SUSTech Southern University of Science and Technology

Lab 1. Matlab Basics

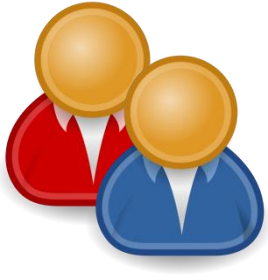
Cheng PENG
Department of Biomedical Engineering
pengc@sustech.edu.cn
Fall 2021

Overview

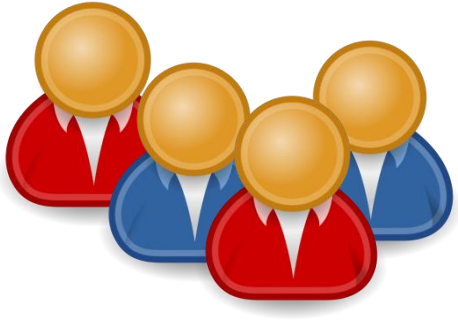
- Objective: to simulate/verify the algorithm of “Signals & Systems” by Matlab.

4 + (1) lab assignments + 2 Projects

By group of 2 students



By group of 4 students



Labs and Projects

- Lab 1. Matlab Basics
- Lab 2. Linear Time-Invariant Systems
- Lab 3. Fourier Series Representation of Periodic Signals
- Lab 4. The Continuous-Time Fourier Transform
- Lab 5. Preparation for project 1
- Project 1. Speech Synthesis and Perception with Envelope Cue
- Project 2. Basic Principle of OFDM Technology

	Lab assignment	**Project**
	4+1 4 for 4 chapters, 1 for project #1	2
	2 members	4 members
	3/2/2/2/1 weeks Hard deadline: please submit your report before the next lab	3/3 weeks Deadline: TBA
	Report + Matlab code	Essay + Matlab code, 1 presentation of either project

- Edit your report in the following format:

不是填表格，
没有固定格式
要求

Write an short introduction to the lab assignment

Type down Question 1
Give you answer to Q1, add the figures if necessary

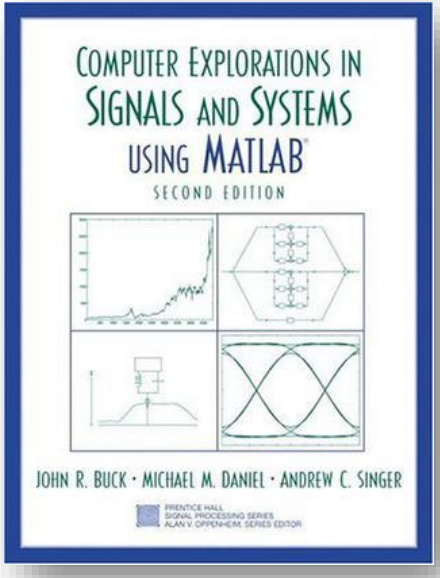
Type down Question 2
Give you answer to Q2, add the figures if necessary

Type down Question 3
Give you answer to Q3, add the figures if necessary

…

- Attach your Matlab codes if any.
- Submit reports in 'pdf' (preferable) or 'docx'. DO NOT submit files in Mac-only formats, e.g. 'pages'
- Submit to Blackboard

- Reference:
  - John R. Buck, Michael M. Daniel, Andrew C. Singer. *Computer Explorations in Signals and Systems Using MATLAB*. 2nd edition. Prentice Hall, 2002.



一丹图书馆/Yidan Library 4层阅览区 TN911.72 /E 49 第4排B面
隐藏详情

可用性:
(2个复本, 0个可用, 0个请求)

1 - 2 的 2 记录

条码	类型	政策	描述	状态
SUST0050038431	图书	60天外借		外借截止于 05/11/2021 23:59:00 CST
SUST0050017759	图书	60天外借		外借截止于 13/10/2021 23:59:00 CST

What's Simulation?

The proposed Algorithm B for LTE system is better than the existing Algorithm A

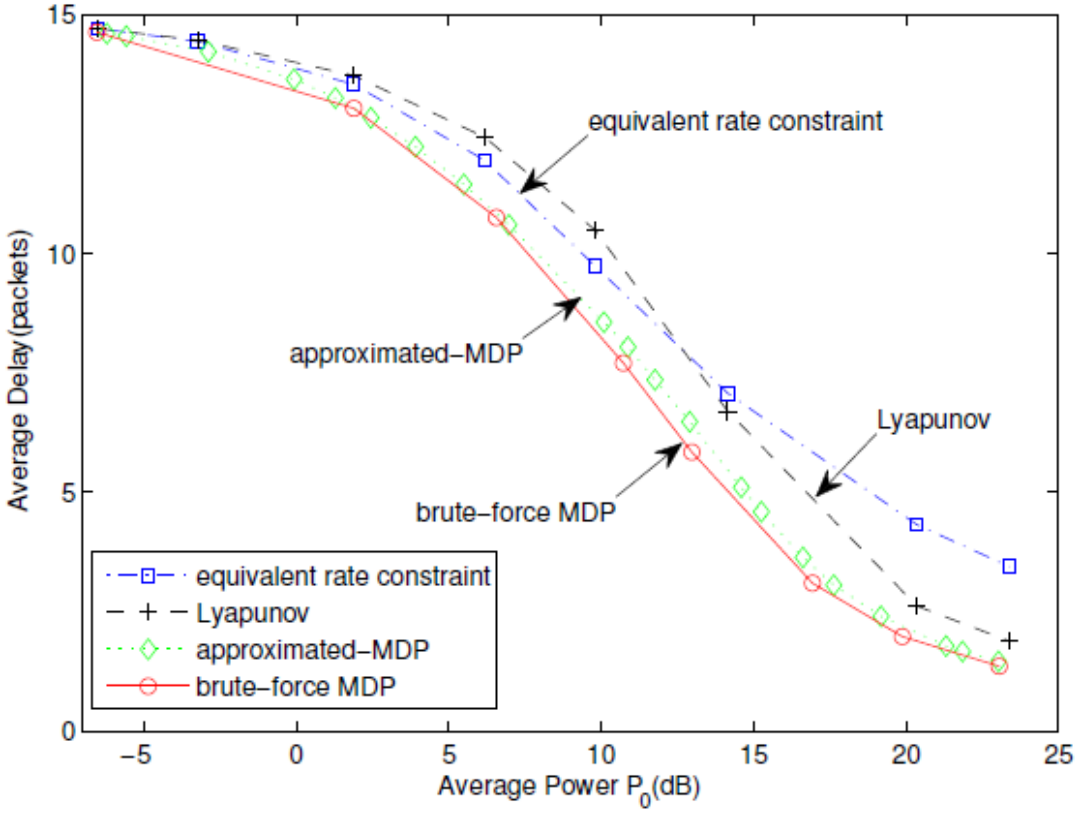
Derive the math expression of performance for Algorithm A & B

Test Algorithm A & B by computer with typical system configuration: 100 users/cell, 250m cell radius and blah-blah

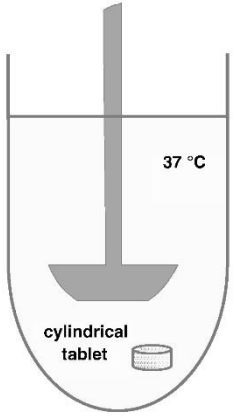
Solid math background is necessary

Simulation: You need to write programs to simulate the performance of Algorithm A &B.

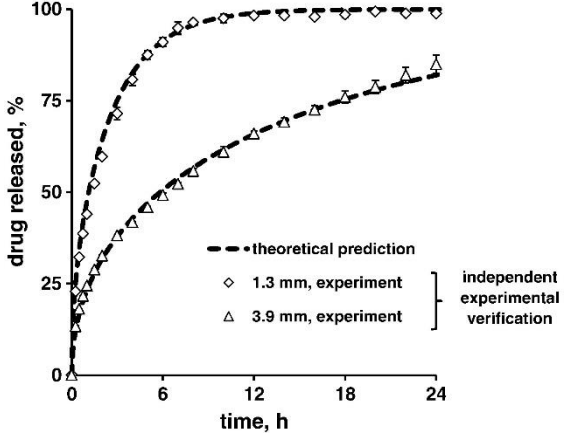
We usually use Matlab.


15
10
5
0
−5
0
5
10
15
20
25
Average Delay (packets)
Average Power P_0 (dB)
equivalent rate constraint
approximated−MDP
Lyapunov
brute−force MDP
equivalent rate constraint
Lyapunov
approximated−MDP
brute−force MDP

- Modeling of diffusion controlled drug delivery



Theoretically predicted (dotted curves) and experimentally verified (symbols) impact of the height of Kollidon SR-based tablets on diprophylline release

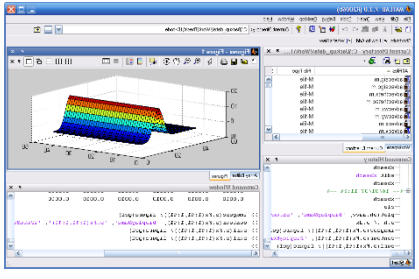


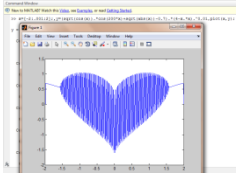
J. Siepmann, Journal of Controlled Release, 2012

Matlab Tutorial

What's Matlab

- MATLAB (MATrix LABoratory)
  - For technical computing, especially for matrix calculation
  - Easy-to-use, save your efforts in coding
- It has its own language
  - Everything in Matlab is **matrix**
  - Interpreted language





- It has its own working environment
- It has a vast collection of computational algorithm in library
- It can interact with other program language, like C and Fortran

Poll(Anonymous)

1 answer(s) at most

设置

Have you ever used Matlab?

A Yes, frequently. I am familiar with Matlab

B Yes, but not familiar with it

C Seldom, almost forget how to use it

D Never. What is Matlab?

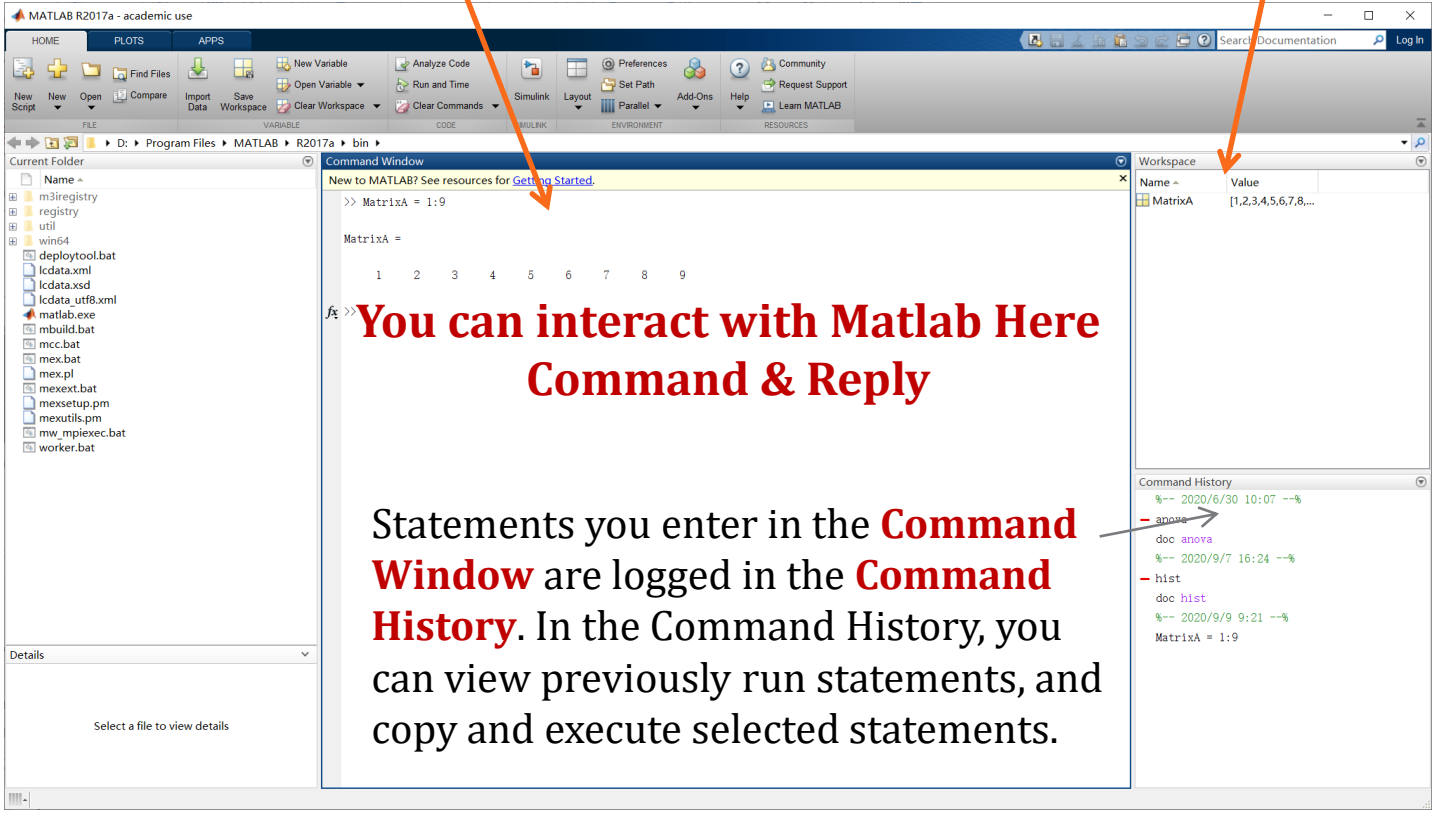
E others

提交

Matlab Environment

Type your Matlab statement in **Command Window**

All currently defined variables will be stored listed out in the **Workspace**.



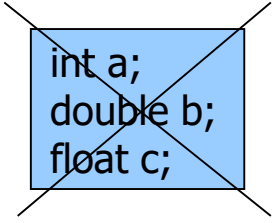
Statements you enter in the **Command Window** are logged in the **Command History**. In the Command History, you can view previously run statements, and copy and execute selected statements.

Language Overview

- Variable
  - Every variable is a matrix
- Operation
  - Matrix-based operation
- Flow control
  - If, for, while, break, …
  - Like C language
- Function

How to Define Variables?

- No need to claim types. i.e.,



You can give value of any type to a variable wherever you want.

- Example:
  - a=5
  - a='hello world'
  - a=5.5
  - b=10+1i
  - a=a*b
  - a=a+b
- Latest value of variable will be stored in the workspace
- Semicolon (;) will stop the interaction with command window

Every Variable is a Matrix

- A vector `x = [1 2 5 1] or x = [1,2,5,1]` Row Vector

```
x =
     1     2     5     1
```

- A matrix `x = [1 2 3; 5 1 4; 3 2 -1]`

```
x =
     1     2     3
     5     1     4
     3     2    -1
```

Matrix

- Transpose `y = x'`

```
y =
     1
     2
     5
     1
```

Column Vector

Conjugate transpose

Long Vector & Matrix

- **`t =1:10`**

```
t =
     1    2    3    4    5    6    7    8    9    10
```

- **`k =2:-0.5:-1`**

```
k =
    2.0000    1.5000    1.0000    0.5000         0   -0.5000   -1.0000
```

- **`B = [1:4; 5:8]`**

```
x =
     1     2     3     4
     5     6     7     8
```

Generating Vectors from functions

zeros(M,N)	M*N matrix of zeros	`x = zeros(1,3)`
		`x =`
		`0 0 0`

ones(M,N)	M*N matrix of ones	`x = ones(1,3)`
		`x =`
		`1 1 1`

rand(M,N)	M*N matrix of uniformly distributed random numbers on (0,1)	`x = rand(1,3)`
		`x =`
		`0.9501 0.2311 0.6068`

Figure out the difference between 'rand' and 'randn'

How?

Concatenation of Matrices

- `x = [1 2], y = [4 5], z=[ 0 0]`

- `A = [x y]`

```
     1    2    4    5
```

- `B = [x ; y]`

```
     1  2
     4  5
```

`C = [x y ; z]`

Error using vertcat

Dimensions of matrices being concatenated are not consistent.

Operators (Arithmetic)

+ addition

- subtraction

***** multiplication

/ division

^ power

‘ complex conjugate transpose

Matrices Operations

- Given A and B:

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9]

A =

     1     2     3
     4     5     6
     7     8     9
```

```
>> B = [3, 5, 2; 5, 2, 8; 3, 6, 9]

B =

     3     5     2
     5     2     8
     3     6     9
```

What about

A^2

A/B

A\B

2*A

A+1j*B

(A+1j*B)'

Addition

```
>> X = A+B

X =

     4     7     5
     9     7    14
    10    14    18
```

Subtraction

```
>> Y = A-B

Y =

    -2    -3     1
    -1     3    -2
     4     2     0
```

Product

```
>> Z = A*B

Z =

    22    27    45
    55    66   102
    88   105   159
```

```
>> Z2 = B*A

Z2 =

    37    47    57
    69    84    99
    90   108   126
```

Transpose

```
>> T = A'

T =

     1     4     7
     2     5     8
     3     6     9
```

Have a try

- Generate matrixes (arrays) :

```
x =
     2    3    4
y1 =
     1    1    1
y2 =
     0    0    0
     1    1    1
y3 =
    1.6   1.4   1.2
```

- Calculate
  - z1 = x * y1'
  - z2 = x' * y1
  - z3 = x * y2'
  - What about
    - z4 = x' * y2
    - z5 = x * y1

Element-Wise Operators

In the previous example, please compare

A^2	**v.s.**	**A.^2**
A*B	**v.s.**	**A.*B**
(A+1j*B)'	**v.s.**	**(A+1j*B).'**

- **.*** element-by-element multiplication
- **./** element-by-element division
- **.^** element-by-element power
- **.'** transpose

Matrix index

- The matrix indices begin from 1 (not 0 (as in C))
- The matrix indices must be **positive integer**

What do these two statements mean?

```
A =

     3     5     3
     6     8     2
     2     7     3
```

```
>> A(6)

ans =

     7
```

```
>> A(3,2)

ans =

     7
```

```
>> A(2,:)

ans =

     6     8     2
```

```
>> A(1:2,2)

ans =

     5
     8
```

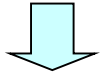
A(-2), A(0)

Error: ??? Subscript indices must either be real positive integers or logicals.

A(4,2)

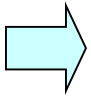
Error: ??? Index exceeds matrix dimensions.

```
A = [1 2 3; 5 1 4; 3 2 1]
   A =
            1    2    3
            5    1    2
            3    4   -1
```



```
x = A(1,:)
x=
    1   2   3
```

```
y = A(3 ,:)
y=
    3  4  -1
```



```
b = x .* y
b=
    3  8 -3
```

```
c = x . / y
c=
  0.33   0.5   -3
```

```
d = x .^2
d=
     1    4    9
```

```
K= x^2
Erorr:
 ??? Error using ==> mpower  Matrix must be square.
B=x*y
Erorr:
??? Error using ==> mtimes Inner matrix dimensions must agree.
```

Useful Commands

- who
- whos
- clear
- clc
- dir
- help/doc

Tutorial: Representing signals

- Handout: 1.1 Tutorial

- Please keep your handout and bring to the next class