Using MATLAB for Axis-shifting

Recall that to perform axis-shifting from $Re = 0$ to $Re = s$ in the Routh-Hurwitz method, we substitute $\lambda + s$ for λ in the characteristic equation, then collect terms on λ to obtain a modified characteristic equation.

For example, suppose the initial characteristic polynomial is

$$\lambda^3 + 6\lambda^2 + 12\lambda + 8 = 0$$

and suppose we want to know how many roots have real parts greater than $s = -1$. We axis-shift to $s = -1$, then perform a Routh-Hurwitz analysis on the shifted polynomial. To perform the axis-shift, we substitute $\lambda - 1$ for λ and collect terms:

$$(\lambda - 1)^3 + 6 * (\lambda - 1)^2 + 12 * (\lambda - 1) + 8 = 0 = \lambda^3 + 3\lambda^2 + 3\lambda + 1$$

Often, the algebra required to perform axis-shifting is tedious, especially for larger-order polynomials and/or shifts to axes that are not integer values. The following MATLAB command can be used to perform axis-shifting for this example:

```
>> poly([1 1 1])+6*[0 poly([1 1])]+12*[0 0 poly([1])+[0 0 0 8]
```

which returns the coefficients of the shifted polynomial, i.e, [1 3 3 1]

Note the need to add zeros at the front of most of the terms in this expression. This is necessary because

```
>> poly([1 1 1])
```

returns the 4 coefficients of the third-order polynomial whose roots are (1, 1, 1) ; in other words, it creates the MATLAB representation of $(\lambda - 1)^3$, which is a vector of length 4. Similarly, >>poly([1 1]) creates the MATLAB representation of $(\lambda - 1)^2$, which is a vector of length 3. In order to correctly add these two vectors together, we need to add a zero (as the first term) to poly([1 1]) so that it is also a third-order polynomial, with $0 * \lambda^3$ as the first term. In other words, the command

```
>> poly([1 1 1])+6*[poly([1 1])]
```

returns an error message, because the two vectors are not the same size. However, the command

```
>> poly([1 1 1])+6*[0 poly([1 1])]
```

calculates the MATLAB representation of $(\lambda - 1)^3 + 6 * (\lambda - 1)^2$

This approach can be easily modified for other polynomials and/or for shifts to other axes. It can also be modified to accommodate symbolic coefficients. For example, to shift the polynomial

$$\lambda^4 + 3\lambda^3 + C * \lambda^2 + k * \lambda + 5 = 0$$

to the axis $s = -2$, we can use

```
>> syms C,k
```

followed by

```
>> poly([2 2 2 2])+3*[0 poly([2 2 2])]+C*[0 0 poly([2 2])+k*[0 0 0 poly([2])+[0 0 0 0 5]
```