Ve460 Control Systems Analysis and Design
A quick guide to Matlab Symbolic Toolbox

Jun Zhang

Shanghai Jiao Tong University

Matlab Symbolic Toolbox

Create symbolic variables:

```
>> syms x y s n
```

Create symbolic functions:

```
>> f=x^3+y^3;
>> g=(x^2-1)*(x^4+x^3+x^2+x+1)*(x^4-x^3+x^2-x+1);
```

Perform algebraic computations:

```
>> factor(f)
ans =
 [ x + y, x^2 - x*y + y^2]

>> expand(g)
ans =
 x^10 - 1
```

So we get

$$x^3 + y^3 = (x + y)(x^2 - xy + y^2).$$

Something harder:

```
>> z=factor(x^8+98*x^4*y^4+y^8)
z =
 [ x^4 - 4*x^3*y + 8*x^2*y^2 + 4*x*y^3 + y^4,
         x^4 + 4*x^3*y + 8*x^2*y^2 - 4*x*y^3 + y^4]
```

This gives us the factoring

$$\begin{aligned} x^8 + 98x^4y^4 + y^8 &= (x^4 - 4\,x^3\,y + 8\,x^2\,y^2 + 4\,x\,y^3 + y^4) \\ &\quad \cdot(x^4 + 4\,x^3\,y + 8\,x^2\,y^2 - 4\,x\,y^3 + y^4) \end{aligned}$$

To generate the above equation in LaTeX, we used the commands:

```
>> latex(z(1))
ans =
 x^4 - 4\, x^3\, y + 8\, x^2\, y^2 + 4\, x\, y^3 + y^4

>> latex(z(2))
ans =
 x^4 + 4\, x^3\, y + 8\, x^2\, y^2 - 4\, x\, y^3 + y^4
```

Now in the LaTeX file, we can simply copy & paste:

```
\begin{eqnarray*}
  x^8+98 x^4 y^4+y^8
   &=&(x^4 - 4\, x^3\, y + 8\, x^2\, y^2 + 4\, x\, y^3 + y^4) \\
   & & \cdot (x^4 + 4\, x^3\, y + 8\, x^2\, y^2 - 4\, x\, y^3 + y^4)
\end{eqnarray*}
```

Some Calculus calculations:

```
>> diff(sin(x)^2)
ans =
 2*cos(x)*sin(x)

>> int(sin(x)^2)
ans =
 x/2 - sin(2*x)/4

>> taylor(sin(x), 'Order', 8)
ans =
 - x^7/5040 + x^5/120 - x^3/6 + x
```

A real example

The linearized model of a nonlinear observer for a Permanent Magnet Synchronous Motor (PMSM) can be written as

$$\frac{d}{dt}\begin{bmatrix}\delta\tilde{\boldsymbol{\psi}}\\ \delta\tilde{\theta}\\ \delta\hat{\omega}_i\end{bmatrix} = \underbrace{\begin{bmatrix}-(\omega_0\mathbf{J}+\mathbf{K}) & \mathbf{K}\mathbf{J}\boldsymbol{\psi}_{a0} & 0\\ -k_p\boldsymbol{\lambda}^T\mathbf{J} & -k_p\boldsymbol{\lambda}^T\boldsymbol{\psi}_{a0} & -k_i\\ \boldsymbol{\lambda}^T\mathbf{J} & \boldsymbol{\lambda}^T\boldsymbol{\psi}_{a0} & 0\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}\delta\tilde{\boldsymbol{\psi}}\\ \delta\tilde{\theta}\\ \delta\hat{\omega}_i\end{bmatrix}}_{x} + \underbrace{\begin{bmatrix}0\\1\\0\end{bmatrix}}_{b}\delta\omega,$$

$$\delta\hat{\omega} = \underbrace{\begin{bmatrix}k_p\boldsymbol{\lambda}^T\mathbf{J} & k_p\boldsymbol{\lambda}^T\boldsymbol{\psi}_{a0} & k_i\end{bmatrix}}_{c}\begin{bmatrix}\delta\tilde{\boldsymbol{\psi}}\\ \delta\tilde{\theta}\\ \delta\hat{\omega}_i\end{bmatrix},$$

where ω_0 is the motor angular velocity, k_p, k_i are PI controller parameters, $\boldsymbol{\psi}_{a0} = \begin{bmatrix}d & q\end{bmatrix}^T$, and

$$\mathbf{J} = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix}k_1\\ k_2\end{bmatrix}\begin{bmatrix}d & q\end{bmatrix}, \quad \boldsymbol{\lambda} = \frac{1}{d^2+q^2}\begin{bmatrix}d\\ q\end{bmatrix}.$$

We want to calculate the Transfer Function (TF) from the input $\delta\omega$ to the output $\delta\hat{\omega}$, that is,

$$G(s) = c(s\mathbf{I} - A)^{-1}b.$$

We wrote the following Matlab code:

```
J=[0 -1; 1 0];

syms k1 k2 d q w0 s kp ki real;

K=[k1;k2]*[d q];
l=[d;q]/(d^2+q^2);

A=simplify( [ -(w0*J+K) K*J*[d;q] zeros(2,1);
             -kp*l'*J -kp*l'*[d;q] -ki;
             l'*J l'*[d;q] 0]);
b=[0 0 1 0]';
c=[kp*l'*J kp*l'*[d;q] ki];

Gs=simplify( c*inv(s*eye(4)-A)*b )
```

This yields the results

```
Gs =
 (ki + kp*s)/(s^2 + kp*s + ki)
```

Therefore, we obtain the desired TF as

$$G(s) = \frac{k_i + k_p s}{s^2 + k_p s + k_i}.$$