Christopher Lum
lum@uw.edu

# Lecture 02d
# Using ‘minreal’ in Matlab to Perform Transfer Function Pole/Zero Cancellation


The YouTube video entitled ‘Using ‘minreal’ in Matlab to Perform Transfer Function Pole/Zero Cancellation’ that covers this lecture is located at https://youtu.be/LQf2Vd-frsA.

## Outline

-Pole/Zero Cancellation

## Pole/Zero Cancellation

If a transfer function has a zero and a pole in the same location, they should cancel out.  However, numerical tools such as Matlab may not automatically recognize this cancellation due to a variety of reasons.  It therefore becomes necessary to use the ‘minreal’ function to perform this cancellation.

**Example 1:  Transfer function written as a polynomial**

Consider a transfer function of the form

$$G_1(s) = \frac{s^2+3s+2}{s^3+2s^2-11s-12} \qquad \textit{(Eq.1)}$$

```
G1[s_] = (s^2 + 3 s + 2)/(s^3 + 2 s^2 - 11 s - 12);
```

When the transfer function is written as a polynomial, it is not immediately obvious where the poles and zeros are.

We can find the poles and zeros

```
Solve[Numerator[G1[s]] == 0, s]
Solve[Denominator[G1[s]] == 0, s]
```

```
{{s → -2}, {s → -1}}
```

```
{{s → -4}, {s → -1}, {s → 3}}
```

So we see that there are

zeros at $s = -2$, $s = -1$
poles at $s = -4$, $s = -1$, $s = 3$

The pole and zeros at $s = -1$ should cancel so we see that we can write this as

$$G_{1,\text{minimal}} = \frac{(s+2)(s+1)}{(s+4)(s+1)(s-3)} = \frac{(s+2)}{(s+4)(s-3)}$$

```
G1minimal[s_] = (s + 2) / ((s + 4) (s - 3));

G1minimal[s] == G1[s] // Simplify
```

```
True
```

However, if we enter in the original form of the transfer function (Eq.1) into Matlab, it does not automatically recognize this

**Example 2: Matlab ‘zpk’**

Even if you create a system using ‘zpk’, the cancellation may not immediately occur and you should use ‘minreal’ to cancel the poles/zeros.

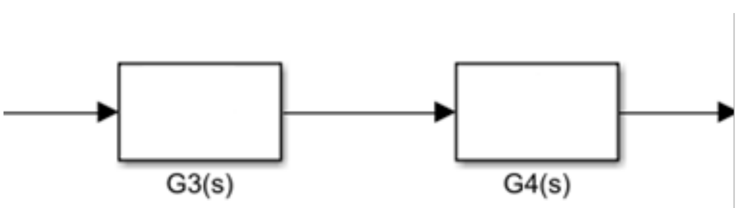
**Example 3: Composing Transfer Functions (in Series)**

The issue of minimal realizations comes into place when composing transfer functions together or performing block diagram algebra. For example, consider the system shown below with

$$G_3(s) = \frac{(s+1)}{(s+2)(s+3)}$$

$$G_4(s) = \frac{(s+2)}{(s+1)(s+3)}$$

```
G3[s_] = (s + 1) / ((s + 2) (s + 3));
G4[s_] = (s + 2) / ((s + 1) (s + 3));
```




It is obvious that the closed loop system is given by

$$T_1(s) = G_4(s)\, G_3(s)$$

$$= \frac{(s+2)}{(s+1)(s+3)} \frac{(s+1)}{(s+2)(s+3)}$$

$$T_1(s) = \frac{1}{(s+3)^2}$$

```
T1[s_] = G4[s] × G3[s]
```

$$\frac{1}{(3+s)^2}$$

However, in Matlab we again need to use ‘minreal’ to perform this cancellation

**Example 4: Composing Transfer Functions (feedback)**

In another example, consider the system with a simple feedback system


We know that the closed loop characteristic equation is simply

$$T_2(s) = \frac{G_3(s)}{1+G_3(s)}$$

$$\texttt{T2[s\_]} = \frac{\texttt{G3[s]}}{\texttt{1 + G3[s]}} \texttt{ // Simplify}$$

$$\frac{1+s}{7+6s+s^2}$$

Which is a 2nd order system as expected.

However if you do this operation directly in Matlab, you again need to use ‘minreal’ to ensure a minimal realization.

**Example 5: Numerical Roundoff Error**

$$\texttt{G5[s\_]} = \frac{\texttt{Expand[(s + 1.003) (s + 2)]}}{\texttt{Expand[(s + 1.001) (s + 3) (s + 4)]}}$$

$$\frac{2.006 + 3.003\,s + s^2}{12.012 + 19.007\,s + 8.001\,s^2 + s^3}$$

We see that there is a pole at $s = -1.001$ and zero at $s = -1.003$.

If you now use ‘minreal’ with the second parameter of TOL to specify the cancellation tolerance. Note that documentation on TOL is lacking and this does not appear to simply be the Euclidean distance in the s-plane for the cancellations. For example, the example above the distance between the poles is 0.002 but a tolerance of 0.001 leads to cancellations.