

MATLAB MODULE 2

Control System Toolbox and Symbolic Math Toolbox

Symbolic Math Toolbox

MATLAB's Symbolic Math (**Symbolic Math** ematics) toolbox allows users to perform symbolic mathematical computations using MATLAB. The only basic requirement is to declare symbolic variables before they are used. For control systems analysis and design, symbolic math toolbox is particularly important because of the following:

1. Transfer functions and other expressions can be entered in symbolic form as we write them in the notebook.
2. Symbolic expressions can be manipulated algebraically and simplified.
3. It is straightforward to convert symbolic polynomials to the vectors of corresponding power-term coefficients.
4. Expressions can be 'pretty printed' for clarity in the MATLAB command window.

Type **help symbolic** on the MATLAB command prompt to see all the functionalities available with Symbolic Math toolbox. In this section, we will learn commands of Symbolic Math toolbox particularly useful for control engineering.

First we demonstrate the power of Symbolic Math toolbox by calculating inverse Laplace transform. The following

MATLAB session gives the steps performed during the calculation of inverse Laplace transform of $G(s) = \frac{b}{(s+a)^2 + b^2}$ using Symbolic Math.

```

>> syms s a b;
>> G=b/((s+a)^2+b^2);
>> G

G =
b/((s+a)^2+b^2)

>> pretty(G)

      b
      -----
      2      2
    (s + a)  + b

>> ilaplace(G)

ans =
exp(-a*t)*sin(b*t)

```

Declare s, a , and b as symbolic variables

Enter symbolic transfer function

Normal print

Pretty print to improve readability

Inverse Laplace transform of $G(s)$

Exercise M2.8

Initialize **s** and **T** as symbols. Using Symbolic Math tools, find the response of the first- order system

$G(s) = \frac{K}{Ts+1}$, with the step excitation of strength A .

Hint: Find the inverse Laplace transform of $G(s) = \frac{A}{s} \times \frac{K}{Ts+1}$.

Exercise M2.9

Find manually the Laplace transform of $g(t) = 1 + \frac{1}{3}e^{-4t} - \frac{4}{3}e^{-t}$. Using Symbolic Math tools, declare **t** as symbolic variable and countercheck your result. Use the function **laplace(g)** to calculate the Laplace transform.

The function **simple(G)** finds simplest form of $G(s)$ with the least number of terms. **simplify(G)** can be used to combine partial fractions. Symbolic Math toolbox also contains other commands that can change the look of the displayed result for readability and form. Some of these commands are: **collect(G)** - collects common-coefficient terms of $G(s)$; **expand(G)** - expands product of factors of $G(s)$; **factor(G)**-factors $G(s)$; **vpa(G, places)** - stands for variable precision arithmetic (this command converts fractional symbolic terms into decimal terms with a specified number of decimal places).

Consider the function $G(s) = \frac{1}{s} + \frac{1}{3} \times \frac{1}{s+4} - \frac{4}{3} \times \frac{1}{s+1}$. MATLAB response to the simplification of $G(s)$ using the function **simplify** is shown below.

```
>> syms s
```

```
>> G = (1/s)+(1/3)*(1/(s+4))-(4/3)*(1/(s+1))
```

```
G =
```

```
1/s+1/3/(s+4)-4/3/(s+1)
```

```
>> pretty(G)
```

$$\frac{1}{s + 1/3} - \frac{4/3}{s + 4}$$

```
>> pretty(simplify(G))
```

$$\frac{4}{s(s + 4)(s + 1)}$$

In the above example, the symbolic fractions 1/3 and 4/3 will be converted to 0.333 and 1.33 if the argument **places** in **vap(G, places)** is set to 3.

```
>> pretty(vpa(G,3))
```

$$\frac{0.333}{s + 1.33} - \frac{1.33}{s + 4.}$$

Exercise M2.10

Consider the function $g(t) = 1 + \frac{b}{a-b} \times e^{-at} - \frac{a}{a-b} \times e^{-bt}$. Show using Symbolic Math tools that the simplified form of its Laplace transform is $G(s) = \frac{ab}{s(s+a)(s+b)}$.

Basic mathematical operators +, −, ×, and / are applicable to symbolic objects also. For example, simplification of the closed-loop system with forward gain K , forward-path transfer function $G(s) = \frac{2.5}{s(s+5)}$, and feedback-path transfer function $H(s) = \frac{1}{0.1s+1}$, is given by $M(s) = \frac{KG(s)}{1+G(s)H(s)}$. This can be done using Symbolic Math toolbox as follows

```
>> syms s K;
```

```
>> G = 2.5/(s*(s+5));
```

```
>> H = 1/(0.1*s+1);
```

```
>> M = K*G/(1+G*H)
```

```
M =
```

```
5/2*K/s/(s+5)/(1+5/2/s/(s+5)/(1/10*s+1))
```

```
>> pretty(M)
```

$$\frac{5/2 \cdot K}{s(s+5) \left(1 + \frac{5/2}{s(s+5) \left(\frac{1}{10s+1} \right)} \right)}$$

```
>> pretty(simplify(M))
```

$$\frac{5/2 \cdot (s+10) K}{s^3 + 15s^2 + 50s + 25}$$

With Symbolic Math toolbox, symbolic transfer functions can easily be converted to the LTI (Linear Time-Invariant) transfer function objects. This conversion is done in two steps. The first step uses the command `numden` to extract the symbolic numerator and denominator of $G(s)$. The second step converts, separately, the numerator and denominator to vectors using the command `sym2poly`. The last step consists of forming the LTI transfer function object by using the vector representation of the transfer function's numerator and denominator. The command `sym2poly` doesn't work on symbolic expressions with more than one variable.

As an example, we form the LTI object from $M(s)$ obtained in the above example with $K = 1$.

```
>> K = 1;
>> M = K*G/(1+G*H);
>> [nums, dens] = numden(M)
```

```
nums =
```

```
5*s*(s+5)*(s+10)
```

Extract numerator and denominator polynomials in symbolic form

```
dens =
```

```
2*s*(s+5)*(s^3+15*s^2+50*s+25)
```

```
>> num = sym2poly(nums)
```

```
num =
```

```
5    75    250    0
```

Convert numerator and denominator symbolic polynomials to the vectors of coefficients of power terms

```
>> den = sym2poly(dens)
```

```
den =
```

```
2    40    250    550    250    0
```

```
>> M_LTI = tf(num,den)
```

Form LTI transfer function from numerator and denominator coefficient vectors

```
Transfer function:
```

```
5 s^3 + 75 s^2 + 250 s
```

```
-----
2 s^5 + 40 s^4 + 250 s^3 + 550 s^2 + 250 s
```

```
>> M_LTI = minreal(M_LTI)
```

```
Transfer function:
```

The command [poly2sym](#) converts polynomial coefficient vectors to symbolic polynomial. Learn more about [poly2sym](#) using online help.