EENG 428 Introduction to Robotics Laboratory

EXPERIMENT 2

Symbolic MATLAB

Objectives:

This experiment familiarizes students with the skills of creating symbolic variables and expressions, defining symbolic matrices and the associated simplification with the MATLAB environment.

1. Creating Symbolic variables and Expressions

MATLAB allows you to create symbolic math expressions. This is useful when you don't want to immediately compute an answer, or when you have a math "formula" to work on but don't know how to "process" it.

The key function in MATLAB to create a symbolic representation of data is: ***sym()***, or ***syms()*** if you have multiple symbols to make.

a. Defining Symbolic Expressions

We can define symbolic functions using the ***sym*** and ***syms*** commands. Here is an example of creating a symbolic function for $f = ax^2 + bx + c$:

```
>> syms a b c x % define symbolic math variables

>> f = sym('a*x^2 + b*x + c');
```

From now on we can use the ***f*** symbol to represent the given function.

b. Evaluation of Symbolic Expressions

The key function ***subs*** (which stands for substitute) is used to replace symbolic variables with either new symbolic variables or with actual values. The syntax for the function is: ***subs(symbolic function, list of symbols, list of values)***. Here is an example:

```
>>syms x f
>>   f = sym( 'a*x^2 + b*x + c' );

>> subs(f,x,5)

ans =

25 * a + 5 * b + c

>> subs(f,[x a b c],[5 1 2 3])

ans =
```

```
38
```

c. Plotting Symbolic Functions

In MATLAB, we can plot a symbolic function over one variable by using the ***ezplot*** function. Here is an example:

```
>>syms x y

>> y = sin(x)

 y =

 sin(x)

 >> ezplot(y)
```

d. *"Pretty"* Printing Symbolic Functions

When you want to print a symbolic function to make it easier for the user of the program to read, you can use the "pretty" function. Here is an example:

```
 >>syms x

>>syms f

>>f = sin(x)^2 + cos(x)^2;

        pretty( f )

                              2        2

                        sin(x)  + cos(x)
```

2. Symbolic Matrices

a. Creating a Matrix of Symbolic Variables

A matrix A whose elements are symbolic variables can be constructed of these variables by firstly defining them, and then explicitly defining A in terms of these variables.

Example 2.1 Symbolic Matrix Definition

Define the following symbolic matrix, and find its inverse.

$$A = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & c32 & c34 \end{bmatrix}$$

A is defined as:

```
>> syms a11 a12 a13 a21 a22 a23 a31 a32 a33

>> A=[a11 a12 a13 ; a21 a22 a23;a31 a32 a33]
```

, returns A as:

```
A =



[ a11, a12, a13]

[ a21, a22, a23]

[ a31, a32, a33]
```

, while the inverse is found as:

```
>> inv(A)
```

, which is returned as:

```
[  (a22*a33 - a23*a32)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31), -(a12*a33 - a13*a32)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31),  (a12*a23 - a13*a22)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31)]

[ -(a21*a33 - a23*a31)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31),  (a11*a33 - a13*a31)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31), -(a11*a23 - a13*a21)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31)]

[  (a21*a32 - a22*a31)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31), -(a11*a32 - a12*a31)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31),  (a11*a22 - a12*a21)/(a11*a22*a33 - a11*a23*a32 - a12*a21*a33 + a12*a23*a31 + a13*a21*a32 - a13*a22*a31)]
```

b. Generating Elements While Creating a Matrix* (with the newest versions of MATLAB, only)

A symbolic matrix can be directly defined without having the need to firstly define its symbolic elements. This is achieved by defining the matrix symbol and its dimension once at a time, as follows:

Example : Define a 2-by-4 symbolic matrix

```
A = sym('A', [2 4])

A =

[ A1_1, A1_2, A1_3, A1_4]
```

```
[ A2_1, A2_2, A2_3, A2_4]
```

c. Creating a Matrix of Symbolic Numbers

A numerical matrix can be rewritten in a precise symbolic form by converting it to a symbolic matrix.

Example: Creating a Matrix of Symbolic Numbers

The matrix

$$A = \begin{bmatrix} 1 & 0.125 & 0.75 \\ 0.375 & 0.625 & 0.875 \\ 0.5 & 1.125 & 0.25 \end{bmatrix}$$

Can be rewritten in its precise form as a symbolic matrix.

A is firstly defined as:

```
a=[1.0000    0.1250    0.7500 ;  0.3750    0.6250    0.8750 ;  0.5000    1.1250    0.2500]

a =

    1.0000    0.1250    0.7500

    0.3750    0.6250    0.8750

    0.5000    1.1250    0.2500
```

A is written in its precise form by:

```
A=sym(a)

A =

[   1, 1/8, 3/4]

[ 3/8, 5/8, 7/8]

[ 1/2, 9/8, 1/4]
```

3. Homework

Considering the following questions, provide your answers in a report including your codes and associated outputs and/or plots by copying them and pasting on the report pages. To be submitted as a hard copy on next lab session. You should work individually and group work will be penalized.

1. Write a MATLAB code defines and plots the following symbolic function and evaluates its value at the point x=2, where a=3, b=4 and c=5.
$$f = ax^2 + bx + c$$
2. Write a MATLAB code that defines two symbolic 3x3 matrices A and B, and then adds the first row of A to the first column of B and finds 3A+7B.