MATLA Cheat Sheet — Youhao HU

MATLAB Command line

- **dir**: list files and folders in current folder
- **dir**('*.txt'): find txt files in current folder
- **pwd**: show current folder
- **cd**: change current folder
- **delete**('filename'): delete file
- **movefile**('oldpath/oldname','newpath/newname'): rename or move file
- **what**: list Matlab/simulink files in current folder
- **more on**: display output one screen at a time, space to continue, q to quit
- **more off**: display output continuously
- **warning off**: turn off warning

Find files using Regular Expression

- **dir**('*.txt'): Find the files with extension "txt"
- **dir**('*motor*'): Find the files with "motor" in the file name

Table 1. Some rules of the regular expression

Code	Meaning
*	Match any characters
.	Match any single character
\w	Match a word character (letter/digit/underscore/Chinese character)
\s	Match a single space
\d	Match a single digit
\b	Match a single word boundary (Namely the beginning/end of a word)
^	Match the beginning of a line
$	Match the end of a line

Legend/title rendered in LaTeX

```
h = legend(legend1,legend2);% for instance,
    legend1 = $x_1$
h.Interpreter = 'latex';
h.FontSize    = fontsize;
h.Location    = 'northeast';
h.Orientation = 'horizon';
title('nameoftitle','interpreter','latex')% for
    instance, name = $V_o$
```

Multi lines of title

```
title({'This␣is␣the␣first␣line'; 'This␣is␣the␣
    second␣line'});
```

Save figure

```
saveas(gcf,'filename','png'); % No resolution
    ratio option
print(gcf,'-dpng', '-r300', 'filename'); % 300
    dpi png
```

Data interpolation / Prediction of missing data

Match the data **iL1out** with the simulation time vector **toutSIM**.

- Original data: **(t,iL1out)**
- New data: **(toutSIM,iL1out)**

```
iL1out = interp1(t,iL1out,toutSIM)
```

- Original data: **(t,iL1out)**
- Wanted point: **(t1,iL1)**

```
iL1 = interp1(t,iL1out,t1)
```

The greatest common divisor / The least common multiple

- **gcd**(a,b): The greatest common divisor of a and b
- **lcm**(a,b): The least common multiple of a and b

GA to tune parameters in Simulink

```
clc,clear,close all

SimulinkModel = 'DCMotorPID'; % Simulink model
    name, case-insensitive.
open(SimulinkModel);
fun = @GATestFun;
nvars = 4;
lb = [0;0;0;20000];
ub = [.1;.1;.1;40000];
MaxGen  = 1;
PopSize = 10; % Large size brings better result,
    but takes more time.
options = optimoptions('ga', 'PopulationSize',
    PopSize, 'MaxGenerations', MaxGen,
    'Display', 'iter');
[x,fval,exitflag] = ga(fun, nvars, [], [], [],
    [], lb, ub, [], options);


%-----------------------------------------
%--- Display the parameters tuned by GA ---
%-----------------------------------------

kp = x(1)
ki = x(2)
kd = x(3)
fN = x(4)


%-----------------------------------------
%--- Show the result ---
%-----------------------------------------

hws = get_param(SimulinkModel,'modelworkspace');
hws.assignin('kp',kp);
hws.assignin('ki',ki);
hws.assignin('kd',kd);
hws.assignin('fN',fN);

simout = sim(SimulinkModel);

t          = simout.simout.Time;
SpeedError = simout.simout.Data;
plot(t,SpeedError,'LineWidth',1.5)
title('$Tracking␣Error$','interpreter','latex')
grid on

%-----------------------------------------------
%--- The cost function to evaluate the error ---
%-----------------------------------------------

function cost = GATestFun(inputpara)

SimulinkModel = 'DCMotorPID';
kp = inputpara(1);
ki = inputpara(2);
kd = inputpara(3);
fN = inputpara(4);

hws = get_param(SimulinkModel, 'modelworkspace');
hws.assignin('kp',kp)
hws.assignin('ki',ki)
hws.assignin('kd',kd)
hws.assignin('fN',fN)
simout = sim(SimulinkModel);
```

```
cost    = rms(simout);
end
```

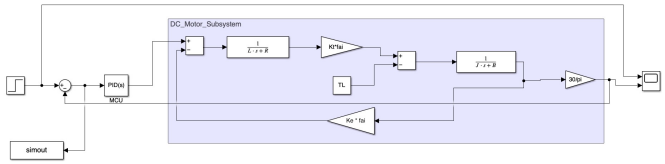

Figure 1. Simulink Model

Table 1. Parameters in the Simulink Model

Parameters	Values	Parameters	Values
Ke	0.1	**fai**	0.1
Kt	**Ke** * 30 / pi	**J**	0.001
L	0.005	**B**	0.01
R	0.1	**TL**	0

Resample the data

Suppose there is a data set (t, V_{RL}) obtained from SIMULINK, we want to resample it to reduce the data size:

```
told    = simuout.out.simout.Time;
VRLold  = simuout.out.simout.Data(:,1);

dtold   = mean(diff(t)); % Original sampling time
dtnew   = .2e-3;         % New even sampling time
Q       = round(dtnew/dtold); % Resample rate
tnew    = t(1):dtnew:length(VRL);
VRLnew  =
    resample(simuout.out.simout.Data(:,1),1,Q);
```

Solve function

```
syms x
eq = x^2 == 1;
solve(eq,x)
```

Solve function numerically

```
syms x
eq = sin(x)^2 == 1;
vpasolve(eq,x,5) % 5 digits, default is 32
    digitss
```

Symbol value to numemrical value

```
syms x
solSym = solve(x^2 == 1,x)
solNum = vpa(solSym,5) % 5 digits, default is 32
    digits
```

Subsitute value into symbolic expression

```
syms x
subs(x^2,x,2)
```

Symbolic to double

```
syms x
SymVar = int(x^2,x,0,1);
VpaVar = vpa(SymVar,5); % 5 digits, default is
    32 digits
DblVar = double(VpaVar);
```

Polynomial fit

```
x = 0:0.1:.5;
y = sin(x);
p = polyfit(x,y,1); % 1st order polynomial fit

plot(x,y)
hold on
plot(x,p(1) * x + p(2))
```

Troubleshooting

Q: SIMULINK does not work properly even with identical parameters.

A: Check the workspace priority, making sure variables are well defined and no other higher priority variables with the same name.

- **Higher priority:** Model workspace.
- **Lower priority:** Base workspace.