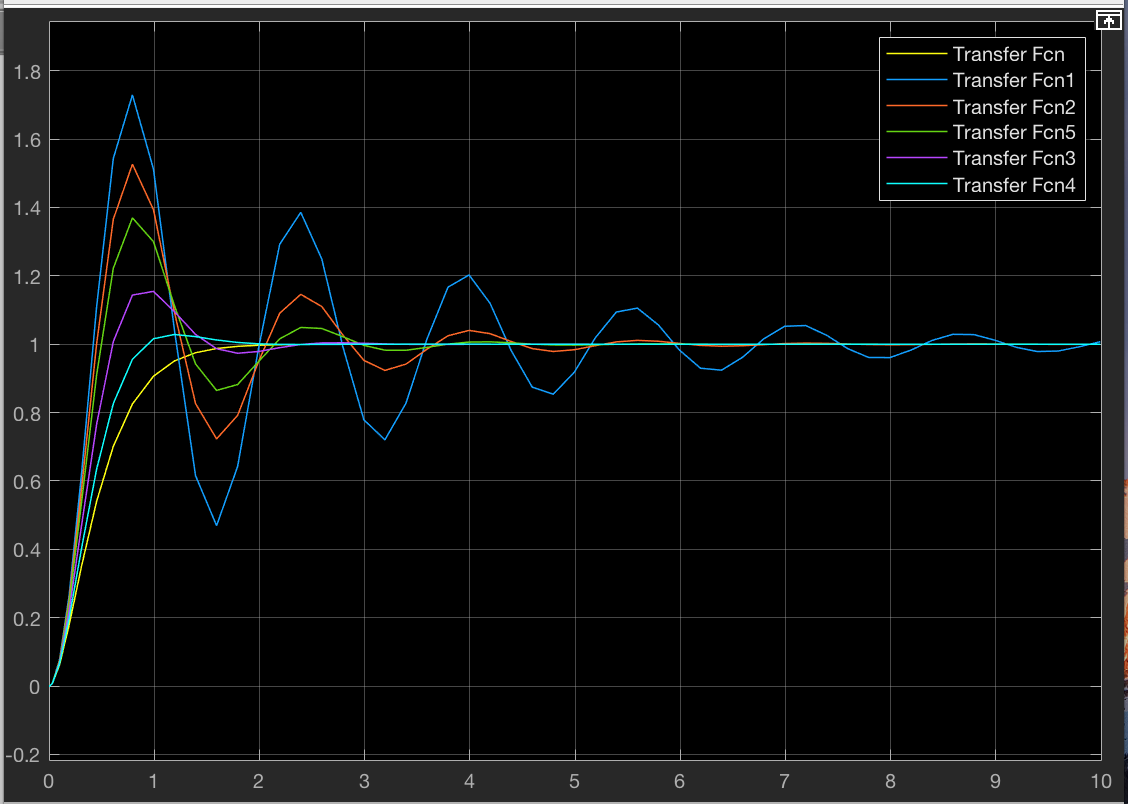
**2.4.1** **Changing the Damping Ratio**

Discussion: Build a block diagram that outputs the step response for each transfer function with . The first transfer function G(s)= has

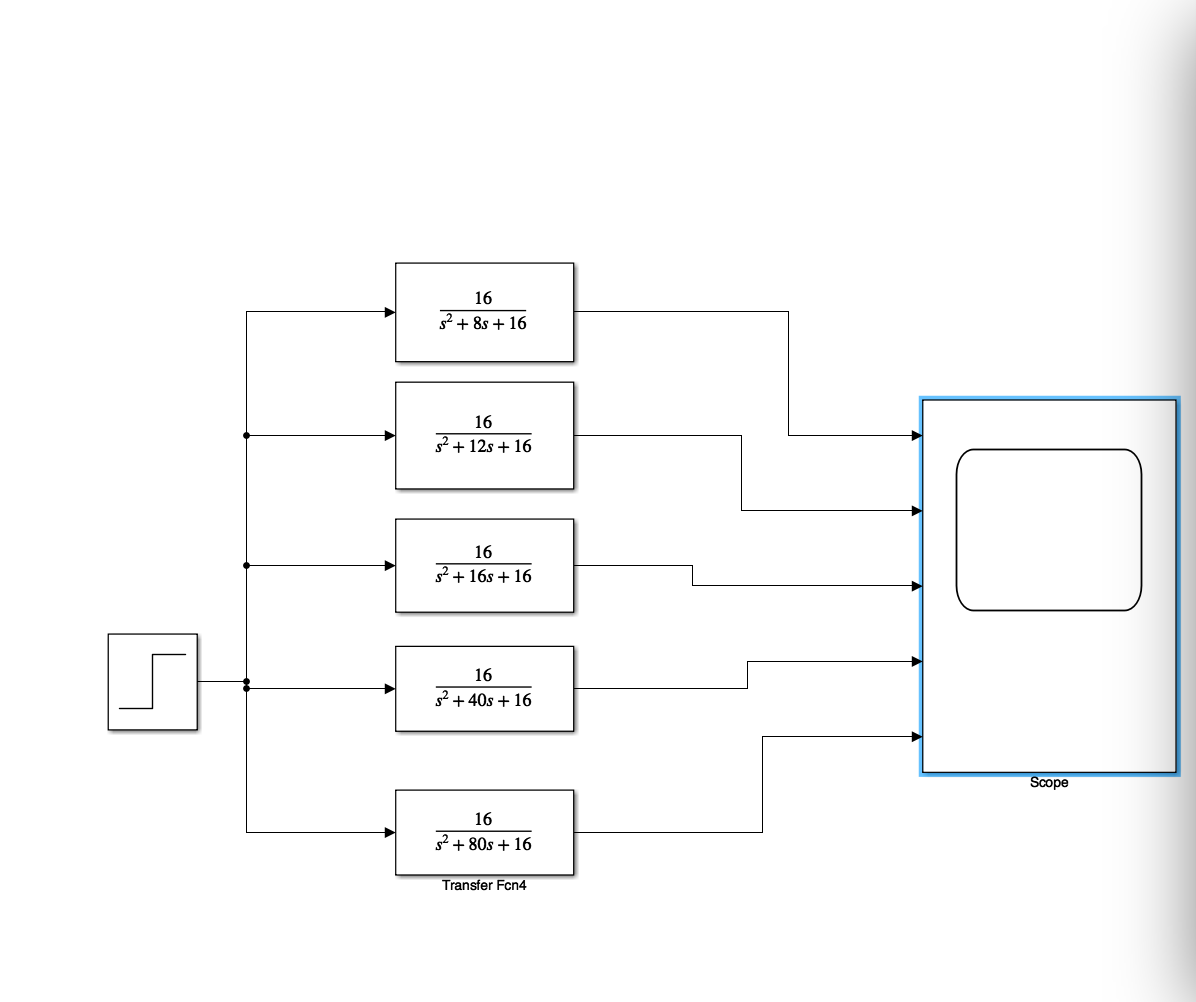
**Figure 2.4.1**



**Figure 2.4.2 Step Response Outputs of figure 2.4.1**



**Figure 2.4.3 Block diagram that outputs the step response of each transfer function with**



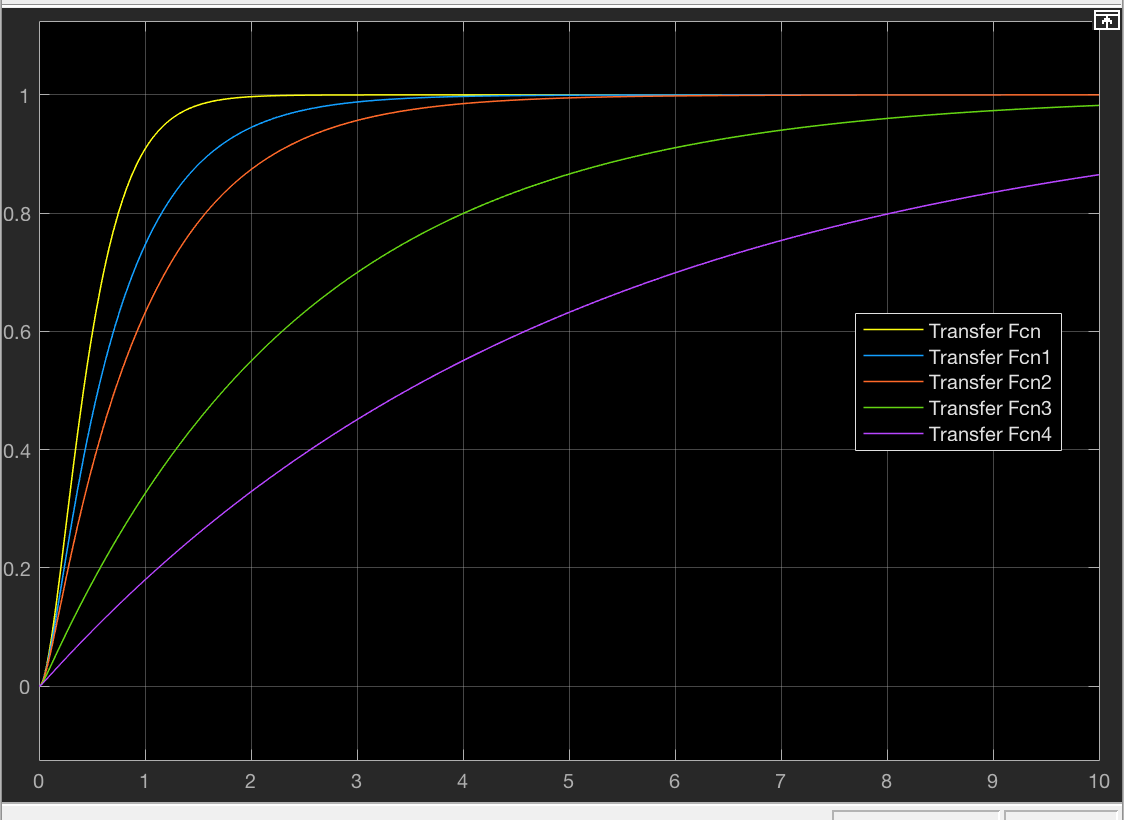
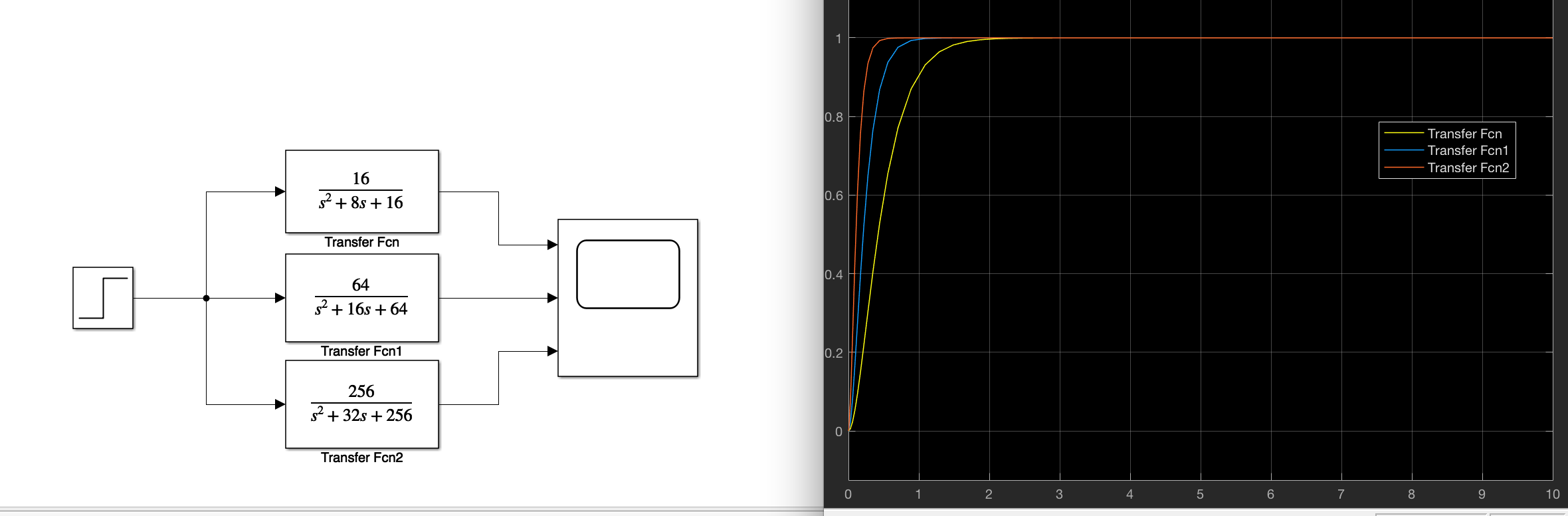
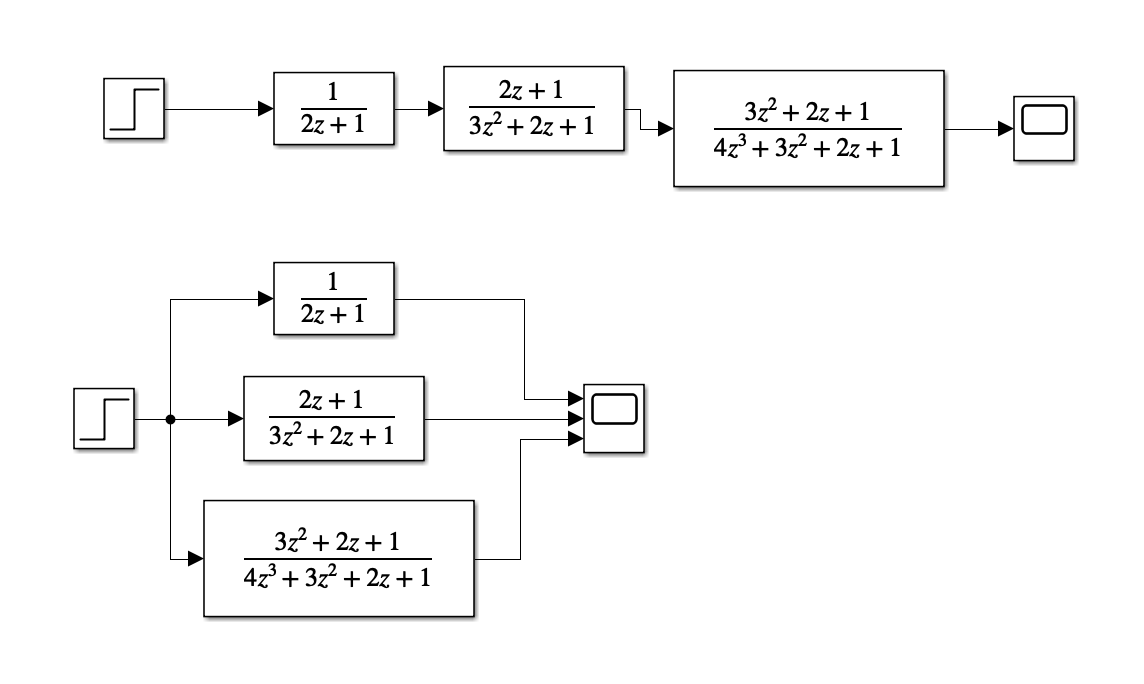
**Figure 2.4.4 Step Response output from figure 2.4.3**

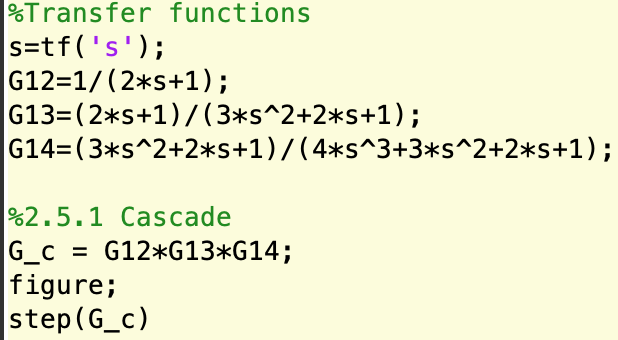
Figure 2.4.5 Plot a graph of G(s)= with its original natural frequency, double the original natural frequency, and quadruple the original natural frequency.

**2.5.1**

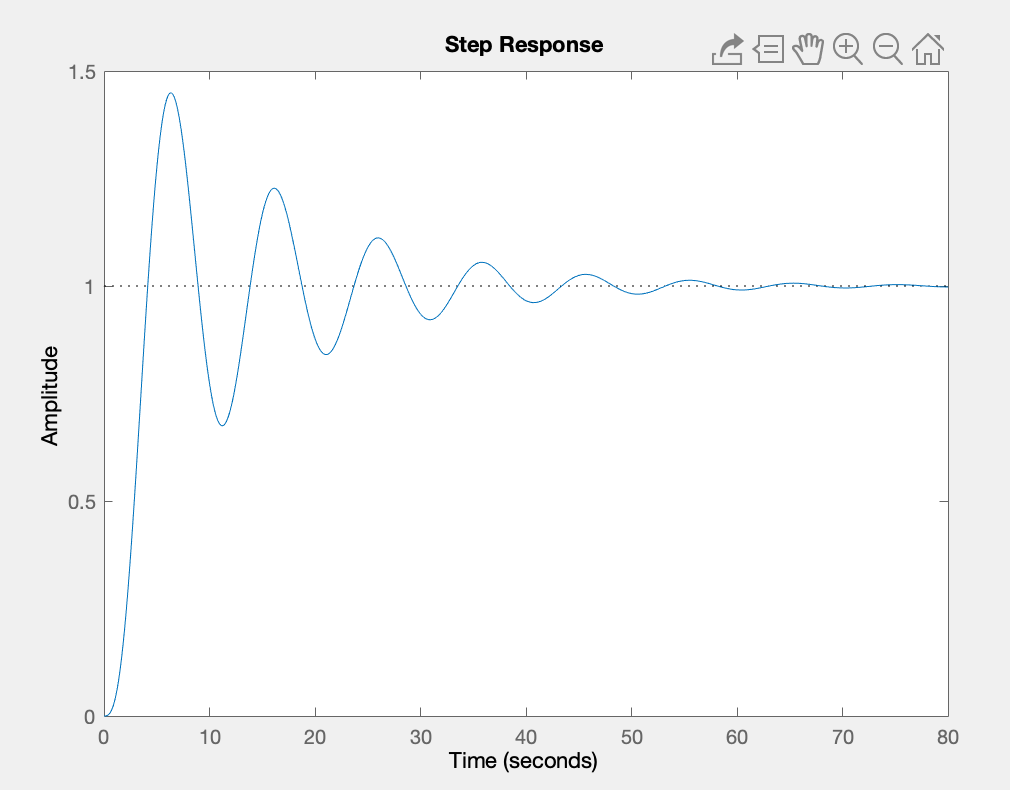
Simulink:



Code:

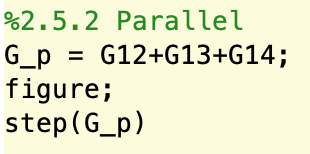
****

Output:



2.5.2 Parallel

Code:

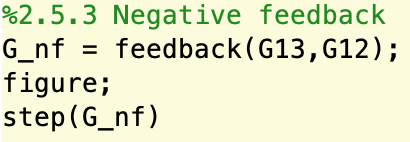


Output:



2.5.3 Negative feedback

Code:

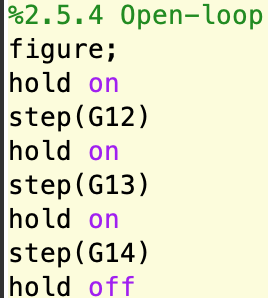


Output:

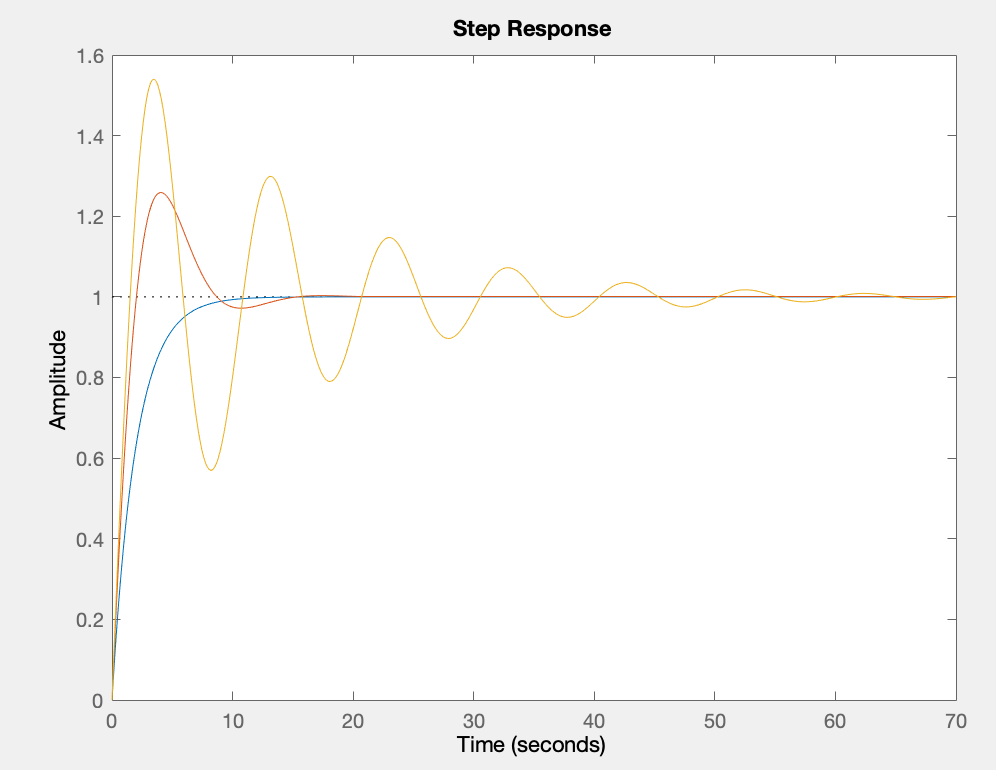


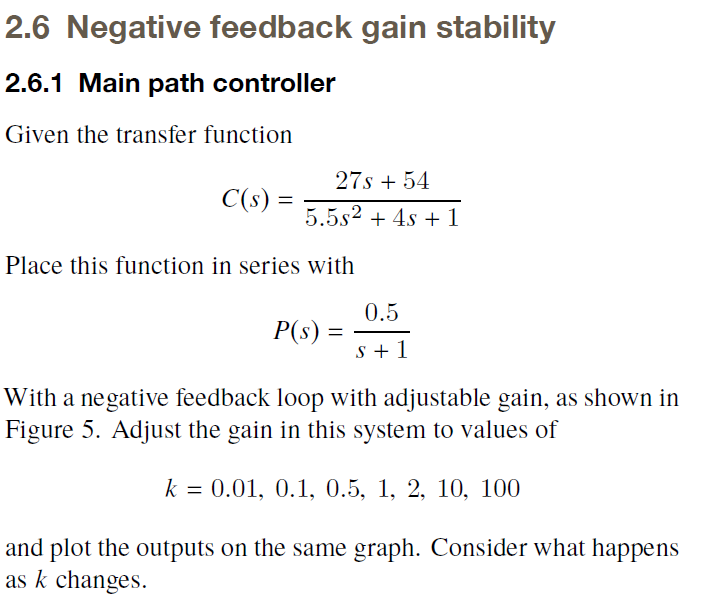
2.5.4 Open-loop

Code:

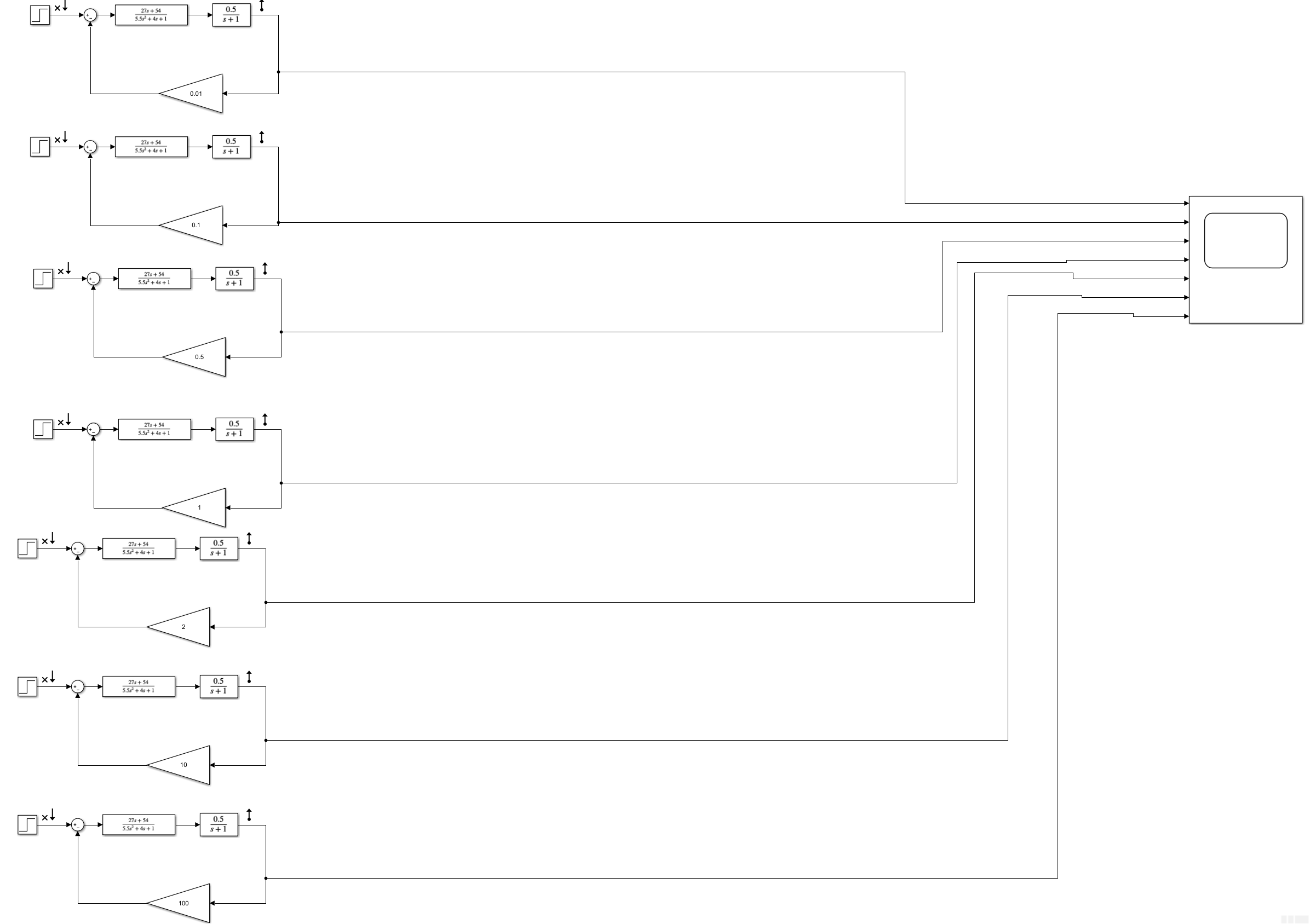


Output:

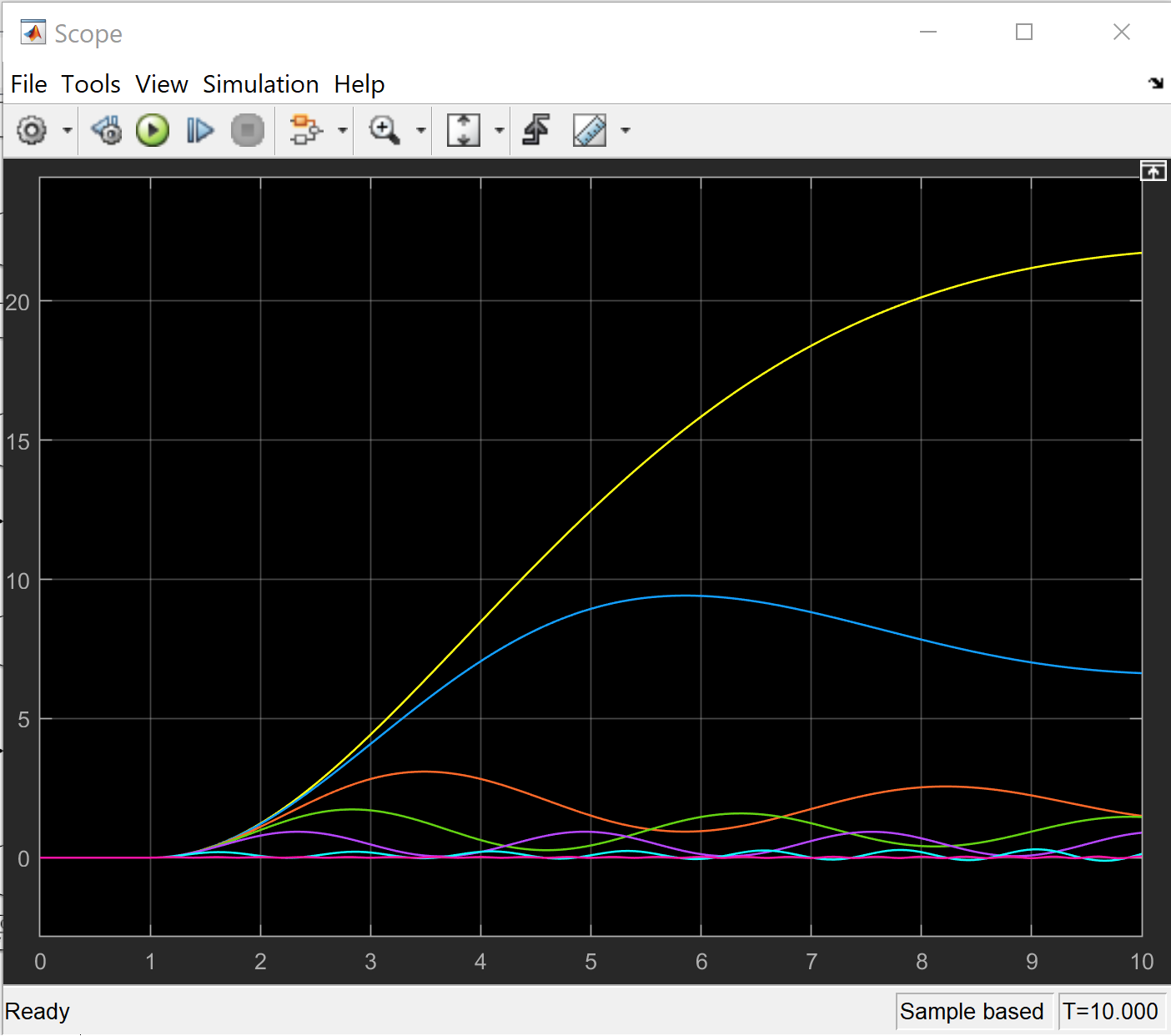
****

****

***2.6 prompt***

****

***Simulink design for multiple outputs part 2.6.1***

****

***Superimposed output for 2.6.1 displaying the effects of altering gain in the closed loop system.***

**Conclusion:**

As stated initially in the abstract portion of this lab report, the Purpose of Lab 3 was to familiarize students with MATLAB and Simulink’s capabilities in regards to modeling transfer functions and closed loop systems. MATLAB’s capabilities were demonstrated through the following applications : determining roots of polynomials, determining polynomial coefficients, converting between expanded form and zero-pole-gain (zpk) form and handling transfer function operations, working with partial fraction decomposition and inverse Laplace operations as well as plotting these types of functions. In regards to Simulink, the following functionalities were showcased: plotting with simulink, Simulink’s Linear System Analyzer which was used to plot plot and analyze various parameters such as impulse response, rise time, percent overshoot, etc., and block routing with closed loop systems containing negative feedback components.   
Overall, This was a very informative lab and really tested student skill sets in regards to analyzing control systems through MATLAB’s IDE and Sumulink’s simulation environment. After completing this lab, it can be said that a more in depth understanding has been obtained in regards to the current control system’s content presented and analyzed in this report. There is no need to go babble on. The report is 30 pages long. The results speak for themselves. Hopefully we have displayed our competence.