Introduction to MATLAB II

Steve Gu
Jan 25, 2007
Outline

• Matrix Operation
  – Matrix functions
  – Element-wise operations

• Dynamic Systems
  – Classification
  – 2nd→1st Order Equations

• Introduction to Simulink
Part I

• Matrix Operation
  – Matrix functions
  – Element-wise operations
Matrix Functions

• Many mathematical functions work on the individual entries of a matrix or vector
• For example, \texttt{abs}(M) takes the absolute value of each entry of \texttt{M}:

\begin{verbatim}
>> abs( [1.2 -0.3 -0.5; 0.4 -0.7 0.2] )
     1.2    0.3    0.5
     0.4    0.7    0.2
\end{verbatim}

• Other such functions are \texttt{sin}, \texttt{cos}, \texttt{sqrt}, etc.
Matrix Functions

• Many functions work on the columns of matrices

• Example, the max function:
  - \( \text{max}(v) \) finds the maximum entry of a vector
  - \( \text{max}(M) \) returns a row vector containing the maximum entry of each column of a matrix
  - \( \text{max}(\text{max}((M))) \) returns the maximum entry a matrix
Matrix Functions

```
>> max( [3 5 2 -3] )
   5
>> M = [3 8 2 -1; 5 3 7 3; 6 -10 5 2; 9 3 4 3]
   3  8  2  -1
   5  3  7  3
   6 -10 5  2
   9  3  4  3
>> max( M )  % maximum in each column
   9  8  7  3
>> max( M' )  % maximum in each row
   8  7  6  9
```
Similar Functions

• Similar functions include:
  - max   min   sum
  - mean  sort

```plaintext
>> M = [1 2 3; 4 5 6; 7 8 9]
   1       2       3
   4       5       6
   7       8       9

>> sum( M )
   12      15      18

>> min( M )
   1       2       3
```
Similar Functions

- Some functions work on the entire column:
  - sort

```matlab
>> M = rand(3, 6)
0.21116 0.61004 0.77120 0.74940 0.55795 0.20212
0.55104 0.43918 0.06933 0.09868 0.82398 0.24698
0.19782 0.94107 0.20374 0.41644 0.91337 0.96385

>> sort(M)
0.19782 0.43918 0.06933 0.09868 0.55795 0.20212
0.21116 0.61004 0.20374 0.41644 0.82398 0.24698
0.55104 0.94107 0.77120 0.74940 0.91337 0.96385
```
Matrix Functions

• some mathematical functions work on the entire matrix
• For example, \texttt{det(M)} takes the determinant of matrix \texttt{M}:

\begin{verbatim}
>> det( [1 2;3 4] )
>>ans
   -2
\end{verbatim}

• Other such functions are \texttt{eig}, \texttt{norm}, \texttt{cond}, etc.
Element-wise Products

• How have heard repeatedly that if $A = (a_{i,j})$ and $B = (b_{i,j})$ are matrices, then $AB$ means matrix multiplication:

$$
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}
\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}
= 
\begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}
$$

• Often though, the simple product is useful:

$$
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}
\times
\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}
= 
\begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{pmatrix}
$$
Element-wise Products

- Matlab allows you to do this with the . * operator:

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 2 & 4 \\
8 & 16 & 32
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 4 & 12 \\
32 & 80 & 192
\end{bmatrix}
\]
Element-wise Powers

• You also learned that if \( A = (a_{i,j}) \) then \( A^n \) means repeated matrix multiplication:

\[
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^2 = \begin{pmatrix} a_{11}^2 + a_{12}a_{21} & a_{11}a_{12} + a_{12}a_{22} \\ a_{21}a_{11} + a_{22}a_{21} & a_{21}a_{12} + a_{22}^2 \end{pmatrix}
\]

• Again, sometimes you just want to raise each entry to an exponent:

\[
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{\otimes 2} = \begin{pmatrix} a_{11}^2 & a_{12}^2 \\ a_{21}^2 & a_{22}^2 \end{pmatrix}
\]
Element-wise Powers

• Matlab allows you to do this with the .^ operator:

```matlab
>> A = [1 2 3; 4 5 6]
      1   2   3
      4   5   6
>> A.^2
      1   4   9
      16  25  36
>> A.^6
      1   32  243
  1024 3125 7776
```
Dot Product

• Note that matrix multiplication is a series of dot products:

```matlab
>> A = [1 2; 3 4];
>> B = [2 3; 5 7];
>> A*B

    12    17
    26    37

>> [A(1,:)*B(:,1)  A(1,:)*B(:,2)  A(2,:)*B(:,1)  A(2,:)*B(:,2)];

    12    17
    26    37
```
Part II

• Dynamic Systems
  – Classification
  – 2nd $\rightarrow$ 1st Order Equations
Dynamic Systems

\[ x(n + 1) = f(x(n), u(n), n) \]
\[ y(n) = h(x(n), n) \]

- What we consider is Linear, Deterministic, Stationary, Discrete Dynamic Systems:

\[ x(0) = x_0 \]
\[ x(n + 1) = Fx(n) + Gu(n) \]
\[ y(n) = Hx(n) . \]
Last Week…

The equation for the motion:

\[ y(n) = 2ay(n - 1) - by(n - 2) \]

Remark: Second Order Difference Equation
2nd $\rightarrow$ 1st Order Equations

\[ a(n + 1) = 2ba(n) - ca(n - 1) \]

We let

\[ x(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} a(n) \\ a(n - 1) \end{bmatrix} \]

\[ \begin{align*}
  x(0) &= x_0 \\
  x(n + 1) &= Fx(n) + Gu(n) \\
  y(n) &= Hx(n)
\end{align*} \]
2nd→1st Order Equations

We let

\[
x(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} a(n) \\ a(n-1) \end{bmatrix}
\]

so that

\[
x(n+1) = \begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} a(n+1) \\ a(n) \end{bmatrix} = \begin{bmatrix} 2ba(n) - ca(n-1) \\ a(n) \end{bmatrix}
\]

\[
= \begin{bmatrix} 2bx_1(n) - cx_2(n) \\ x_1(n) \end{bmatrix} = \begin{bmatrix} 2b & -c \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = \begin{bmatrix} 2b & -c \\ 1 & 0 \end{bmatrix} x(n)
\]

This has the form

\[
x(n+1) = Fx(n) \quad \text{with} \quad F = \begin{bmatrix} 2b & -c \\ 1 & 0 \end{bmatrix}.
\]
Also, we have: \( G=0, \ H=[1,0] \)

Therefore, we’ve reduced 1 second order equation to a system of 2 first order equations

\[
\begin{align*}
  x(0) &= x_0 \\
  x(n+1) &= Fx(n) + Gu(n) \\
  y(n) &= Hx(n)
\end{align*}
\]

With

\[
F = \begin{bmatrix}
  2b & -c \\
  1 & 0
\end{bmatrix}
\]
Part III

• Introduction to Simulink
Starting Simulink

1. Start MATLAB
2. Click the Simulink icon on MATLAB toolbar; Enter the `simulink` command at the MATLAB prompt
3. Starting Simulink displays the Simulink Library Browser
Simulink Library Browser

Displays a tree-structured view of the Simulink block libraries
Signal Generator

Generate various waveforms

Parameters and Dialog Box
Simulation Parameters

- Set the simulation parameters by choosing Simulation Parameters from the Simulation menu.
Starting the Simulation

- Pull down the Simulation menu and choose the Start command (or Ctrl+T)
Blocksets

Blocksets are specialized collections of Simulink blocks built for solving particular classes of problems.
A Working Example

\[ a(n + 1) = 2ba(n) - ca(n - 1) \]
Results & Demo

\[ y(n) = Cx(n) + Du(n) \]

\[ x(n+1) = Ax(n) + Bu(n) \]
• Thank you
• Q&A