EE213 Lab 3: Laplace Transforms with MATLAB

2014-02-04

Contents

1 Objective 1

2 Numeric commands 1
  2.1 Transfer function representations 1
  2.2 Pole-zero representation 2
  2.3 Exercise 2
  2.4 Partial Fraction Representation 2
    2.4.1 Exercises 3

3 Time Domain Signals 3
  3.1 Inverse transforms 3
  3.2 Products of rational functions 3
    3.2.1 Exercises 4

4 Symbolic Operations 4
  4.1 Laplace transforms and inverse transforms 4
    4.1.1 Exercises 4

1 Objective

The objective of this laboratory is to introduce some MATLAB commands that are useful when treating Laplace transforms. Some of the commands are purely numeric; some operate on symbolic variables. The numeric commands are useful when all the circuit element values are specified and the only non-numeric parameter in the input and output signals is time or complex frequency; these commands calculate and/or plot the answers as functions at specific times or complex frequencies. The symbolic commands allow time, \( t \), the complex frequency \( s \), and circuit values to be undetermined variables; answers are expressed in terms of these variables.
2 Numeric commands

2.1 Transfer function representations

A general form for many Laplace transforms is as a ratio of polynomials—in other words, a rational function—in the Laplace variable \( s \):

\[
X(s) = \frac{n_k s^k + n_{k-1} s^{k-1} + n_{k-2} s^{k-2} + \cdots + n_1 s^1 + n_0 s^0}{d_j s^j + d_{j-1} s^{j-1} + d_{j-2} s^{j-2} + \cdots + d_1 s^1 + d_0 s^0}
\]  

(1)

The variable \( k \) is the degree of the numerator; \( j \) is the degree of the denominator. If \( k < j \), the rational function is called proper; otherwise, the function is improper.

In MATLAB, these commands can be represented numerically as two row vectors of coefficients of the polynomials in descending powers of \( s \):

\[
\text{num} = [n_k n_{k-1} \cdots n_0]
\]

\[
\text{den} = [d_j d_{j-1} \cdots d_0]
\]

In many MATLAB commands, such as \texttt{zp2tf} or \texttt{tf2zp}, the letters \texttt{tf} are used as a memory aid (and data type) for this representation.

2.2 Pole-zero representation

The pole-zero form of a Laplace transform is essentially a factored form of a rational function:

\[
X(s) = g \frac{(s - z_k)(s - z_{k-1})(s - z_{k-2})\cdots(s - z_2)(s - z_1)}{(s - p_j)(s - p_{j-1})(s - p_{j-2})\cdots(s - p_2)(s - p_1)}
\]  

(2)

The roots \( z_i \) of the numerator are referred to as the zeros of the function; the roots of the denominator \( p_m \) are referred to as the poles. The ratio of the two highest coefficients is the gain, \( g = \frac{n_k}{d_j} \).

In MATLAB, the pole-zero form is represented by two column vectors \( \text{z} \) and \( \text{p} \) for the zeros and poles similarly to the transfer function form. Additionally, a scalar \( \text{g} \) represents the gain. Typical MATLAB commands that interact with this representation are \texttt{zp2tf}, \texttt{tf2zp}, and \texttt{pzmap}. MATLAB determines the appropriate representation in input (and sometimes output) depending on whether the input vectors are rows or columns for many related commands.

To convert from the transfer function representation to the pole-zero representation, use the MATLAB command \( [\text{z}, \text{p}, \text{g}] = \text{tf2zp}(\text{num}, \text{den}) \). To plot the zeros and poles in the complex plane, the MATLAB command is \( \text{pzmap}(\text{num}, \text{den}) \). The zeros are represented by 0s, and the poles by Xs.

2.3 Exercise

Use MATLAB to determine the poles, zeros, and gain of each of the rational functions below:

\[
F(s) = \frac{2s^4 - 6s^3 + 18s^2 + 20s + 8}{.5s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627}
\]  

(3)

\[
G(s) = \frac{2s^4 + 6s^3 + 8s^2 + 2s + 1}{s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627}
\]  

(4)

Plot the poles and zeros of each of these functions. Without calculations, characterize \( f(t) \) and \( g(t) \), the one-sided time functions that form Laplace pairs with \( F(s) \) and \( G(s) \).
Use the MATLAB command `zp2tf` to verify that MATLAB calculates the proper expanded polynomial functions for a given set of poles, zeros, and gain.

### 2.4 Partial Fraction Representation

The partial fraction representation of a rational function is:

\[
X(s) = \sum_{l=1}^{j} \frac{r_l}{s - p_l} + \sum_{l=0}^{k-j} d_l s^l
\]  

(5)

The roots of the denominator \(p_l\) are still referred to as the poles. The numerators \(r_l\) are referred to as the residues, and the polynomial \(\sum_{l=0}^{k-j} d_l s^l\) is referred to as the direct term. When the rational function is proper, the direct term is zero.

In MATLAB, two column vectors of the residues and poles, and a row vector representing the direct term constitute the residue form.

Each residue and pole in the column vectors correspond; the direct term polynomial row vector is given in descending order.

To convert from a rational polynomial form to a residue form, use \([r, p, d] = \text{residue}(\text{num}, \text{den})\).

#### 2.4.1 Exercises

Use MATLAB to determine the residue representation of the below functions.

\[
F(s) = \frac{2s^4 - 6s^3 + 18s^2 + 20s + 8}{5s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627}
\]  

(6)

\[
G(s) = \frac{2s^4 + 6s^3 + 8s^2 + 2s + 1}{s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627}
\]  

(7)

Characterize \(f(t)u(t)\) and \(g(t)u(t)\) without performing any calculations.

The `residue()` command also converts from the residue form to the transfer function form, given the proper input, i.e. \([\text{num}, \text{den}] = \text{residue}(r, p, d)\). Use this command to verify the above residue conversion is correct.

### 3 Time Domain Signals

#### 3.1 Inverse transforms

If a Laplace-transformed signal is proper and represented as a transfer function, the inverse transform is easily plotted with the MATLAB command `impulse`; this command can also return the vectors used to plot the inverse. If the Laplace transform is not proper, impulses will be not be computed or plotted.

1. Plot: `impulse(num, den)`

2. Return vectors: `[h, t] = impulse(num, den)`

`impulse` also accepts column vectors of poles and zeros and the scalar gain.
3.2 Products of rational functions

Another operation that often comes up is the need to multiply two rational transfer functions. In other words, the products of the numerators and denominators, respectively, need to be taken. Each polynomial product can be taken using the conv command (convolution):

\[
\text{prod} = \text{conv}(\text{poly1}, \text{poly2}).
\]

If the polynomials of the rational function are represented as column vectors of poles and zeros (e.g., \(a(s), b(s)\)) and the coefficients of the highest power (e.g., \(c_a, c_b\)), then the polynomial \(c(s) = a(s)b(s)\) can be calculated in MATLAB as \(\text{poly} = [\text{poly}_a; \text{poly}_b]\) and the coefficient vector \(\text{col}_c = \text{col}_a + \text{col}_b\).

3.2.1 Exercises

Use MATLAB to plot the inverse Laplace transforms of each of the following functions.

\[
\begin{align*}
F(s) &= \frac{2s^4 - 6s^3 + 18s^2 + 20s + 8}{.5s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627} \\
G(s) &= \frac{2s^4 + 6s^3 + 8s^2 + 2s + 1}{s^7 + 5s^6 + 27s^5 + 128s^4 + 454s^3 + 1015s^2 + 1208s + 627}
\end{align*}
\]

4 Symbolic Operations

4.1 Laplace transforms and inverse transforms

Laplace transforms and their inverses may be calculated using symbolic variables \(t, s\), representing time and the Laplace variable. These variables must be declared in advance using the syms command. The inverse Laplace transform must also be multiplied by the Heaviside step function \(\text{heaviside}(t + \text{abs}(\text{eps}))\). This ensures that the transform is one-sided (why is this?).

Here is a short example snippet:

```matlab
syms a t y
f = exp(-a * t)
laplace(f,y)
ilaplace(laplace(f,y))
```

4.1.1 Exercises

Use these commands to verify all of the transform pairs given in the lab.