# CMPE 323: Signals and Systems

## Dr. LaBerge

# Lab 05 Report

## Laplace Transforms, Roots, Poles, Region of Convergence

Sabbir Ahmed

# Table of Content

## 1. Introduction

This lab provides the opportunity to manipulate Laplace Transforms using both functions embedded in MATLAB and new functions that you will create in MATLAB.

## 2. Equipment

A computer with MATLAB installed.

## 3. Procedure

### 3.1 Determining Poles and Zeros

Consider the causal, stable, LTI system with a Laplace Transform described by the coefficients:

$bs = [1\ 2\ 5]$

$as = [1.0000\ 7.0000\ 14.8125\ 12.8750\ 6.5000]$

where the coefficients are in descending orders of the Laplace variable, $s$, with $s_o$ is the constant term on the right. Write the transfer function, $H(s)$ as the ratio of polynomials in $s$. Use the MATLAB function $roots(x)$ to determine the zeros and poles of $H(s)$.

### 3.2 Creating a Pole Zero Plot

Create a MATLAB function to plot the poles, zeros and region of convergence (ROC) for the system described by the arrays given in 3.1.

### 3.3 Partial Fraction Expansion

Use the MATLAB function $residue(b, a)$ to perform the partial fraction expansions. Use those coefficients to compute and plot each of the components of the total impulse response, $h(t)$ of the system from Part 3.1 as a function of time.

### 3.4 Additional Practice

Consider the causal LTI system described by the differential equation

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + 5y = 2x$$

where $x$ and $y$ are the input the output, respectively. Create the pole zero plot. Compute and plot each of the components of the impulse response, and then the total impulse response of this system. Describe why the ROC looks the way it does.

## 4. Results

The arrays of coefficients, $bs = [1\ 2\ 5]$ and $as = [1.0000\ 7.0000\ 14.8125\ 12.8750\ 6.5000]$ were converted into its polynomial form transfer function,

$$H(s) = \frac{s^2 + 2s + 5}{s^4 + 7s^3 + 14.1825s^2 + 12.875s + 6.5}$$

The MATLAB function $roots(x)$ was used on the numerator coefficient array, $bs$, to find the zeros and on the denominator coefficient array, $as$, to find the poles.

```
% zeros_ =
%   -1.0000 + 2.0000i
%   -1.0000 - 2.0000i
% poles_ =
%   -4.0000 + 0.0000i
%   -2.0000 + 0.0000i
%   -0.5000 + 0.7500i
%   -0.5000 - 0.7500i
```

The zeros of the transfer function were a complex conjugate pair, and the poles consisted of 2 real numbers with a complex conjugate pair.

The $polezero\_plot(b,\ a,\ s\_or\_z,\ ROC)$ function was implemented on MATLAB (see Appendix A). Simple plotting methods were used for the function – the $area()$ method was used to shade in the region of convergence with additional styles and its alpha set to 0.5 for transparency. Several conditions were checked to verify the function yields accurate plots, including limits for the real axis if any of the bounds were infinite.

The function was verified by the system from Part 3.1,

`polezero_plot(b,a,'s',[-1 inf])` which generated the following pole zero plot:

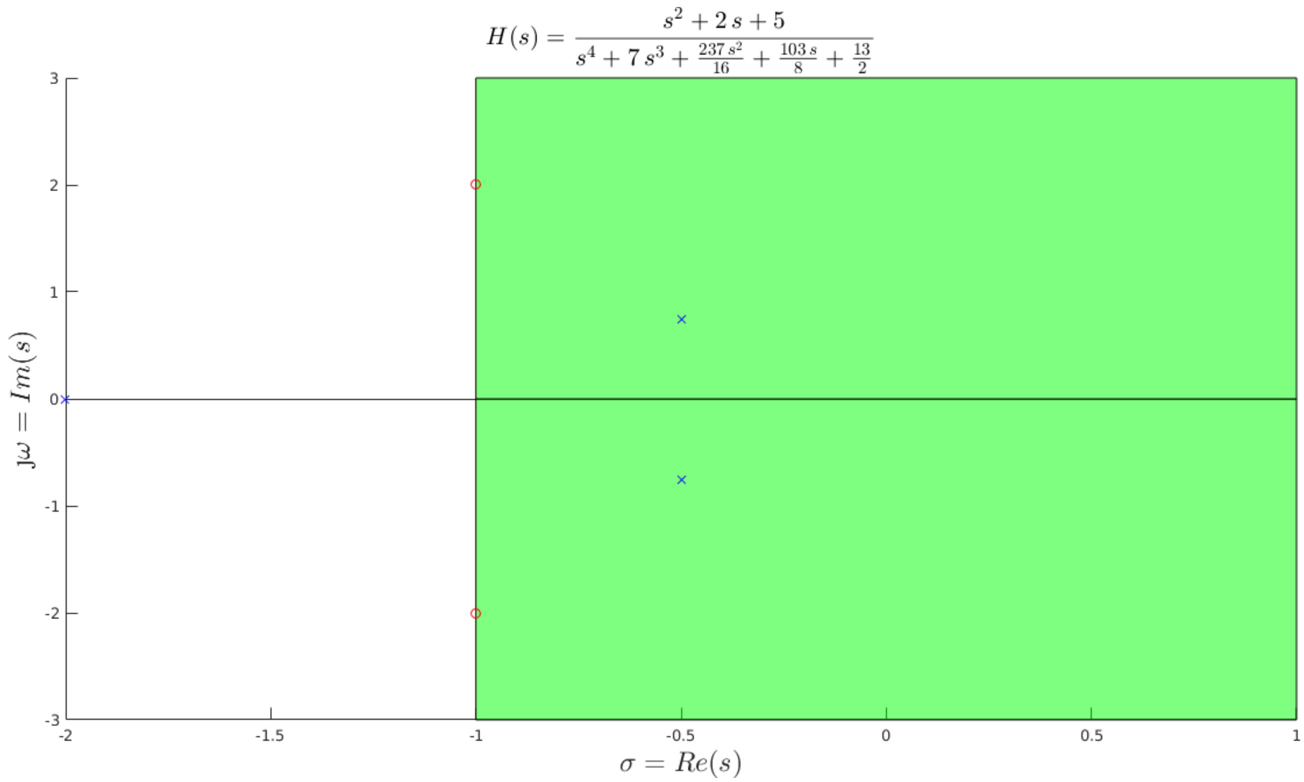


**Figure 1: Pole-Zero Plot of $H(s)$**

The MATLAB function $residue(b, a)$ was used with the same system from Part 3.1. The method returned an empty array for the $k$ for impulse response, which eliminated the requirements of implementing a unit delta function. 4 roots and poles were also computed, with 2 of them being real values and the other 2 were complex conjugate pairs of each other. The coefficients are as follows:

```
% r =

%   -0.5073 + 0.0000i

%    0.8889 + 0.0000i

%   -0.1908 - 0.3718i

%   -0.1908 + 0.3718i

% p =

%   -4.0000 + 0.0000i

%   -2.0000 + 0.0000i

%   -0.5000 + 0.7500i

%   -0.5000 - 0.7500i

% k =

%       []
```

The real fractions were converted from their original forms

$$\frac{r}{s - p}$$

to $re^{pt}u(t)$

and the complex conjugate pairs to $e^{\sigma t}\big(Ecos(\omega t) + Fsin(\omega t)\big)u(t)$.

The individual components of the impulse response were computed and plotted, then added up to form the total impulse response.

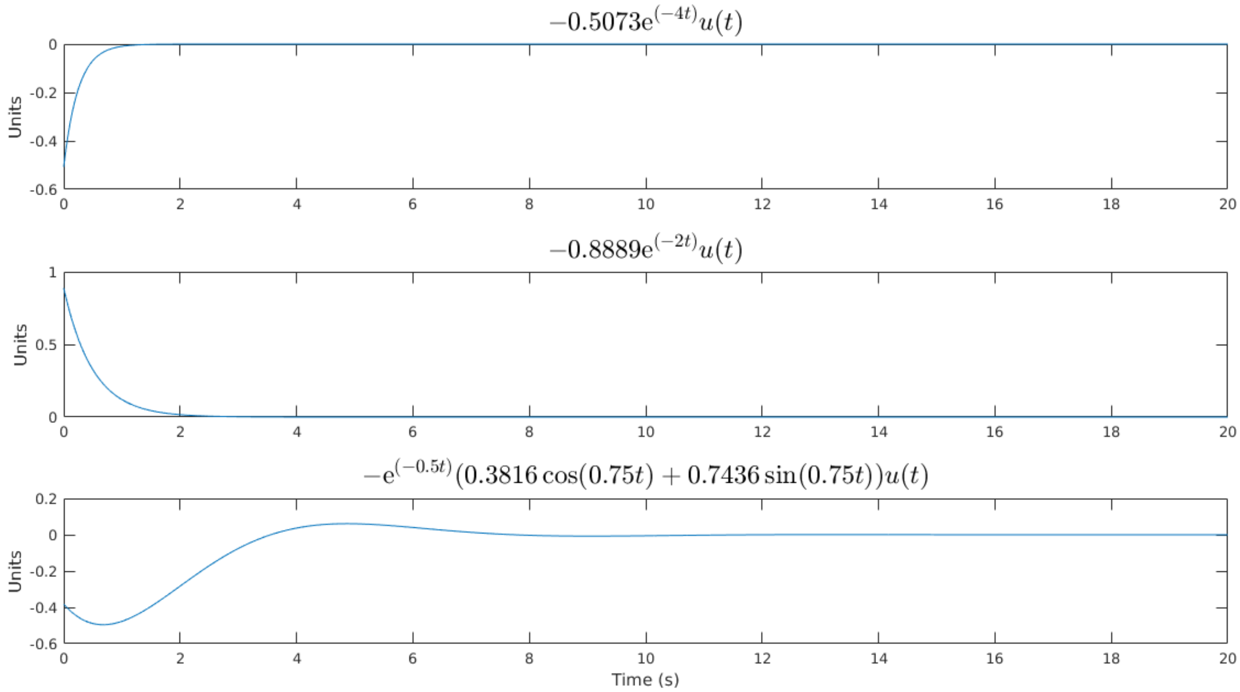


**Figure 2: Components of the Impulse Response $h(t)$ of $H(s)$**

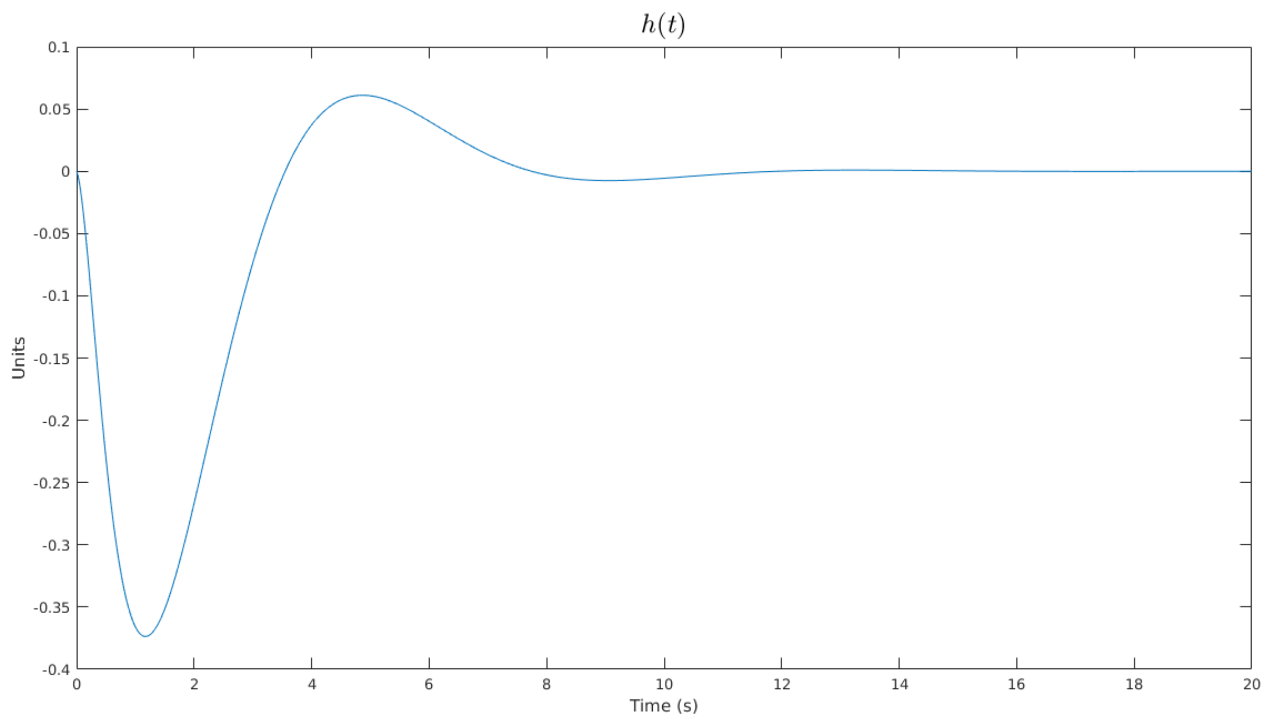


**Figure 3: The Impulse Response $h(t)$ of $H(s)$**

A second pair of coefficient arrays were generated from the differential equation, $bs = [2]$ and $as = [1\ 4\ 5]$. The arrays were passed in as parameters into the pole zero plot function, `polezero_plot(b,a,'s',[-2 inf])`. The ROC was determined from the maximum of the real parts of the poles to infinity since it was specified as a causal LTI system.

```
% zeros_ =

%    Empty matrix: 0-by-1

% poles_ =

%   -2.0000 + 1.0000i

%   -2.0000 - 1.0000i
```

The following pole zero plot was generated:

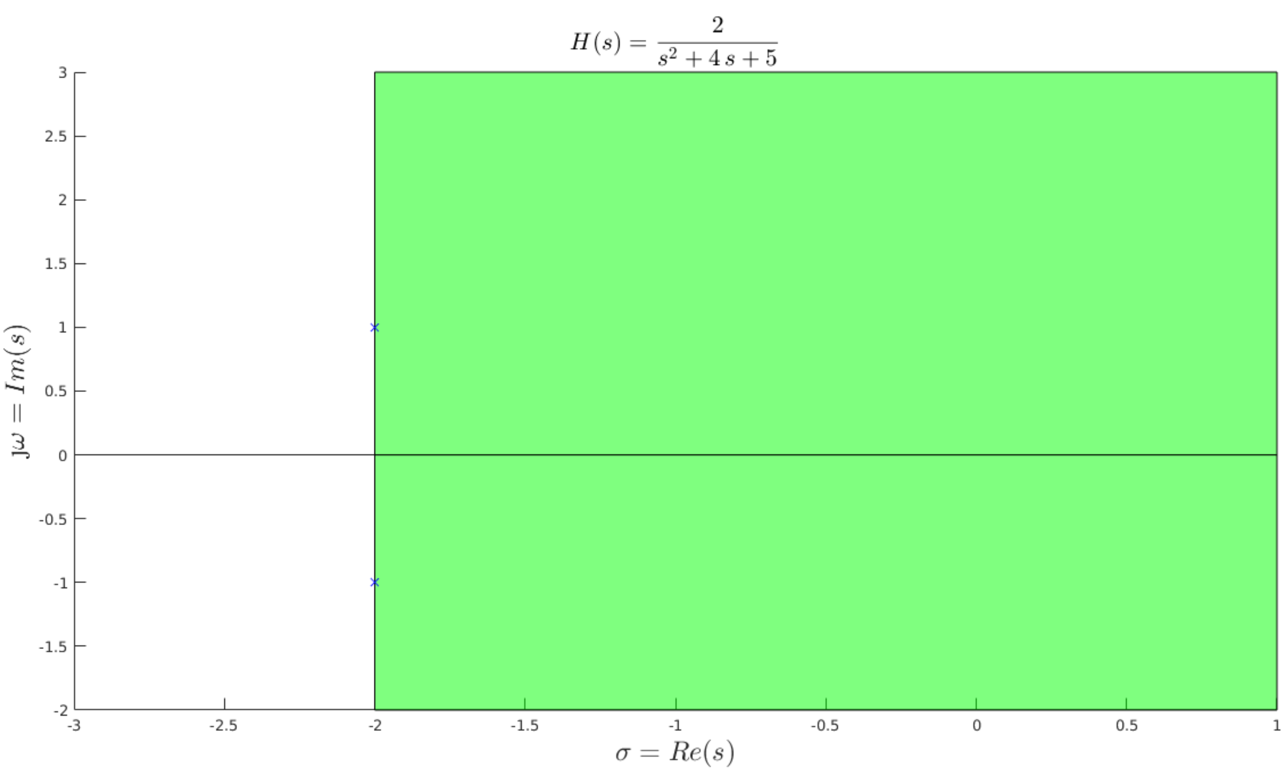


**Figure 4: Pole Zero Plot of the Second Transfer Function**

The $residue(b,\ a)$ method was used to yield the following:

```
% r =
%    0.0000 - 1.0000i
%    0.0000 + 1.0000i
% p =
%   -2.0000 + 1.0000i
%   -2.0000 - 1.0000i
% k =
%      []
```

The output suggests only one pair of complex conjugate pair as the partial fraction expansion. Therefore, there was only one component of the impulse response, plotted below:

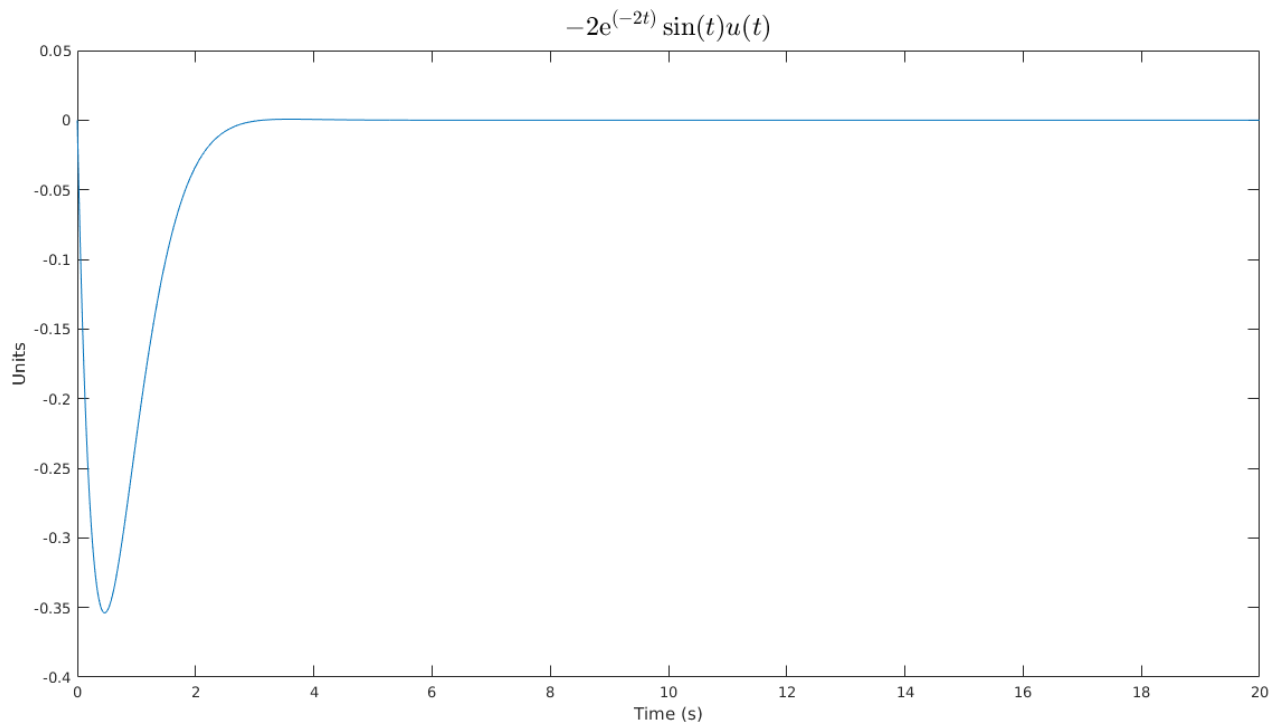


**Figure 5: The Impulse Response of the Second Transfer Function**

## 5. Appendices

### 5.1 Appendix A

Please refer to the attached script as Appendix A. The definition and documentation of the function polezero_plot() that was used for the entire lab is located there.