ECE 3413 Lab 02
Polynomials and Transfer Functions

Leomar Durán

9th February 2023

1 Introduction

The purpose of this experiment is to reinforce and build on the introduction to Matlab from Lab 01.

This lab reviews operations on polynomials and teaches how to work with transfer functions.

In this example, along with converting between the sum and factored forms of polynomials, we will also experiment with converting between various forms of the transfer function, namely ratio of polynomials, zero/pole/gain form and the time-domain form.

We will also review the circuit mesh analysis from Principles of Electric Circuits.

2 Procedure

2.1 Parts 1.1, 1.2 – Roots and polynomial forms

We revisit the use of the representation of polynomials using row vectors of coefficients from highest to lowest order, as well as the use of `poly` to convert roots to the polynomials with those roots and `roots` to convert polynomials to their roots.

2.2 Part 1.3 – Ratio of polynomials and zero/pole/gain forms

We revisit the `tf` function that creates a transfer function as a ratio of polynomials and formally introduce the `zpk` mentioned in Lab 01. This latter function creates a transfer function as a vector of zeroes, a vector of poles, and the gain.

2.2.1 Conversion between transfer function forms

In addition to creating transfer functions from vectors, `tf` and `zpk` can be used to convert back and forth between the different forms of transfer functions.

2.3 Part 1.4 – Partial fraction expansion

2.3.1 Purpose

The reason for partial fraction decomposition is preparation for the inverse Laplace transform. The inverse Laplace transform expects the function to be transformed to have a form such as the ones in the $F(s)$ column of Table 1. Partial fraction decomposition helps us meet that goal.

Table 1: Laplace transforms.

$f(t)$	$F(s)$
$t^n u(t)$	$\dfrac{n!}{s^{n+1}}$
$e^{-at}u(t)$	$\dfrac{1}{s+a}$
$\cos(\omega t)u(t)$	$\dfrac{s}{s^2+\omega^2}$
$\sin(\omega t)u(t)$	$\dfrac{\omega}{s^2+\omega^2}$

For example, let's say we have

$$G_5(s) = \frac{5(s+2)}{s(s^2+6s+34)}. \tag{1}$$

It is necessary first to perform partial fraction decomposition and rewrite it in a form

$$G_5(s) = \frac{A}{s} + \frac{Bs+C}{s^2+6s+34}, \tag{2}$$

where the denominators are each as factored out as they can be. This produces

$$G_5(s) = \frac{{}^{10}\!/_{34}}{s} + \frac{(-{}^{10}\!/_{34})s + {}^{110}\!/_{34}}{s^2+6s+34}. \tag{3}$$

What next? The denominator of the second fraction $s^2+6s+34$ should be in the form $\hat{s}^2+\omega^2$ in Table 1. (Note that this is not the same s as in $G_5(s)$, so we will refer to it as $\hat{s}$.) In other words, $s^2+6s+34$ must be a sum of two squares.

To find the first square $\hat{s}^2$, we need to complete the first square. "Completing the square" refers to the rule $(a-b)^2 = a^2-2ab+b^2$. We can find b^2 if we know a^2 and $2ab$. Well $a^2 = s^2$ and $2ab = 6s$. Thus $a = s$, and $-2ab = 2s(b) = 6s$. So $b = -{}^6\!/_2 = -3$. This makes $b^2 = (-3)^2 = 9$.

We now know that $s^2+6s+34 = (s^2+6s+9)+\omega^2 = s^2+6s+(9+\omega^2)$. Thus $9+\omega^2 = 34$. We have $\omega = \sqrt{25} = 5$.

So $s^2+6s+34 = (s^2+6s+9)+5^2$. Now we can factor $a^2-2ab+b^2$ to $(a-b)^2$. So $s^2+6s+34 = (s+3)^2+5^2$, and more specifically, we have

$$\begin{cases} \hat{s} = s+3, \\ \omega = 5. \end{cases} \tag{4}$$

Our transfer function now has the form

$$G_5(s) = \frac{{}^{10}\!/_{34}}{s} + \frac{(-{}^{10}\!/_{34})s + {}^{110}\!/_{34}}{(s+3)^2+5^2}. \tag{5}$$

The last step is splitting the second function into ${}^{\hat{s}}\!/\!_{(\hat{s}^2+\omega^2)}$ and ${}^{\omega}\!/\!_{(\hat{s}^2+\omega^2)}$ forms. Since the denominator is already in the correct form, we need to split the numerator.

$$(-{}^{10}\!/\!_{34})s + {}^{110}\!/\!_{34} = D\hat{s} + E\omega = D(s+3) + E(5). \tag{6}$$

Well, $D = -{}^{10}\!/\!_{34}$ by inspection of the s^1 coefficients, so that leaves

$$\frac{110}{34} = -\frac{10}{34}(3) + E(5) \tag{7}$$

Thus $E = {}^{28}\!/\!_{34}$.

Finally, we have

$$G_5(s) = \frac{{}^{10}\!/\!_{34}}{s} + \frac{(-{}^{10}\!/\!_{34})(s+3)}{(s+3)^2+5^2} + \frac{({}^{28}\!/\!_{34})3}{(s+3)^2+5^2} \tag{8}$$

which is ready for inverse Laplace transformation. The coefficients may be extracted from the numerators giving

$$G_5(s) = \left(\frac{10}{34}\right)\frac{1}{s} - \left(\frac{10}{34}\right)\frac{(s+3)}{(s+3)^2+5^2} + \left(\frac{28}{34}\right)\frac{3}{(s+3)^2+5^2}. \tag{9}$$

2.3.2 Storing transfer functions

To store transfer functions, I chose to store them each as a numerator and denominator, storing 2 factors, each potentially as trinomials. The factors are then multiplied.

I could have used the ZPK model, but this would have required a separate vector for the gains.

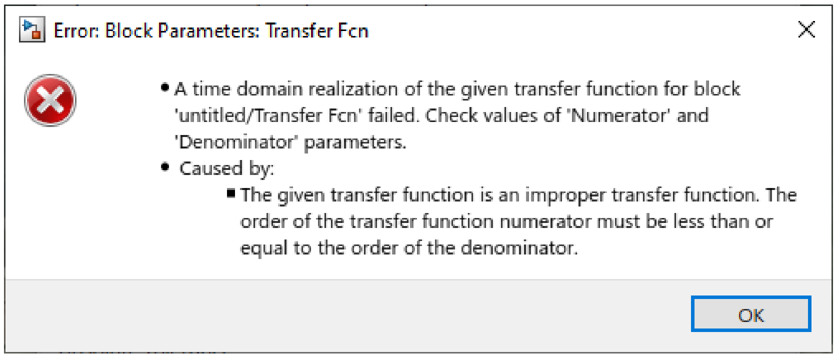
Now when a polynomial with $S+1$ terms is multiplied with a polynomial with $T+1$ terms, the resulting vector will be of length $S+T+1$,

Now if we multiply together 2 polynomials both with $S+1$ terms, we will have a product polynomial with $2S+1$ terms, and 3 polynomials this is the same as a polynomial with $2S+1$ terms and a polynomial of $S+1$ terms, which will give $(2S)+(S)+1 = 3S+1$ terms. Thus, multiplying $P+1$ polynomials with $S+1$ terms will produce a polynomial with $(P+1)S+1$ terms.

Let $Q := P+1$ and $T = S+1$, then multiplying Q polynomials with T terms each will produce a polynomial of $Q(T-1)+1$ terms. Thus, we can store the numerator and denominator of each polynomial in $2(3-1)+1 = 5$ terms.

2.3.3 Partial fraction decomposition in Matlab

Matlab provides the `residue` function which accepts a fraction as a vector of coefficients for numerator and another for denominator, returning a vector of its partial fraction numerators, the corresponding poles of the fractions and the direct function, which is the quotient polynomial resulting by dividing the numerator of the original fraction by its denominator. In the case of a proper rational expression there is no direct function. Now Simulink does not allow you to realize an improper rational expression as a transfer function as seen in Fig 1. So I believe this means that it would not be a legal transfer function.


Error: Block Parameters: Transfer Fcn

- A time domain realization of the given transfer function for block 'untitled/Transfer Fcn' failed. Check values of 'Numerator' and 'Denominator' parameters.
- Caused by:
  - The given transfer function is an improper transfer function. The order of the transfer function numerator must be less than or equal to the order of the denominator.

Figure 1: Error from attempting to realize an improper rational expression as a transfer function.

The `residue` function has its disadvantages. First, the function does not differentiate between repeated roots, at least not within its documentation. The order of the roots is not documented, but from the results, it seems to be

1. first by real part from lowest to highest;
2. then by imaginary part from highest to lowest;
3. then in the case of repeat roots, by order of the polynomial giving that root (*i.e.*, $s+1$ before $(s+1)^2$).

Since this is not documented, it may be subject to change, but for now it works. I used this knowledge to convert the partial fraction decompositions from vectors to symbolic objects for display in a Matlab Live Script.

The second disadvantage is that algorithm factors out the denominator down to the monomial, even when this will result in a complex root. This is by design, and it will likely be more useful in this course, but the expected behavior is not to factor out polynomials that will have complex roots. That is, the coefficients and constant term of a polynomial are usually expected to be real numbers (with 1 being the coefficient of the highest order term). Since this is the default behavior in Matlab, I have left it as is.

2.4 Part 2 – Laplace transforms and inverse Laplace transforms

The Laplace transform is used to convert a fraction from time-domain to frequency domain.

Both the Laplace transform function `laplace` and the inverse Laplace transform function `ilaplace` in Matlab are very straightforward and give you almost what you would expect.

The only difference is that Matlab does not represent time-domain form by having the Heaviside step function $u(t)$ as a factor.

Additionally, I chose to add a section to part 2.1 that performs the inverse Laplace transform on transfer functions in zero/pole/gain form for consistency with the other ways that transfer functions are handled.

2.5 Part 3 – Review of solving circuit loops

We are given the circuit in Fig. 2.

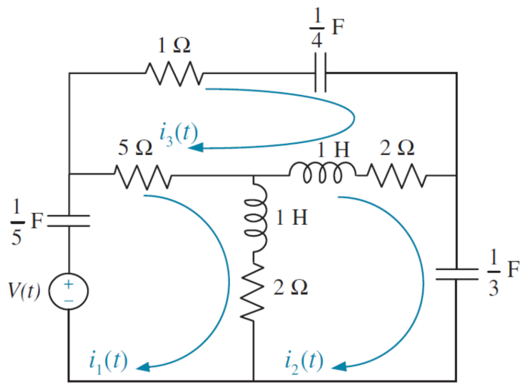


Figure 2: We must find the current loops in this circuit.

We start by labeling the polarities and components as in Fig. 3. As with the directions in finding the reaction forces in a mechanical problem, these polarities are guesses and may be updated later. As a convention, we label the first terminal met by the current as positive and the second terminal as negative. There is no convention for labeling the components. It is just a way to differentiate them.

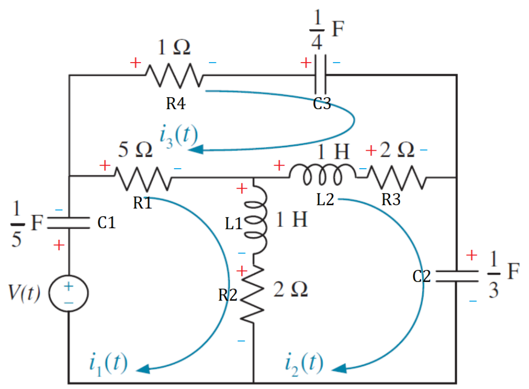


Figure 3: The circuit with polarities and components labeled.

Let's start with loop $i_1(t)$. We apply Kirchhoff's voltage law, which states that the sum of all currents around a loop are zero. We use the first terminal's polarity with the current as the sign of the voltage, starting with the voltage source $V(t)$. (That is, voltage rises are noted as negative, and voltage drops are noted as positive.)

$$\text{KVL}: \quad 0 = \sum V = -V(t) + V_{C1} + V_{R1} + V_{L1} + V_{R2} \tag{10}$$

Next we rewrite these voltages in terms of currents, except for the voltage source, according to 11. Remember that the inductor L_1 and resistor R_2 on the right-hand side of the loop are also in loop $i_2(t)$, and resistor R_1 at the top of the loop is also in loop $i_3(t)$. Notice that when two voltages move in the same direction through the component, we add them. When they clash, we subtract the other current from the one with which we are working.

$$\begin{cases} V_R = IR, \\ v_L = LD_t i, \\ v_C = \dfrac{1}{C}\displaystyle\int_0^t i(\tau)d\tau. \end{cases} \tag{11}$$

$$0 = -V(t) + \frac{1}{C_1}\int_0^t i_1(\tau)d\tau + R_1(i_1 - i_3)(t) + (L_1 D_t + R_2)(i_1 - i_2)(t). \quad (12)$$

Finally, we may take the Laplace transforms, which give us

$$0 = -\mathscr{L}[V(t)](s) + \frac{1}{sC_1}I_1(s) + R_1(I_1 - I_3)(s) + (sL_1 + R_2)(I_1 - I_2)(s). \quad (13)$$

We group together like current terms

$$\mathscr{L}[V(t)](s) = \left(\frac{1}{sC_1} + R_1 + sL_1 + R_2\right)I_1(s) + (-sL_1 - R_2)I_2(s) - R_1 I_3(s). \quad (14)$$

We do the same for loop $i_2(t)$, first finding the sum of the voltage across components

$$0 = \frac{1}{sC_2}I_2(s) + (R_2 + sL_1)(-I_2 + I_1)(s) + (sL_2 + R_3)(I_2 - I_3)(s), \quad (15)$$

then grouping like current terms

$$0 = (R_2 + sL_1)I_1(s) + \left(\frac{1}{sC_2} - R_2 - sL_1 + sL_2 + R_3\right)I_2(s) + (-sL_2 - R_3)I_3(s). \quad (16)$$

We repeat for loop $i_3(t)$.

$$0 = \left(R_4 + \frac{1}{sC_3}\right) I_3(s) + (R_3 + sL_2)(-I_3 + I_2)(s) + R_1(-I_3 + I_1)(s), \quad (17)$$

then

$$0 = R_1 I_1(s) + (R_3 + sL_2) I_2(s) + \left(R_4 + \frac{1}{sC_3} - R_3 + sL_2 - R_1\right) I_3(s). \quad (18)$$

Thus we have the system of equations.

$$\begin{bmatrix} \left(\frac{1}{sC_1} + R_1 + sL_1 + R_2\right) & (-sL_1 - R_2) & R_1 \\ (R_2 + sL_1) & \left(\frac{1}{sC_2} - R_2 - sL_1 + sL_2 + R_3\right) & (-sL_2 - R_3) \\ R_1 & (R_3 + sL_2) & \left(R_4 + \frac{1}{sC_3} - R_3 + sL_2 - R_1\right) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} (s)$$
$$= \begin{bmatrix} \mathscr{L}[V(t)](s) \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

We use Matlab to solve the system of equations in Appendix C.

3 Results

Part 1 – Poles and zeros

1ab. Roots

Calculate the roots of each of the following polynomials

$$P_1 = s^6 + s^5 + 2\,s^4 + 8\,s^3 + 7\,s^2 + 15\,s + 12$$

$$P_2 = s^6 + s^5 + 4\,s^4 + 3\,s^3 + 7\,s^2 + 15\,s + 18$$

The roots for P_1

	CartesianForm	r	thetaDeg
1	0.9979 + 1.6070i	1.8916	58.1624
2	0.9979 - 1.6070i	1.8916	301.8376
3	-1.8615 + 0.0000i	1.8615	180
4	-0.1302 + 1.4299i	1.4358	95.2009
5	-0.1302 - 1.4299i	1.4358	264.7991
6	-0.8739 + 0.0000i	0.8739	180

The roots for P_2

	CartesianForm	r	thetaDeg
1	1.0375 + 1.3227i	1.6811	51.8922
2	1.0375 - 1.3227i	1.6811	308.1078
3	-0.4632 + 1.9333i	1.9880	103.4723
4	-0.4632 - 1.9333i	1.9880	256.5277
5	-1.0743 + 0.6764i	1.2695	147.8047
6	-1.0743 - 0.6764i	1.2695	212.1953

2. Polynomial form

Calculate the polynomial form and roots of

$$P_3 = (s-1)\,(s-2)\,(s+2)\,(s+3)\,(s+4)\,(s+5)$$

The polynomial form is

```
P3_s =
```

$$s^6 + 11\,s^5 + 31\,s^4 - 31\,s^3 - 200\,s^2 - 52\,s + 240$$

The roots of the polynomial are

```
P3_roots = 6x1
   -5.0000
   -4.0000
   -3.0000
   -2.0000
    2.0000
    1.0000
```

3a. Converting to polynomial numerator and denominator.

Represent

```
G1 =

  9 (s+2) (s+3) (s+8) (s-6)
  --------------------------
  s (s+7) (s+10) (s-3) (s-2)

Continuous-time zero/pole/gain model.
```

using polynomials in the numerator and denominator.

In polynomial numerator and denominator, the transfer function

```
G1_tf =

  9 s^4 + 63 s^3 - 288 s^2 - 2052 s - 2592
  ----------------------------------------
   s^5 + 12 s^4 - 9 s^3 - 248 s^2 + 420 s

Continuous-time transfer function.
```

3b. Converting to zero-pole-gain form.

Represent

```
G2 =

         s^4 + 17 s^3 + 99 s^2 + 223 s + 140
  -------------------------------------------------
  s^5 + 32 s^4 + 363 s^3 + 2092 s^2 + 5052 s + 4320

Continuous-time transfer function.
```

using factored forms of the polynomials in the numerator and denominator.

In zero-pole-gain form, the transfer function

```
G2_zpk =

                 (s+7) (s+5) (s+4) (s+1)
  ------------------------------------------------------
  (s+16.79) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.
```

4abc. Partial fraction expansion

Calculate the partial fraction expansion of each of the following transfer functions.

$$G_3 = \frac{5(s+2)}{s(s^2+8s+15)},$$

$$G_4 = \frac{5(s+2)}{s(s^2+6s+9)},$$

$$G_5 = \frac{5(s+2)}{s(s^2+6s+34)},$$

which have the zero-pole-gain forms

```
G3 =

      5 (s+2)
  -------------
  s (s+5) (s+3)

Continuous-time zero/pole/gain model.


G4 =

    5 (s+2)
  ---------
  s (s+3)^2

Continuous-time zero/pole/gain model.


G5 =

        5 (s+2)
  -----------------
  s (s^2 + 6s + 34)

Continuous-time zero/pole/gain model.
```

The partial fraction expansions are

Well, we see that each of their residues (column #1), poles (column #2 in RP matrix), and their direct functions (K)

```
G3_RP = 3x2
   -1.5000   -5.0000
    0.8333   -3.0000
    0.6667         0
```

```
G3_K =

  0x1 empty double column vector


G4_RP = 3x2
   -1.1111   -3.0000
    1.6667   -3.0000
    1.1111         0



G4_K =

  0x1 empty double column vector


G5_RP = 3x2 complex
  -0.1471 - 0.4118i  -3.0000 + 5.0000i
  -0.1471 + 0.4118i  -3.0000 - 5.0000i
   0.2941 + 0.0000i   0.0000 + 0.0000i



G5_K =

  0x1 empty double column vector
```

Thus the partial fraction expansions

G3_partial =

$$\frac{5}{6\,(s+3)} - \frac{3}{2\,(s+5)} + \frac{2}{3\,s}$$

G4_partial =

$$\frac{5}{3\,(s+3)^2} - \frac{10}{9\,(s+3)} + \frac{10}{9\,s}$$

```
G5_partial =
```

$$\frac{5}{17\,s}+\frac{-\frac{5}{34}-\frac{7}{17}\,\mathrm{i}}{s+3-5\,\mathrm{i}}+\frac{-\frac{5}{34}+\frac{7}{17}\,\mathrm{i}}{s+3+5\,\mathrm{i}}$$

Part 1 – Poles and zeros

1ab. Roots

Calculate the roots of each of the following polynomials

$$P_1 = s^6 + s^5 + 2\,s^4 + 8\,s^3 + 7\,s^2 + 15\,s + 12$$

$$P_2 = s^6 + s^5 + 4\,s^4 + 3\,s^3 + 7\,s^2 + 15\,s + 18$$

The roots for P_1

	CartesianForm	r	thetaDeg
1	0.9979 + 1.6070i	1.8916	58.1624
2	0.9979 - 1.6070i	1.8916	301.8376
3	-1.8615 + 0.0000i	1.8615	180
4	-0.1302 + 1.4299i	1.4358	95.2009
5	-0.1302 - 1.4299i	1.4358	264.7991
6	-0.8739 + 0.0000i	0.8739	180

The roots for P_2

	CartesianForm	r	thetaDeg
1	1.0375 + 1.3227i	1.6811	51.8922
2	1.0375 - 1.3227i	1.6811	308.1078
3	-0.4632 + 1.9333i	1.9880	103.4723
4	-0.4632 - 1.9333i	1.9880	256.5277
5	-1.0743 + 0.6764i	1.2695	147.8047
6	-1.0743 - 0.6764i	1.2695	212.1953

2. Polynomial form

Calculate the polynomial form and roots of

$$P_3 = (s-1)\,(s-2)\,(s+2)\,(s+3)\,(s+4)\,(s+5)$$

The polynomial form is

```
P3_s =
```

$$s^6 + 11\,s^5 + 31\,s^4 - 31\,s^3 - 200\,s^2 - 52\,s + 240$$

The roots of the polynomial are

```
P3_roots = 6x1
   -5.0000
   -4.0000
   -3.0000
   -2.0000
    2.0000
    1.0000
```

3a. Converting to polynomial numerator and denominator.

Represent

```
G1 =

  9 (s+2) (s+3) (s+8) (s-6)
  -------------------------
  s (s+7) (s+10) (s-3) (s-2)

Continuous-time zero/pole/gain model.
```

using polynomials in the numerator and denominator.

In polynomial numerator and denominator, the transfer function

```
G1_tf =

  9 s^4 + 63 s^3 - 288 s^2 - 2052 s - 2592
  ----------------------------------------
   s^5 + 12 s^4 - 9 s^3 - 248 s^2 + 420 s

Continuous-time transfer function.
```

3b. Converting to zero-pole-gain form.

Represent

```
G2 =

         s^4 + 17 s^3 + 99 s^2 + 223 s + 140
  -------------------------------------------------
  s^5 + 32 s^4 + 363 s^3 + 2092 s^2 + 5052 s + 4320

Continuous-time transfer function.
```

using factored forms of the polynomials in the numerator and denominator.

In zero-pole-gain form, the transfer function

```
G2_zpk =

                (s+7) (s+5) (s+4) (s+1)
  -------------------------------------------------------
  (s+16.79) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)

Continuous-time zero/pole/gain model.
```

4abc. Partial fraction expansion

Calculate the partial fraction expansion of each of the following transfer functions.

$$G_3 = \frac{5(s+2)}{s(s^2+8s+15)},$$
$$G_4 = \frac{5(s+2)}{s(s^2+6s+9)},$$
$$G_5 = \frac{5(s+2)}{s(s^2+6s+34)},$$

which have the zero-pole-gain forms

```
G3 =

     5 (s+2)
  -------------
  s (s+5) (s+3)

Continuous-time zero/pole/gain model.
```

```
G4 =

   5 (s+2)
  ---------
  s (s+3)^2

Continuous-time zero/pole/gain model.


G5 =

       5 (s+2)
  ----------------
  s (s^2 + 6s + 34)

Continuous-time zero/pole/gain model.
```

The partial fraction expansions are

Well, we see that each of their residues (column #1), poles (column #2 in RP matrix), and their direct functions (K)

```
G3_RP = 3x2
   -1.5000   -5.0000
    0.8333   -3.0000
    0.6667         0



G3_K =

  0x1 empty double column vector


G4_RP = 3x2
   -1.1111   -3.0000
    1.6667   -3.0000
    1.1111         0
```

```
G4_K =

  0x1 empty double column vector


G5_RP = 3x2 complex
  -0.1471 - 0.4118i  -3.0000 + 5.0000i
  -0.1471 + 0.4118i  -3.0000 - 5.0000i
   0.2941 + 0.0000i   0.0000 + 0.0000i



G5_K =

  0x1 empty double column vector
```

Thus the partial fraction expansions

G3_partial =

$$\frac{5}{6\,(s+3)} - \frac{3}{2\,(s+5)} + \frac{2}{3\,s}$$

G4_partial =

$$\frac{5}{3\,(s+3)^2} - \frac{10}{9\,(s+3)} + \frac{10}{9\,s}$$

G5_partial =

$$\frac{5}{17\,s} + \frac{-\frac{5}{34} - \frac{7}{17}\,\mathrm{i}}{s+3-5\,\mathrm{i}} + \frac{-\frac{5}{34} + \frac{7}{17}\,\mathrm{i}}{s+3+5\,\mathrm{i}}$$

Part 3 – Review of solving circuit loops

The matrix of coefficients

```
M =
```

$$\begin{pmatrix} s+\frac{5}{s}+7 & -s-2 & 5 \\ s+2 & \frac{3}{s} & s+2 \\ 5 & s+2 & s+\frac{4}{s}-6 \end{pmatrix}$$

The expected total voltage of each loop

```
y =
```

$$\begin{pmatrix} V_s \\ 0 \\ 0 \end{pmatrix}$$

This requires that the current

```
I =
```

$$\begin{pmatrix} \frac{V_s\,s\,\left(s^4+4\,s^3+s^2+18\,s-12\right)}{\sigma_1} \\ -\frac{V_s\,s^2\,\left(-s^3+9\,s^2+18\,s-8\right)}{\sigma_1} \\ -\frac{V_s\,s^2\,\left(s^3+4\,s^2+4\,s-15\right)}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = 13\,s^5+50\,s^4+53\,s^3+178\,s^2+6\,s-60$$

4 Discussion

This lab did require me to relearn how to do circuit mesh analysis for Part 3.

The lab teaches the process from start to finish used to analyze a circuit to create equations for the current loops, then turn those equations into transfer functions in the time domain where it can be examined graphically.

Using this process to create a graphic representation of a circuit can help to study characteristics that circuits producing similar graphs can have, such as the dampening of the circuit or stability.

Appendix

Leomar Durán

9th February 2023

A Part 1 – Poles and zeroes, Matlab Live Script

Part 1 — Poles and zeros

```
clear
```

1ab. Roots

Calculate the roots of each of the following polynomials

```
% Each polynomial is represented by each row of the matrix
P12 = [ 1 1 2 8 7 15 12 ; ...
        1 1 4 3 7 15 18 ];

% the number of polynomials
nRows = size(P12, 1);

% print the polynominal forms of each in turns of s
syms P [1 2], syms s
for k=1:nRows
    % display the polynomial in an equation
    disp(P(k) == poly2sym(P12(k,:), s))
end % next k
```

$$P_1 = s^6 + s^5 + 2\,s^4 + 8\,s^3 + 7\,s^2 + 15\,s + 12$$

$$P_2 = s^6 + s^5 + 4\,s^4 + 3\,s^3 + 7\,s^2 + 15\,s + 18$$

```
% allocate cell array of roots
P_roots = cell(1, nRows);
% loop through the rows of P12
for k=1:nRows
    % calculate the roots for each polynomial
    P_roots{k} = roots(P12(k,:));
end % next k
```

The roots for P_1

```
% display the roots each in a table showing both forms (Cartesian, polar)
P1_roots_table = complexTable(P_roots{1})
```

P1_roots_table = 6×3 table

	CartesianForm	r	thetaDeg
1	0.9979 + 1.6070i	1.8916	58.1624
2	0.9979 - 1.6070i	1.8916	301.8376
3	-1.8615 + 0.0000i	1.8615	180
4	-0.1302 + 1.4299i	1.4358	95.2009
5	-0.1302 - 1.4299i	1.4358	264.7991
6	-0.8739 + 0.0000i	0.8739	180

The roots for P_2

```
P2_roots_table = complexTable(P_roots{2})
```

P2_roots_table = 6×3 table

	CartesianForm	r	thetaDeg
1	1.0375 + 1.3227i	1.6811	51.8922
2	1.0375 - 1.3227i	1.6811	308.1078
3	-0.4632 + 1.9333i	1.9880	103.4723
4	-0.4632 - 1.9333i	1.9880	256.5277
5	-1.0743 + 0.6764i	1.2695	147.8047
6	-1.0743 - 0.6764i	1.2695	212.1953

2. Polynomial form

Calculate the polynomial form and roots of

```
% the polynomial
P3_poly = poly(-[5 2 3 -1 -2 4]);

% display factored out in terms of s
syms P3 s
disp(P3 == prod(factor(poly2sym(P3_poly, s))))
```

$$P_3 = (s-1)\ (s-2)\ (s+2)\ (s+3)\ (s+4)\ (s+5)$$

The polynomial form is

```
syms s
P3_s = poly2sym(P3_poly, s)
```

P3_s = $s^6 + 11\,s^5 + 31\,s^4 - 31\,s^3 - 200\,s^2 - 52\,s + 240$

The roots of the polynomial are

```
P3_roots = roots(P3_poly)
```

```
P3_roots = 6×1
   -5.0000
   -4.0000
   -3.0000
   -2.0000
    2.0000
    1.0000
```

3a. Converting to polynomial numerator and denominator.

Represent

```
% the transfer function in zero-pole-gain form
G1 = zpk(-[2 3 -6 8], -[0 7 -2 10 -3], 9)
```

```
G1 =

  9 (s+2) (s+3) (s+8) (s-6)
  --------------------------
  s (s+7) (s+10) (s-3) (s-2)

Continuous-time zero/pole/gain model.
```

using polynomials in the numerator and denominator.

In polynomial numerator and denominator, the transfer function

```
G1_tf = tf(G1)
```

```
G1_tf =

  9 s^4 + 63 s^3 - 288 s^2 - 2052 s - 2592
  ----------------------------------------
   s^5 + 12 s^4 - 9 s^3 - 248 s^2 + 420 s

Continuous-time transfer function.
```

3b. Converting to zero-pole-gain form.

Represent

```
% the transfer function in polynomial numerator and denominator form
G2 = tf([1 17 99 223 140], [1 32 363 2092 5052 4320])
```

```
G2 =

         s^4 + 17 s^3 + 99 s^2 + 223 s + 140
  -------------------------------------------------
  s^5 + 32 s^4 + 363 s^3 + 2092 s^2 + 5052 s + 4320

Continuous-time transfer function.
```

```

% $G_2(s) = \frac{s^4 + 17s^3+ 99s^2 + 223s + 140}
%                {s^5 + 32s^4 + 363s^3 + 2092s^2 + 5052s + 4320}}$
```

using factored forms of the polynomials in the numerator and denominator.

In zero-pole-gain form, the transfer function

```
G2_zpk = zpk(G2)
```

```
G2_zpk =

               (s+7) (s+5) (s+4) (s+1)
  ----------------------------------------------------
  (s+16.79) (s^2 + 4.097s + 4.468) (s^2 + 11.12s + 57.6)
```

```
Continuous-time zero/pole/gain model.
```

4abc. Partial fraction expansion

Calculate the partial fraction expansion of each of the following transfer functions.

$$G_3 = \frac{5(s+2)}{s(s^2+8s+15)},\ G_4 = \frac{5(s+2)}{s(s^2+6s+9)},\ G_5 = \frac{5(s+2)}{s(s^2+6s+34)},$$

which have the zero-pole-gain forms

```
% allocate G polynomials
%   2 factors per line
%   3-nominal max for each factor
%   2 lines for numerator and denominator
%   5 functions (first 2 unused)
G_roots = zeros(2, 3, 2, 5);

% get the size of G
[nFactors, nRootTerms, ~, nTfs] = size(G_roots);

% G3
G_roots(:, :, :, 3) = cat(3, [0 0 5 ; 0 1 2], [0 1 0; 1 8 15]);
% G4
G_roots(:, :, :, 4) = cat(3, [0 0 5 ; 0 1 2], [0 1 0; 1 6 9]);
% G5
G_roots(:, :, :, 5) = cat(3, [0 0 5 ; 0 1 2], [0 1 0; 1 6 34]);

% convolve the factors in each line giving G_poly:
% * the resulting length of convolution for operands of length (M + 1),
% (N + 1) is (M + N + 1), so C convolutions of N-vectors each will give
% a length of C(N - 1) + 1
nPolyTerms = (((nRootTerms - 1)*nFactors) + 1);
% allocate
G_poly = zeros(nPolyTerms, 2, nTfs);
% loop through the transfer functions (skip first 2)
for (iTf=3:nTfs)
    for (iLine=1:2)
        G_poly(:, iLine, iTf) = convRows(G_roots(:, :, iLine, iTf));
    end % next iLine
end % next iTf

% print the polynominal forms of each transfer function:
% allocate room for the transfer functions
G = cell(1,nTfs);
% loop through the transfer functions
for iTf=3:nTfs
    G{iTf} = zpk(tf(G_poly(:, 1, iTf)', G_poly(:, 2, iTf)'));
end % next iTf
% print each one
G3 = G{3}
```

```
G3 =

     5 (s+2)
  -------------
  s (s+5) (s+3)

Continuous-time zero/pole/gain model.
```

```
G4 = G{4}
```

```
G4 =

   5 (s+2)
  ---------
  s (s+3)^2

Continuous-time zero/pole/gain model.
```

```
G5 = G{5}
```

```
G5 =

       5 (s+2)
  -----------------
  s (s^2 + 6s + 34)

Continuous-time zero/pole/gain model.
```

The partial fraction expansions are

```
% an (n + 1)-nomial will have n roots.
nPolesPred = (nPolyTerms - 1);
% allocate R (coefficients), P (poles), K (direct term)
RP = zeros(nPolesPred, 2, nTfs);
% all the transfer functions have degree 1-(1+2) = -2,
% so expect no direct function
K = zeros(0, nTfs);
% loop through the transfer functions
for iTf=3:nTfs
    % the #roots predicted may be more than necessary
    [G345_R, G345_P, G345_K] = ...
        residue(G_poly(:, 1, iTf), G_poly(:, 2, iTf));
    % so 0-pad R, P, K
    nPolesActual = numel(G345_P);
    G345_R = [ zeros((nPolesPred - nPolesActual), 1) ; G345_R ];
    G345_P = [ zeros((nPolesPred - nPolesActual), 1) ; G345_P ];
    % collect the residue
    RP(:, 1, iTf) = G345_R;
    RP(:, 2, iTf) = G345_P;
    K(:, iTf) = G345_K;
end % next iTf
```

Well, we see that each of their residues (column #1), poles (column #2 in RP matrix), and their direct functions (K)

```
% copy R, P, K for each function, filtering out zero rows
% (for future use)
G3_RP = RP((RP(:,1,3) ~= 0), :, 3)
```

```
G3_RP = 3×2
   -1.5000   -5.0000
    0.8333   -3.0000
    0.6667         0
```

```
G3_K = K(:, 3)
```

```
G3_K =

  0×1 empty double column vector
```

```
G4_RP = RP((RP(:,1,4) ~= 0), :, 4)
```

```
G4_RP = 3×2
   -1.1111   -3.0000
    1.6667   -3.0000
    1.1111         0
```

```
G4_K = K(:, 4)
```

```
G4_K =

  0×1 empty double column vector
```

```
G5_RP = RP((RP(:,1,5) ~= 0), :, 5)
```

```
G5_RP = 3×2 complex
  -0.1471 - 0.4118i  -3.0000 + 5.0000i
  -0.1471 + 0.4118i  -3.0000 - 5.0000i
   0.2941 + 0.0000i   0.0000 + 0.0000i
```

```
G5_K = K(:, 5)
```

```
G5_K =

  0×1 empty double column vector
```

Thus the partial fraction expansions

```
% cause syntax error if misspelled
OMITNAN = 'omitnan';

% allocate room for the transfer functions
syms G_partial [1 nTfs]
% in terms of s
syms s
% loop through the transfer functions
for iTf=3:nTfs
    % filter out zero rows
    Tf_RP = RP((RP(:,1,iTf) ~= 0), :, iTf);
    % get the poles
    Tf_poles = Tf_RP(:, 2);
    % loop through remaining rows
    nTf_RP = numel(Tf_poles);
    % initialize to 0
    G_partial(iTf) = 0;
    % loop through the residue-pole rows
    for iRP = 1:nTf_RP
        % count the number of repeats so far for this pole (including
        % this one):
        % * poles are ordered so that an increase in previous instances
        % means an increase in order of the current instance
        nInstances = (sum(Tf_poles(1:iRP) == Tf_poles(iRP)));
        % add the fraction to G_partial: R/(s - P)^n
        G_partial(iTf) = (G_partial(iTf) + (Tf_RP(iRP,1)./(s - Tf_poles(iRP))^nInstances));
    end % next iRP
    % add all of the direct terms
    G_partial(iTf) = G_partial(iTf) + sum(K(:, iTf));
end % next iTf
% print each one
G3_partial = G_partial(3)
```

```
G3_partial =
```

$$\frac{5}{6\ (s+3)}-\frac{3}{2\ (s+5)}+\frac{2}{3\,s}$$

```
G4_partial = G_partial(4)
```

```
G4_partial =
```

$$\frac{5}{3\ (s+3)^2}-\frac{10}{9\ (s+3)}+\frac{10}{9\,s}$$

```
G5_partial = G_partial(5)
```

```
G5_partial =
```

$$\frac{5}{17\,s}+\frac{-\frac{5}{34}-\frac{7}{17}\,\mathrm{i}}{s+3-5\,\mathrm{i}}+\frac{-\frac{5}{34}+\frac{7}{17}\,\mathrm{i}}{s+3+5\,\mathrm{i}}$$

```
function complexTable = complexTable(complex)
```

complexTable(complex)

Creates a table showing the Cartesian forms, magnitudes and angles (in [0, 360) [deg]) of the given complex numbers.

Input Arguments

complex : double = vector of complex numbers

Output Arguments

complexTable : table (3-columns) = the table of the Cartesian forms, magnitudes and angles (in [0, 360) [deg]) of each complex numbers

```
    CartesianForm = complex;
    r = abs(complex);
    thetaDeg = angle360(complex);
    complexTable = table(CartesianForm, r, thetaDeg);
end % function complexTable(complex)


function angle360 = angle360(vector)
```

angle360(vector)

Gets an angle from a vector of complex numbers in the domain of [0, 360) [deg].

Input Arguments

vector : double = representing the vector of complex numbers

Output Arguments

acc : the vector of arrays in the domain of [0, 360) [deg

```
    angle360 = mod(rad2deg(angle(vector)) + 360, 360);
end % function angle360(vector)


function acc = convRows(matrix)
```

conv_rows(matrix)

Convolves the rows of a matrix into a row vector.

Input Arguments

T : double = 2D array whose rose to convolve

Output Arguments

acc : the resulting convolved row vector

```
    % initialize
    acc = 1;
    % transpose to loop the matrix by row
    % because Matlab defaults to by column
    for row = matrix'
        % convolve the accumulator with the next row.
        % note that row will be vertical as a column requiring another
        % transpose
        acc = conv(acc, row');
    end % next row
end % function conv_rows(matrix)
```

B Part 2 – Laplace transforms, Matlab Live Script

Part 2 — Laplace transforms

```
clear
```

1. The Laplace transform

Find the Laplace transform of

$$f(t) = 0.0075 - 0.00034e^{-2.5t}cos(22t) + 0.087e^{-2.5t}sin(22t) - 0.0072e^{-8t}.$$

```
% make a symbol t
syms t
% the function f(t) represented by MATLAB
f_t = 0.0075 - 0.00034*exp(-2.5*t)*cos(22*t) + ...
    0.087*exp(-2.5*t)*sin(22*t) - 0.0072*exp(-8*t)
```

```
f_t =
```

$$\frac{87\sin(22\,t)\,e^{-\frac{5\,t}{2}}}{1000} - \frac{17\cos(22\,t)\,e^{-\frac{5\,t}{2}}}{50000} - \frac{9\,e^{-8\,t}}{1250} + \frac{3}{400}$$

The Laplace transform of $f(t)$,

```
% make a symbol s
syms s
% this seems to be the best we can do although it just gives the sum of
% rational expressions instead of a rational expression of polynomials
% because you cannot convert symbol to transfer function
F = laplace(f_t, s)
```

```
F =
```

$$\frac{3}{400\,s} - \frac{9}{1250\,(s+8)} - \frac{17\left(s+\frac{5}{2}\right)}{50000\left(\left(s+\frac{5}{2}\right)^2+484\right)} + \frac{957}{500\left(\left(s+\frac{5}{2}\right)^2+484\right)}$$

```
clear
```

2. The inverse of the Laplace transform

Find the inverse Laplace transform of

```
% the transfer function to find the inverse Laplace of
F = zpk(-[3 5 7], [0 8 roots([1 10 100])'], 2)
```

```
F =

     2 (s+3) (s+5) (s+7)
  -------------------------
```

```
    s (s-8) (s^2 + 10s + 100)

  Continuous-time zero/pole/gain model.
```

We can represent the transfer function in Matlab from its roots

```
% make a symbol s
syms s

% get zeros and poles
F_z = F.z{1};
F_p = F.p{1};
% zeroes and poles that are purely real
F_zr = F_z(imag(F_z) == 0);
F_pr = F_p(imag(F_p) == 0);
% zeroes and poles with negative impaginary part
F_zmi = F_z(imag(F_z) < 0);
F_pmi = F_p(imag(F_p) < 0);
% we don't need the positive imaginary parts because these are just the
% conjugates of the negative ones, and we can get them from the
% negative ones

% multiply the real zeroes/real poles (all linear polynomials)
F_linear = prod(s - F_zr)/prod(s - F_pr);
% multiply in the complex ones (all quadratic polynomials)
F_quadratic = prod(factorFromComplexRoot(F_zmi, s))/...
                    prod(factorFromComplexRoot(F_pmi, s));
% multiply linear, quadratic and gain
F_s = F_linear * F_quadratic * F.k(1)
```

```
F_s =
```

$$\frac{2\ (s+3)\ (s+5)\ (s+7)}{s\ ((s+5)^2+75)\ (s-8)}$$

Then, the inverse Laplace transform of $F(s)$,

```
% make a symbol t
syms t
f_t = ilaplace(F_s, t)
```

```
f_t =
```

$$\frac{2145\ e^{8t}}{976}+\frac{79\ e^{-5t}\ (\cos(5\ \sqrt{3}\ t)+9\ \sqrt{3}\ \sin(5\ \sqrt{3}\ t))}{1220}-\frac{21}{80}$$

```
function factor = factorFromComplexRoot(complexRoot, s)
```

factorFromComplexRoot(complexRoot, s)

Returns the polynomial factor $P(s)$ that when fixed to 0, gives the given **complexRoot** for polynomial variable **s**.

Input Arguments

complexRoot : double = root for which to find the polynomial

Output Arguments

factor : the factor polynomial $P(s)$ that equals 0 at **s == complexRoot**

```
    factor = ((s - real(complexRoot)).^2 + imag(complexRoot).^2);
end % function factorFromComplexRoot(complexRoot, s)
```

C Part 3 – Review of solving circuit loops, Matlab Live Script

Part 3 — Review of solving circuit loops

```
clear

% Find the loop current given the loop circuit in subsection 2.5 in the
% lab report.

% symbolic variables
syms s
syms V_s

% define the parameters

% resistors
R1 = 5; % [ohm]
R2 = 2;
R3 = 2;
R4 = 1;

% inductors
L1 = 1; % [H]
L2 = 1;

% currents
C1 = 1/5; % [F]
C2 = 1/3;
C3 = 1/4;

% the columns, representing the coefficients for each current
M_I1 = [ (1/(s*C1) + R1 + s*L1 + R2); (R2 + s*L1); R1 ];
M_I2 = [ (-s*L1 - R2); (1/(s*C2) - R2 - s*L1 + s*L2 + R3); (R3 + s*L2) ];
M_I3 = [ R1; (R3 + s*L2); (R4 + 1/(s*C3) - R3 + s*L2 - R1) ];
```

The matrix of coefficients

```
M = [ M_I1, M_I2, M_I3 ]
```

M =

$$\begin{pmatrix} s+\frac{5}{s}+7 & -s-2 & 5 \\ s+2 & \frac{3}{s} & s+2 \\ 5 & s+2 & s+\frac{4}{s}-6 \end{pmatrix}$$

The expected total voltage of each loop

```
y = [ V_s ; 0 ; 0 ]
```

y =

$$\begin{pmatrix} V_s \\ 0 \\ 0 \end{pmatrix}$$

This requires that the current

```
% Solve for I
I = M^-1*y
```

```
I =
```

$$\begin{pmatrix} \frac{V_s\, s\ (s^4 + 4\, s^3 + s^2 + 18\, s - 12)}{\sigma_1} \\ -\frac{V_s\, s^2\ (-s^3 + 9\, s^2 + 18\, s - 8)}{\sigma_1} \\ -\frac{V_s\, s^2\ (s^3 + 4\, s^2 + 4\, s - 15)}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = 13\, s^5 + 50\, s^4 + 53\, s^3 + 178\, s^2 + 6\, s - 60$$