Plotting Data

\[ \cos\left(\frac{t}{10}\right) \]

Figure by MIT OCW.

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
MatLab Data Types

Adapted from MATLAB Help Sections. Figure by MIT OCW.
Numeric Types

Number representation

Binary: 0, 1 1011101
Decimal: 0-9 93
Hexadecimal 0-9, A,B,C,D,E,F 5D

Bit: a single binary digit
Byte: 8 binary digits
Word: 16 binary digits
Double Word: 32 binary digits
Numeric Types In MatLab

Integers: 12523

Floating points: 1.234e-72

Complex number: 2.3+5.2i

Infinity and NAN: Inf (1/0), NAN (not a number)
## MatLab Integer Type

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Range of Values</th>
<th>Conversion Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed 8-bit integer</td>
<td>-2^7 to 2^7-1</td>
<td>int8</td>
</tr>
<tr>
<td>Signed 16-bit integer</td>
<td>-2^15 to 2^15-1</td>
<td>int16</td>
</tr>
<tr>
<td>Signed 32-bit integer</td>
<td>-2^31 to 2^31-1</td>
<td>int32</td>
</tr>
<tr>
<td>Signed 64-bit integer</td>
<td>-2^63 to 2^63-1</td>
<td>int64</td>
</tr>
<tr>
<td>Unsigned 8-bit integer</td>
<td>0 to 2^8-1</td>
<td>uint8</td>
</tr>
<tr>
<td>Unsigned 16-bit integer</td>
<td>0 to 2^16-1</td>
<td>uint16</td>
</tr>
<tr>
<td>Unsigned 32-bit integer</td>
<td>0 to 2^32-1</td>
<td>uint32</td>
</tr>
<tr>
<td>Unsigned 64-bit integer</td>
<td>0 to 2^64-1</td>
<td>uint64</td>
</tr>
</tbody>
</table>

Adapted from MATLAB Help Sections. Figure by MIT OCW.
MatLab Floating Point Type

Uses IEEE Standard 754: Double type, Single type

\[ +2.2345 \times 10^{-34} \]

### Double

<table>
<thead>
<tr>
<th>Bits</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>Sign (0 = positive, 1 = negative)</td>
</tr>
<tr>
<td>62 to 52</td>
<td>Exponent, biased by 1023</td>
</tr>
<tr>
<td>51 to 0</td>
<td>Fraction ( f ) of the number ( 1.f )</td>
</tr>
</tbody>
</table>

### Single

<table>
<thead>
<tr>
<th>Bits</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Sign (0 = positive, 1 = negative)</td>
</tr>
<tr>
<td>30 to 23</td>
<td>Exponent, biased by 127</td>
</tr>
<tr>
<td>22 to 0</td>
<td>Fraction ( f ) of the number ( 1.f )</td>
</tr>
</tbody>
</table>

Adapted from MATLAB Help Sections. Figures by MIT OCW.
MatLab Complex Number Type

“i” or “j” are specially reserved symbols in MatLab

Complex numbers are represented as:
A+Bi

C=complex(1,2)  C=1+2i
D=real(C)        D=1
E=imag(C)        E=2

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Logical (boolean) Types In MatLab

Logical state (e.g. $5>2$) is represented by:

1 or 0

True or False

Logical types are important in programming when decision must be made depending on the validity (true or false) of some conditions.

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
Character Types In MatLab

‘a’, ‘z’, ‘8’, ‘&’ are characters

MatLab treat any symbol placed inside SINGLE quotes as an array of characters!

Internally, each character is represented by an 8-bit number using Unicode (ASCII) decoding system.
Logical (boolean) Types In MatLab

Logical state (e.g. 5>2) is represented by:
1 or 0
True or False

Logical types are important in programming when decision must be made depending on the validity (true or false) of some conditions.
Constructing Matrix in MatLab

ones: matrix of all ones
zeros: matrix of all zero
eyes: Identity matrix
randn: Random matrix

ones(4, 6, ‘uint32’) creates a 4x6 matrix containing ones represented as unsigned 32 bit integer

randn(2) creates a 2x2 matrix containing random numbers of standard normal distribution

rand(2) creates a 2x2 matrix containing random numbers uniformly distributed between 0 and 1
Basic Linear Algebra

Scalar multi

Add/sub

Matrix multi

Inverse

Transpose

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
MatLab Operators – numeric

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>.*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>./</td>
<td>Right division</td>
</tr>
<tr>
<td>.\</td>
<td>Left division</td>
</tr>
<tr>
<td>+</td>
<td>Unary plus</td>
</tr>
<tr>
<td>-</td>
<td>Unary minus</td>
</tr>
<tr>
<td>:</td>
<td>Colon operator</td>
</tr>
<tr>
<td>.^</td>
<td>Power</td>
</tr>
<tr>
<td>.'</td>
<td>Transpose</td>
</tr>
<tr>
<td>'</td>
<td>Complex conjugate transpose</td>
</tr>
<tr>
<td>*</td>
<td>Matrix multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Matrix right division</td>
</tr>
<tr>
<td>\</td>
<td>Matrix left division</td>
</tr>
<tr>
<td>^</td>
<td>Matrix power</td>
</tr>
</tbody>
</table>

\[
A = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \quad C = \begin{bmatrix} 5 & 6 \end{bmatrix}
\]

\[
A + C = \begin{bmatrix} 6 & 8 \end{bmatrix}
\]

\[A * C \quad \text{bad!}\]

\[A * B = 11\]

\[A .* C = \begin{bmatrix} 5 & 12 \end{bmatrix}\]

\[A ./ C = \begin{bmatrix} 0.200 & 0.333 \end{bmatrix}\]

\[A .\backslash C = \begin{bmatrix} 5 & 3 \end{bmatrix}\]

Adapted from MATLAB Help Sections. Figure by MIT OCW.

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
### MatLab Operators – Relational, Logical

#### Relational

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>~=</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

#### Logical

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>Returns 1 for every element location that is true (nonzero) in both arrays, and 0 for all other elements.</td>
<td>A &amp; B = 01001</td>
</tr>
<tr>
<td></td>
<td>Returns 1 for every element location that is true (nonzero) in either one or the other, or both arrays, and 0 for all other elements.</td>
<td>A</td>
</tr>
<tr>
<td>~</td>
<td>Complements each element of the input array, A.</td>
<td>~A = 10010</td>
</tr>
<tr>
<td>xor</td>
<td>Returns 1 for every element location that is true (nonzero) in only one array, and 0 for all other elements.</td>
<td>xor (A, B) = 10100</td>
</tr>
</tbody>
</table>

Adapted from MATLAB Help Sections. Figure by MIT OCW.

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
One more MatLab Operator – Sequence

“:” is the sequence operator that denote a range

\[
A = 2:5 \quad A = [2 \ 3 \ 4 \ 5] \\
A = 2:3:11 \quad A = [2 \ 5 \ 8 \ 11]
\]

\[
A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]

\[
B = A(2,:) \quad B = [4 \ 5 \ 6]
\]

\[
C = A(:,2) \quad C = \begin{bmatrix}
2 \\
5 \\
8 \\
\end{bmatrix}
\]

\[
D = A(2:3,:) \quad D = \begin{bmatrix}
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]

\[
A(:,2) = [] \quad A = \begin{bmatrix}
1 & 3 \\
4 & 6 \\
7 & 9 \\
\end{bmatrix}
\]

It is very useful to create, decimate, and generate submatrix
Basic Graphic Output in MatLab

\[ X = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10] \]
\[ Y = [1 \ 4 \ 9 \ 16 \ 25 \ 36 \ 49 \ 64 \ 81 \ 100] \]

\texttt{plot(X,Y)}
More Graphic Output

t=1:1:100;
plot(t,cos(t/10));

Plotting Data

Cite as: Peter So, course materials for 2.003J/1.053J Dynamics and Control I, Spring 2007.
MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYYY].
t=1:1:100;
plot(t, cos(t/10), 'bo-');
title('Plotting Data2');
xlabel('t');
ylabel('cos(t/10)');
legend('sim data');
A couple more very useful graphic commands

(1) hold on/hold off – determines whether the next plot command overwrites or not

(2) figure – Creates new figure window

(3) From the figure window, under “edit menu”, the “copy figure” option allows you to copy the figure to the clipboard and then you can cut and paste it into other programs such as MSWord.