Hands On: Introduction to MATLAB - Part 4 - Functions

This is the fourth MATLAB live script of the collection ***Hands On: Using MATLAB in the 267MI "System Dynamics" course***, devoted to introduce the MATLAB/Simulink environment and tools for solving practical problems related to the topics of the 267MI course, i.e. performance analysis of dynamic systems, parametric estimation, identification of models from data, and prediction of the evolution of dynamic systems.

Use this link to go back to the main live script of the collection.

Table of Contents

Objectives

The aim of this live script is to illustrare how to use functions and to define custom function and anonymous functions in MATLAB.

What you should know by the end of this module:

- how to call functions provided as part of MATLAB;
- use of input and output arguments;
- how to create a custom MATLAB function.

An Introduction

Functions are an essential mechanism for grouping a sequence of frequently used MATLAB commands to perform a specific task. Examples include all the standard mathematical functions such as `sin`, `cos`, `tan`, `exp`, `log`, `sqrt`, `abs` and many others. MATLAB also offers the possibility of **defining custom functions**. In this live script, we will only look at the simplest way of doing this, using anonymous functions. This is essential for problems such as solving non-linear equations, integrating a function and minimising or maximising a function. One of the great powers of MATLAB is the ability to apply functions to vectors and matrices with any number of elements in a single call.

The General Sintax

MATLAB provides the ability to define your own functions:

- An M-file is a **MATLAB function** if and only if the first line of the file contains the following text

```
function [output variable list] = function_name (input variable list)
```

- Terminating the file with a keyword is unnecessary: usually, a function ends when the last instruction line of the M-file describing it (which may be an "`end`" MATLAB keyword) is reached.
- You can force termination at any point using the '`return`' instruction (see `help return` for details).

An Example

Calculating the area of any triangle according to the formula

$$\mathcal{A} = \sqrt{s \cdot (s-a) \cdot (s-b) \cdot (s-c)}\,, \quad s = \frac{a+b+c}{2}$$

where a, b, c are the lengths of the triangle sides.

The function then (in its simplest version) has 3 input parameters and only 1 output parameter:

```
edit evalTriangleArea
```

Please note the initial comment lines in the M-file text. They are usefull to implement the help contents for the custom MATLAB function.

To see how the help command uses such helping lines, just run the following

```
help evalTriangleArea
```

```
  The function evalTriangleArea(a, b, c)
  computes the area of a triangle whose
  sides have length a, b and c.
  Inputs:
      a,b,c: Lengths of sides
  Output:
      A: area of triangle
  Usage:
      Area = evalArea(2,3,4);
  Written by XXX, MM/DD/YY.
```

Using the custom function

- assigning an output variable

```
format short
Area = evalTriangleArea(10,15,20)
```

```
Area = 72.6184
```

- without assigning the output variable

```
evalTriangleArea(10,15,20)
```

```
ans = 72.6184
```

Some Built-in Functions

You may find the list of built-in elementary mathematical function by typing

```
help elfun
```

```
  Elementary math functions.

  Trigonometric.
    sin         - Sine.
    sind        - Sine of argument in degrees.
    sinh        - Hyperbolic sine.
    asin        - Inverse sine.
    asind       - Inverse sine, result in degrees.
    asinh       - Inverse hyperbolic sine.
    cos         - Cosine.
    cosd        - Cosine of argument in degrees.
    cosh        - Hyperbolic cosine.
    acos        - Inverse cosine.
    acosd       - Inverse cosine, result in degrees.
    acosh       - Inverse hyperbolic cosine.
    tan         - Tangent.
    tand        - Tangent of argument in degrees.
    tanh        - Hyperbolic tangent.
    atan        - Inverse tangent.
    atand       - Inverse tangent, result in degrees.
    atan2       - Four quadrant inverse tangent.
    atan2d      - Four quadrant inverse tangent, result in degrees.
    atanh       - Inverse hyperbolic tangent.
    sec         - Secant.
    secd        - Secant of argument in degrees.
    sech        - Hyperbolic secant.
    asec        - Inverse secant.
    asecd       - Inverse secant, result in degrees.
    asech       - Inverse hyperbolic secant.
    csc         - Cosecant.
    cscd        - Cosecant of argument in degrees.
    csch        - Hyperbolic cosecant.
    acsc        - Inverse cosecant.
    acscd       - Inverse cosecant, result in degrees.
    acsch       - Inverse hyperbolic cosecant.
    cot         - Cotangent.
    cotd        - Cotangent of argument in degrees.
    coth        - Hyperbolic cotangent.
    acot        - Inverse cotangent.
    acotd       - Inverse cotangent, result in degrees.
    acoth       - Inverse hyperbolic cotangent.
    hypot       - Square root of sum of squares.
    deg2rad     - Convert angles from degrees to radians.
    rad2deg     - Convert angles from radians to degrees.

  Exponential.
```

```
    exp         - Exponential.
    expm1       - Compute exp(x)-1 accurately.
    log         - Natural logarithm.
    log1p       - Compute log(1+x) accurately.
    log10       - Common (base 10) logarithm.
    log2        - Base 2 logarithm and dissect floating point number.
    pow2        - Base 2 power and scale floating point number.
    realpow     - Power that will error out on complex result.
    reallog     - Natural logarithm of real number.
    realsqrt    - Square root of number greater than or equal to zero.
    sqrt        - Square root.
    nthroot     - Real n-th root of real numbers.
    nextpow2    - Next higher power of 2.

  Complex.
    abs         - Absolute value.
    angle       - Phase angle.
    complex     - Construct complex data from real and imaginary parts.
    conj        - Complex conjugate.
    imag        - Complex imaginary part.
    real        - Complex real part.
    unwrap      - Unwrap phase angle.
    isreal      - True for real array.
    cplxpair    - Sort numbers into complex conjugate pairs.

  Rounding and remainder.
    fix         - Round towards zero.
    floor       - Round towards minus infinity.
    ceil        - Round towards plus infinity.
    round       - Round towards nearest integer.
    mod         - Modulus (signed remainder after division).
    rem         - Remainder after division.
    sign        - Signum.
```

or using the MATLAB help browser (search for **Elementary Math**)

```
doc Elementary Math
```

Functions Applied to a Vector

Most MATLAB functions have been modified (overloaded) so they work with **inputs** which are **vectors** as well as scalars. When an argument is a vector the function is applied to each element of the vector, producing a vector of the same size as the input vector.

Examples

```
sqrt(1:10)
```

```
ans = 1×10
    1.0000    1.4142    1.7321    2.0000    2.2361    2.4495    2.6458    2.8284 ···
```

```
xv = -1:1/2:1 % a row vecor with 5 elements
```

```
xv = 1×5
```

```
   -1.0000   -0.5000         0    0.5000    1.0000
```

```
sin( (pi/2)*xv )
```

```
ans = 1×5
   -1.0000   -0.7071         0    0.7071    1.0000
```

Sometimes the result of the function has fewer elements than the input:

```
sum(xv)
```

```
ans = 0
```

```
max(xv)
```

```
ans = 1
```

Summary

Using this live script you have:

- learnt about MATLAB's wide range of elementary mathematical functions;
- learnt how to write an own custom MATLAB function;
- learnt that many of MATLAB's functions accept a vector as input, creating a vector as output, where the function has been evaluated at each element of the input vector.

Back to the Index

Use this link to go back to the main live script of the collection.

Back to the Previous Part: Vectors & Matrices

Use this link to go back to the previous live script of this collection.

Go to the Next Part: LTI Systems

Use this link to go to the next live script of the collection.