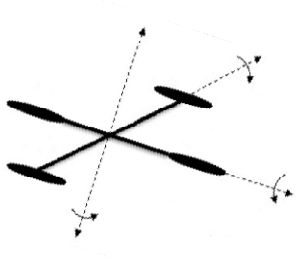
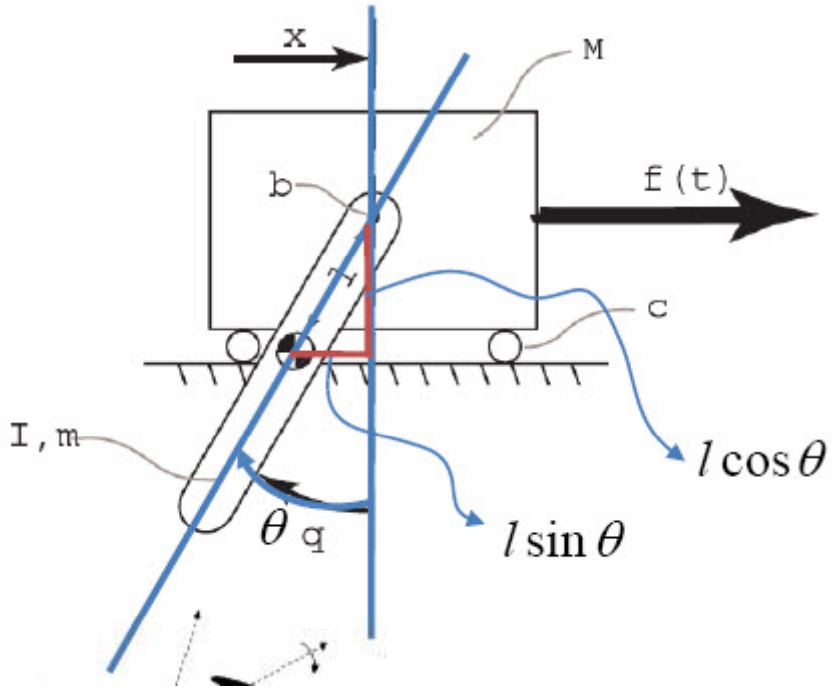
# M A T L A B

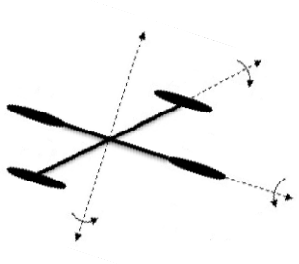
## 동역학을 이용한 시뮬레이션



- Cart type Pendulum 동역학




$$(M+m)\ddot{x}+c\dot{x}-ml\ddot{\theta}=f(t)$$
$$(ml^2+I)\ddot{\theta}-ml\ddot{x}+mgl\theta+f_\theta\dot{\theta}=0$$

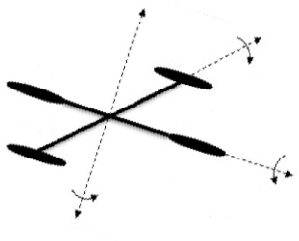


$$(M+m)\ddot{x}+c\dot{x}-ml\ddot{\theta}=f(t)$$
$$(ml^2+I)\ddot{\theta}-ml\ddot{x}+mgl\theta+f_\theta\dot{\theta}=0$$

$$\begin{bmatrix} M+m & -ml \\ -ml & ml^2+I \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}=\begin{bmatrix} -c\dot{x}+F \\ -mgl\theta-f_\theta \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}=\begin{bmatrix} M+m & -ml \\ -ml & ml^2+I \end{bmatrix}^{-1}\begin{bmatrix} -c\dot{x}+F \\ -mgl\theta-f_\theta \end{bmatrix}$$

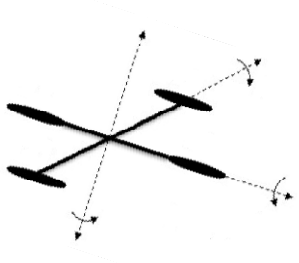
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}=\frac{1}{(M+m)(ml^2+I)+m^2l^2}\begin{bmatrix} ml^2+I & ml \\ ml & M+m \end{bmatrix}\begin{bmatrix} -c\dot{x}+F \\ -mgl\theta-f_\theta \end{bmatrix}$$



$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(ml^2+I)+m^2l^2} \begin{bmatrix} ml^2+I & ml \\ ml & M+m \end{bmatrix} \begin{bmatrix} -c\dot{x}+F \\ -mgl\theta - f_\theta \dot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(ml^2+I)+m^2l^2} \begin{bmatrix} (ml^2+I)(-c\dot{x}+F)+ml(-mgl\theta - f_\theta \dot{\theta}) \\ ml(-c\dot{x}+F)+(M+m)(-mgl\theta - f_\theta \dot{\theta}) \end{bmatrix}$$

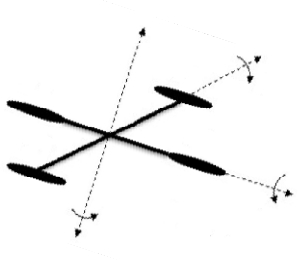
$$\boxed{\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(ml^2+I)+m^2l^2} \begin{bmatrix} -c(ml^2+I)\dot{x} - m^2l^2g\theta - mlf_\theta \dot{\theta} + (ml^2+I)F \\ -cml\dot{x} - mgl(M+m)\theta - (M+m) f_\theta \dot{\theta} + mlF \end{bmatrix}}$$



$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{(M+m)(ml^2+I)+m^2l^2} \begin{bmatrix} -c(ml^2+I)\dot{x} - m^2l^2g\theta - mlf_\theta\dot{\theta} + (ml^2+I)F \\ -cml\dot{x} - mgl(M+m)\theta - (M+m)f_\theta\dot{\theta} + mlF \end{bmatrix}$$

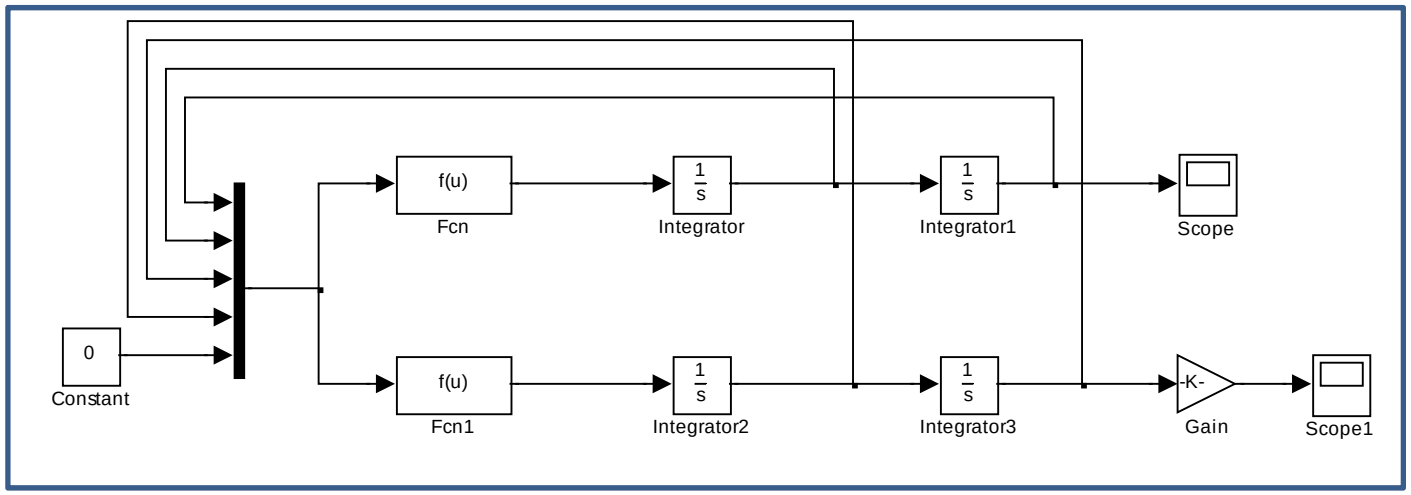
$$M=5,\ m=1,\ l=1,\ g=9.8,\ c=0.01,\ f_\theta=0.01,\ I=0.02$$

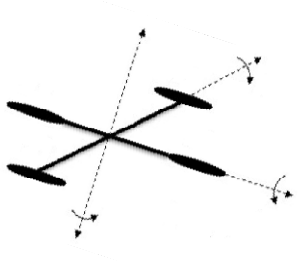
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{6.012} \begin{bmatrix} -0.0102\dot{x} - 9.8\theta - 0.01\dot{\theta} + 1.02F \\ -0.01\dot{x} - 58.8\theta - 0.06\dot{\theta} + F \end{bmatrix}$$

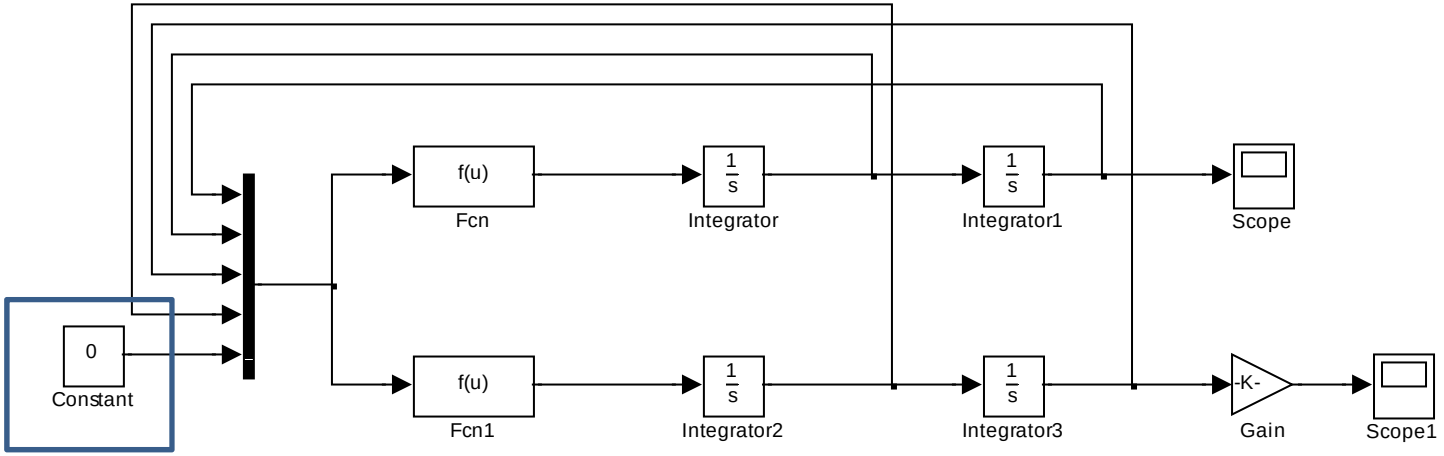


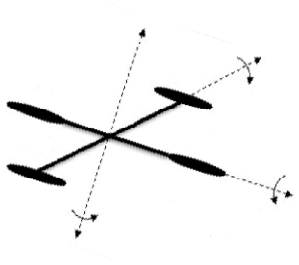
- Cart type Pendulum 시뮬레이션

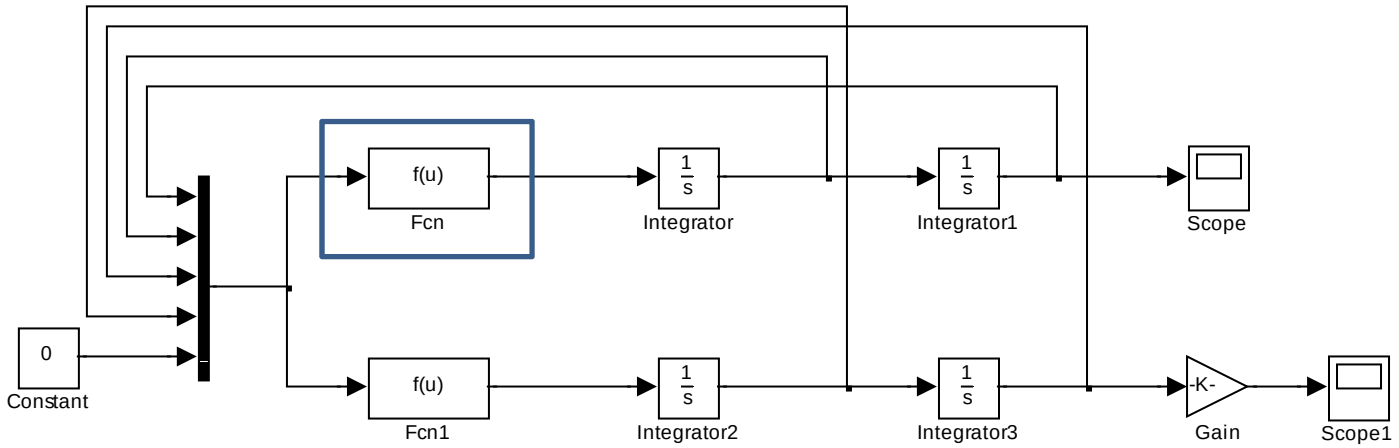
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{6.012} \begin{bmatrix} -0.0102\dot{x} - 9.8\theta - 0.01\dot{\theta} + 1.02F \\ -0.01\dot{x} - 58.8\theta - 0.06\dot{\theta} + F \end{bmatrix}$$

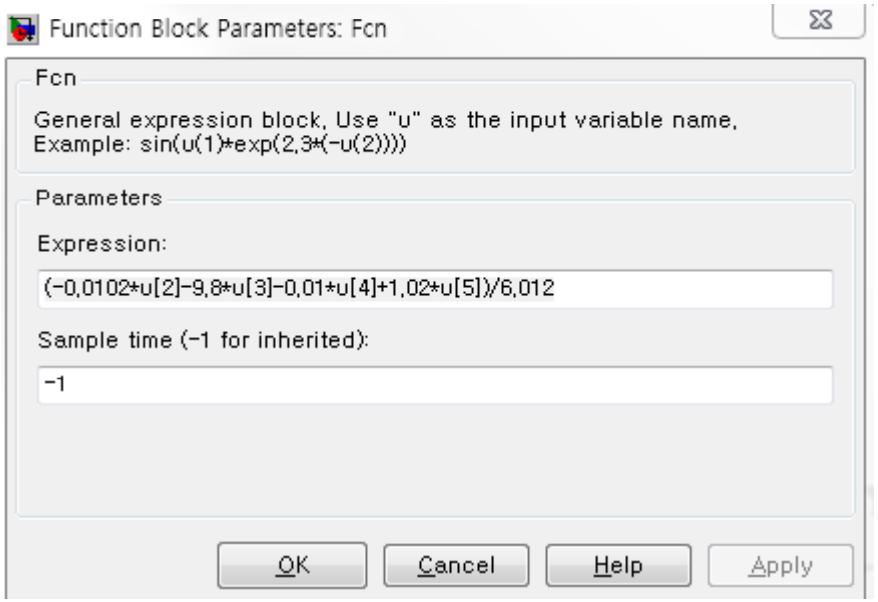


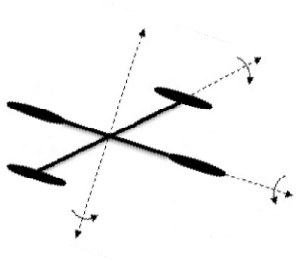



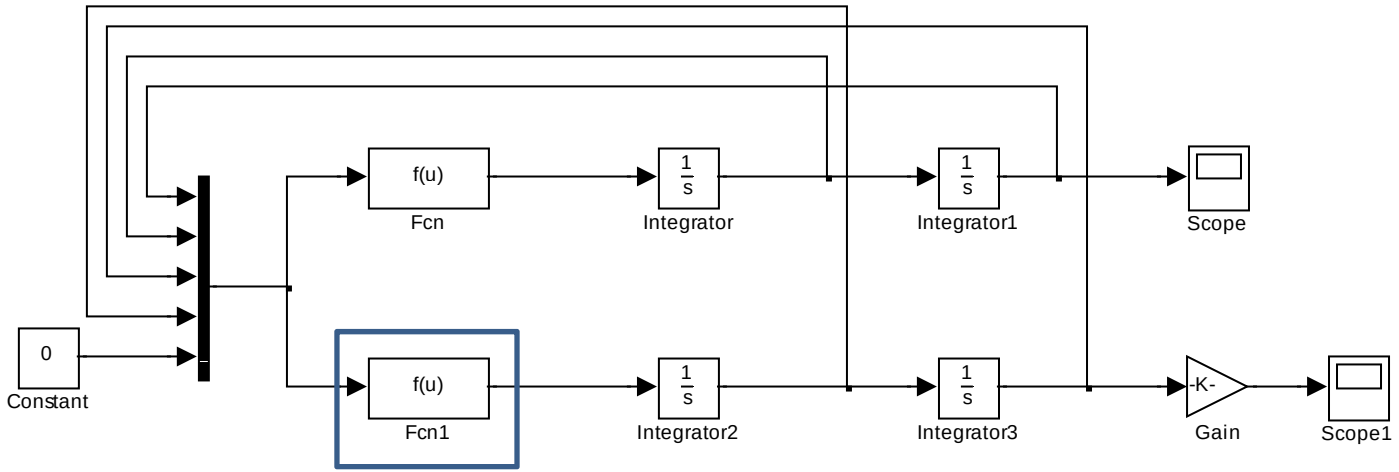

0
Constant
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1

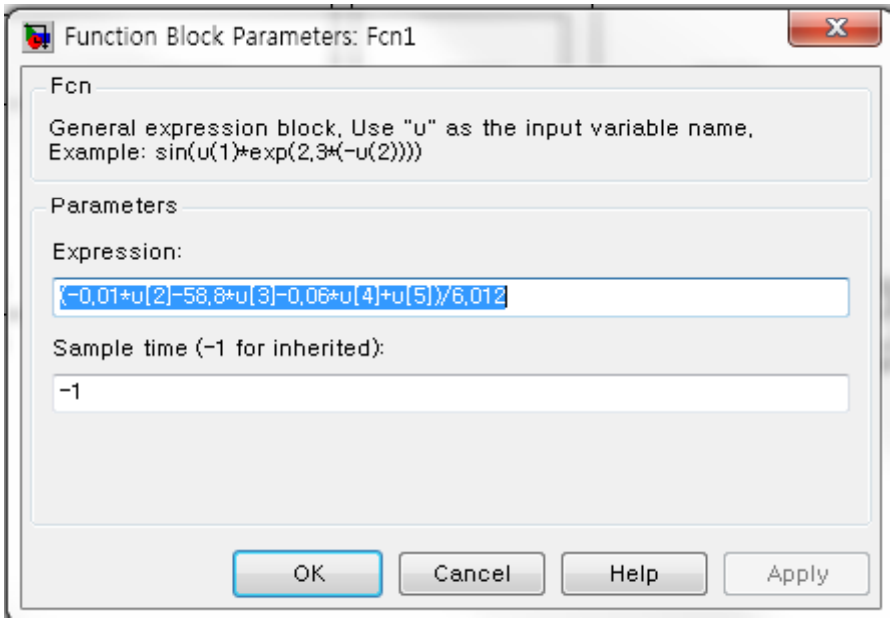


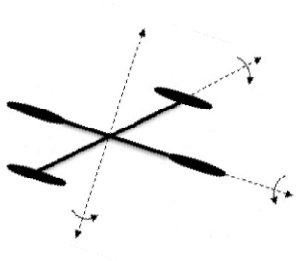

f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1

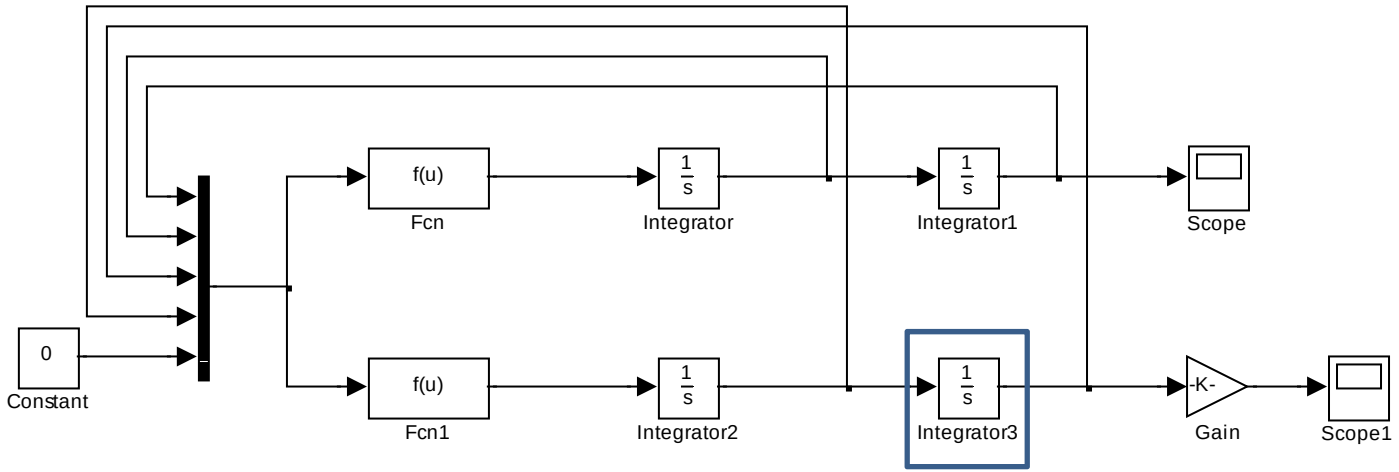

Function Block Parameters: Fcn
Fcn
General expression block, Use "u" as the input variable name,
Example: sin(u(1)*exp(2,3*(-u(2))))
Parameters
Expression:
(-0,0102*u[2]-9,8*u[3]-0,01*u[4]+1,02*u[5])/6,012
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply

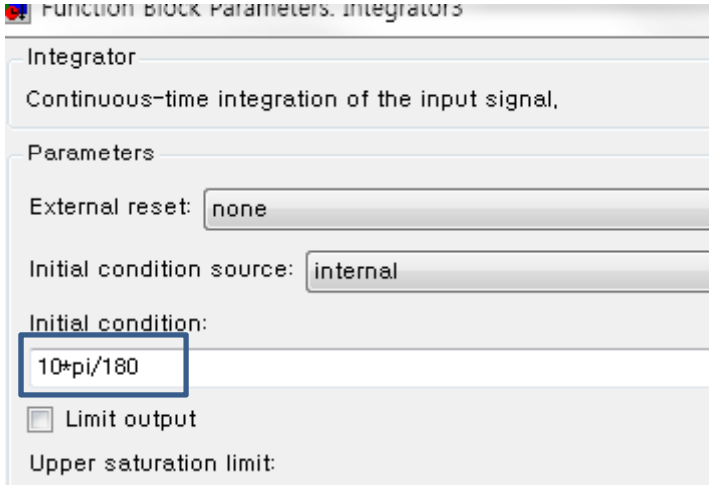


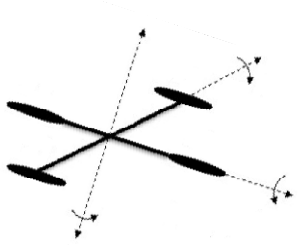

f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1

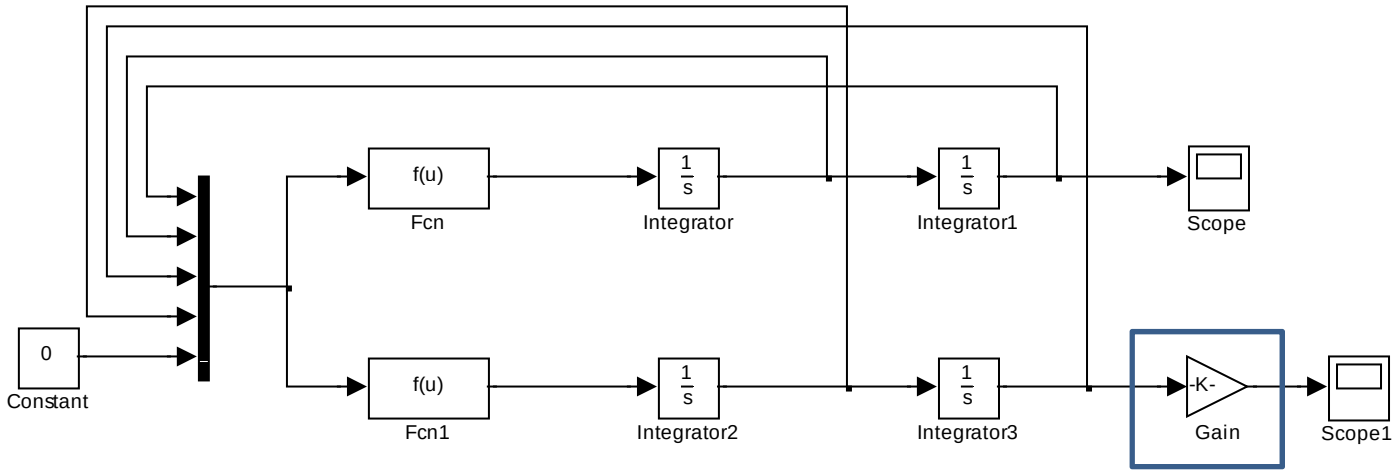

Function Block Parameters: Fcn1
Fcn
General expression block. Use "u" as the input variable name.
Example: sin(u(1)*exp(2.3*(-u(2))))
Parameters
Expression:
(-0.01*u(2)-58.8*u(3)-0.06*u(4)+u(5))/6.012
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply

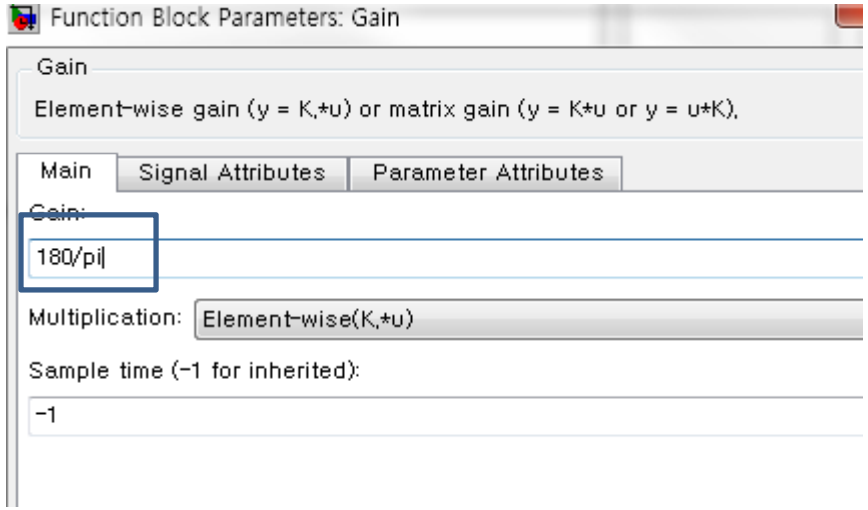


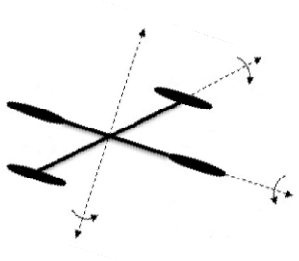

f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1

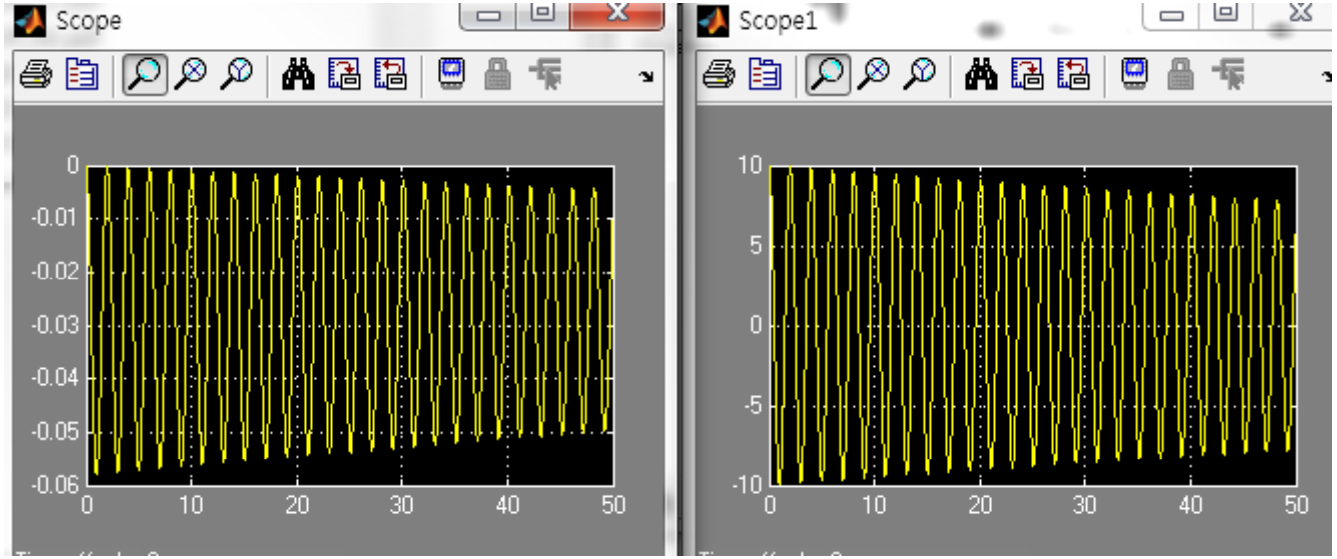

Function Block Parameters: Integrator3
Integrator
Continuous-time integration of the input signal.
Parameters
External reset:
none
Initial condition source:
internal
Initial condition:
10*pi/180
Limit output
Upper saturation limit:

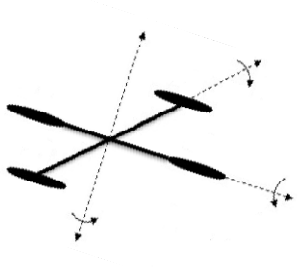



f(u)
Fcn
1/s
Integrator
1/s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1/s
Integrator2
1/s
Integrator3
-K-
Gain
Scope1


Function Block Parameters: Gain
Gain
Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Main
Signal Attributes
Parameter Attributes
Gain:
180/pi
Multiplication:
Element-wise(K.*u)
Sample time (-1 for inherited):
-1

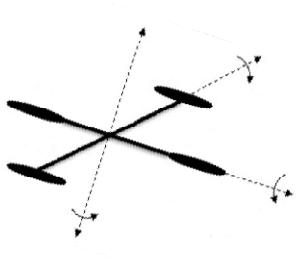


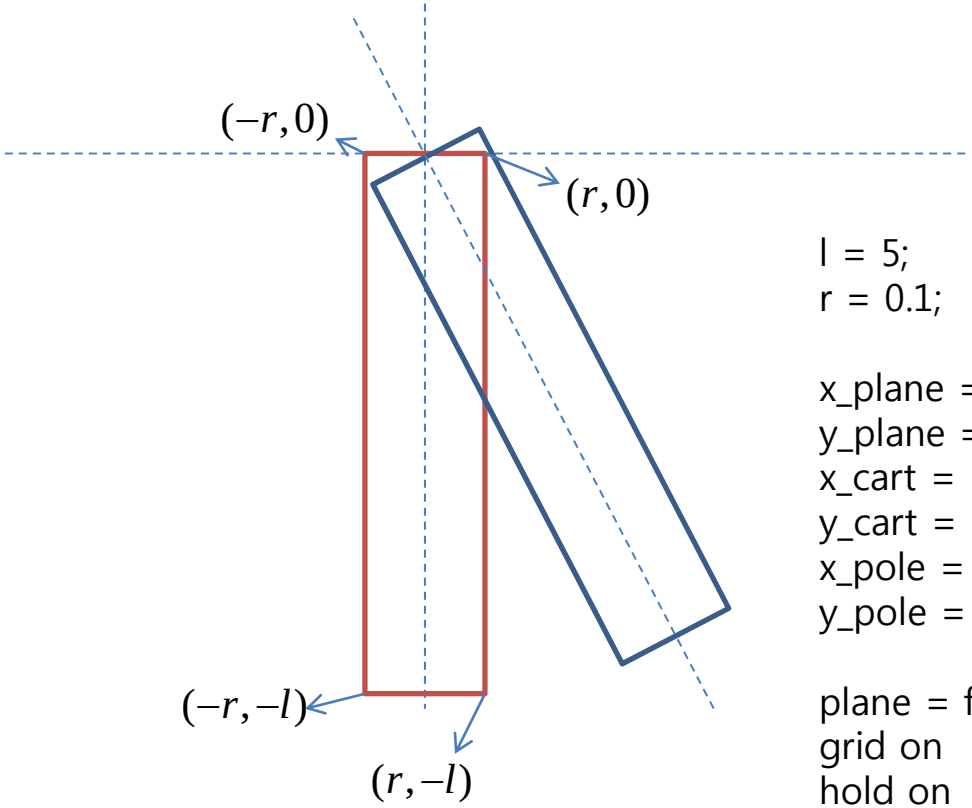

Scope
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
0
10
20
30
40
50
Scope1
10
5
0
-5
-10
0
10
20
30
40
50



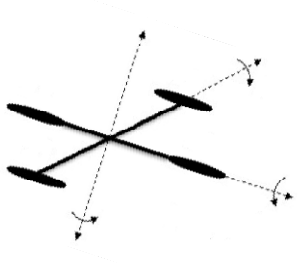
# M A T L A B

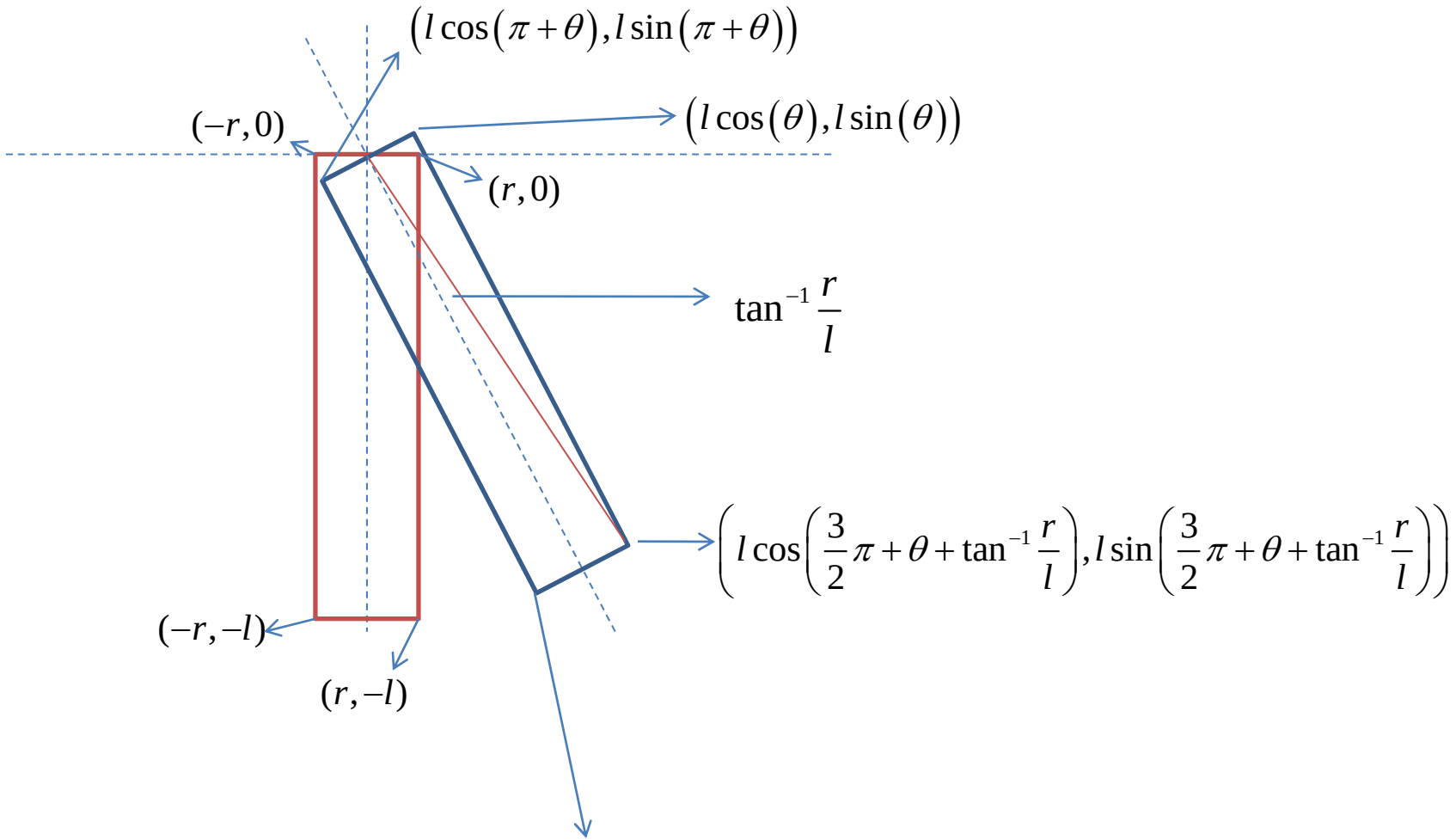
## 애니메이션

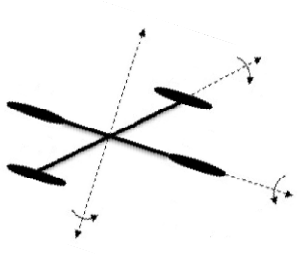




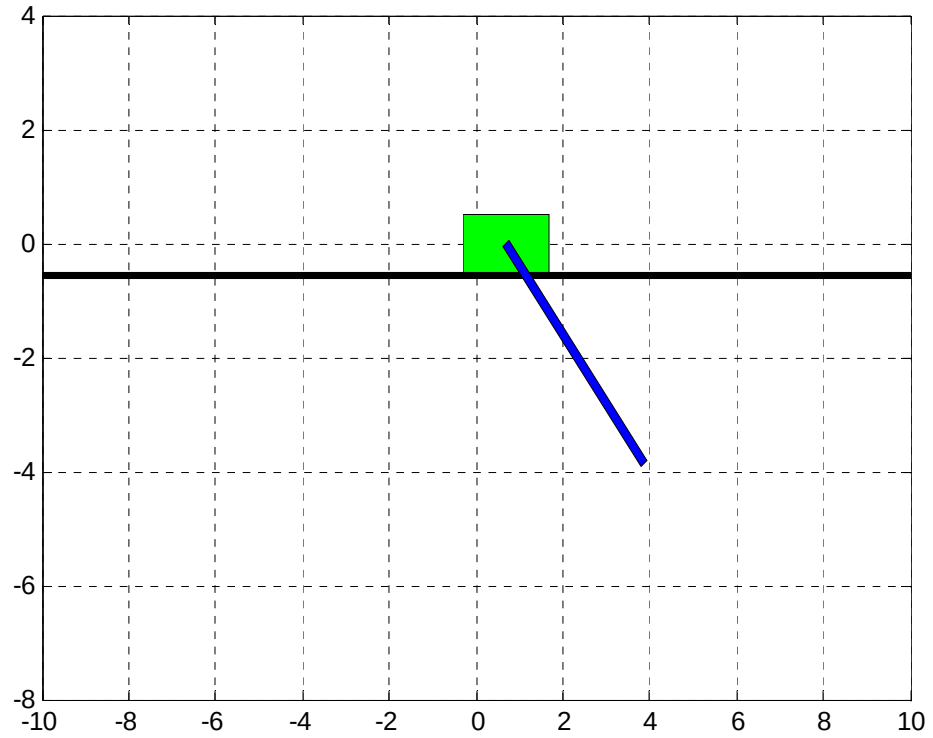

```
l = 5;
r = 0.1;

x_plane = [-10 -10 10 10];
y_plane = [-0.6 -0.5 -0.5 -0.6];
x_cart = [-1 -1 1 1];
y_cart = [-0.5 0.5 0.5 -0.5];
x_pole = [-r -r r r];
y_pole = [-l 0 0 -l];

plane = fill(x_plane, y_plane, 'k');
grid on
hold on
cart = fill(x_cart, y_cart, 'g','EraseMode','normal');
pole = fill(x_pole, y_pole, 'b','EraseMode','normal');
hold off
axis([-10 10 -8 4]);
```




$(l\cos(\pi+\theta), l\sin(\pi+\theta))$
$(-r, 0)$
$(l\cos(\theta), l\sin(\theta))$
$(r, 0)$
$\tan^{-1}\frac{r}{l}$
$\left(l\cos\left(\frac{3}{2}\pi+\theta+\tan^{-1}\frac{r}{l}\right), l\sin\left(\frac{3}{2}\pi+\theta+\tan^{-1}\frac{r}{l}\right)\right)$
$(-r, -l)$
$(r, -l)$
$\left(l\cos\left(\frac{3}{2}\pi+\theta-\tan^{-1}\frac{r}{l}\right), l\sin\left(\frac{3}{2}\pi+\theta-\tan^{-1}\frac{r}{l}\right)\right)$

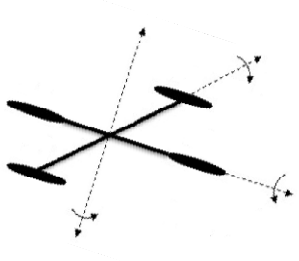


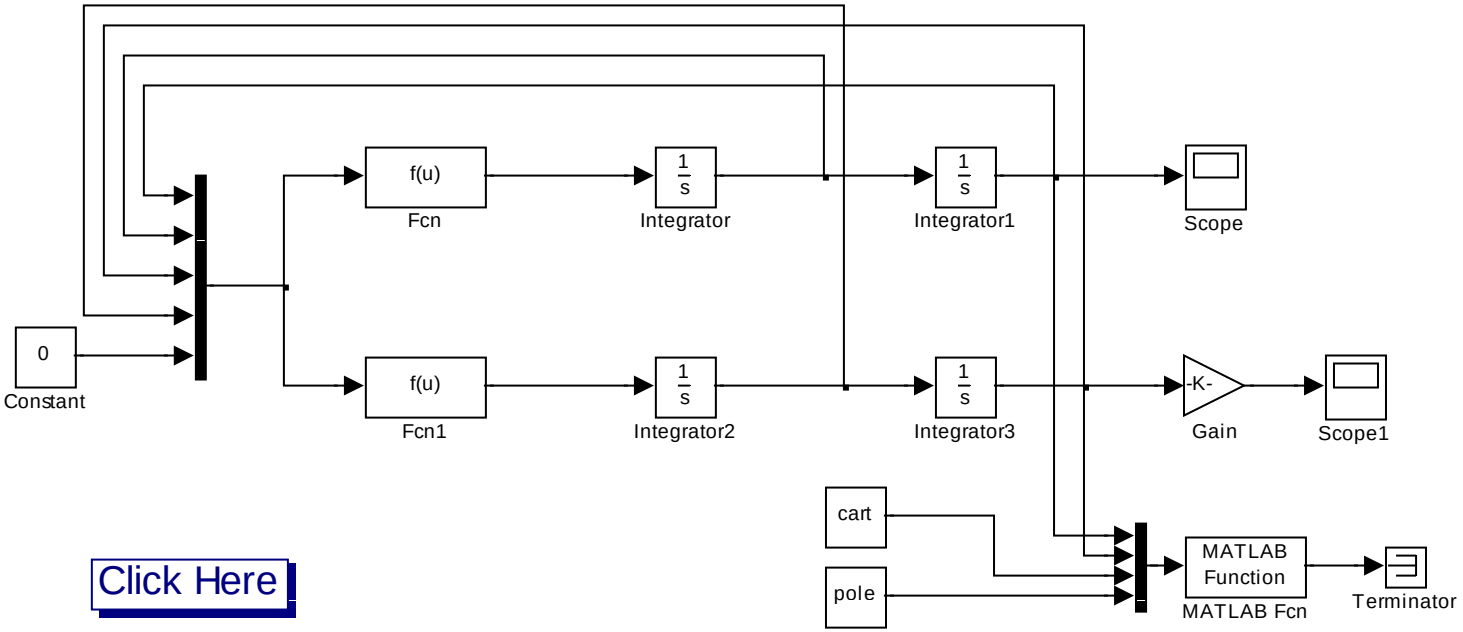
```
l = 5;
r = 0.1;

x_plane = [-10 -10 10 10];
y_plane = [-0.6 -0.5 -0.5 -0.6];
x_cart = [-1 -1 1 1];
y_cart = [-0.5 0.5 0.5 -0.5];
x_pole = [-r -r r r];
y_pole = [-l 0 0 -l];

plane = fill(x_plane, y_plane, 'k');
grid on
hold on
cart = fill(x_cart, y_cart, 'g','EraseMode','normal');
pole = fill(x_pole, y_pole, 'b','EraseMode','normal');
hold off
axis([-10 10 -8 4]);

t = 0:0.01:20*pi/180;

for t = 0:0.01:20*pi/180
    updatedX_cart = x_cart + t;
    temp_x = [l*cos(3/2*pi+t-atan(r/l)) r*cos(pi+t) r*cos(t) l*cos(3/2*pi+t+atan(r/l))];
    temp_y = [l*sin(3/2*pi+t-atan(r/l)) r*sin(pi+t) r*sin(t) l*sin(3/2*pi+t+atan(r/l))];
    updatedX_pole = t + temp_x;
    updatedY_pole = temp_y;
    set(cart, 'Xdata', updatedX_cart);
    set(pole, 'Xdata', updatedX_pole,'Ydata', updatedY_pole);
    drawnow;

    for i=1:5000000
        temp = i;
    end
end
```



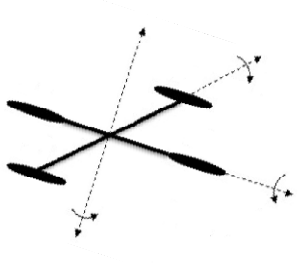
# M A T L A B

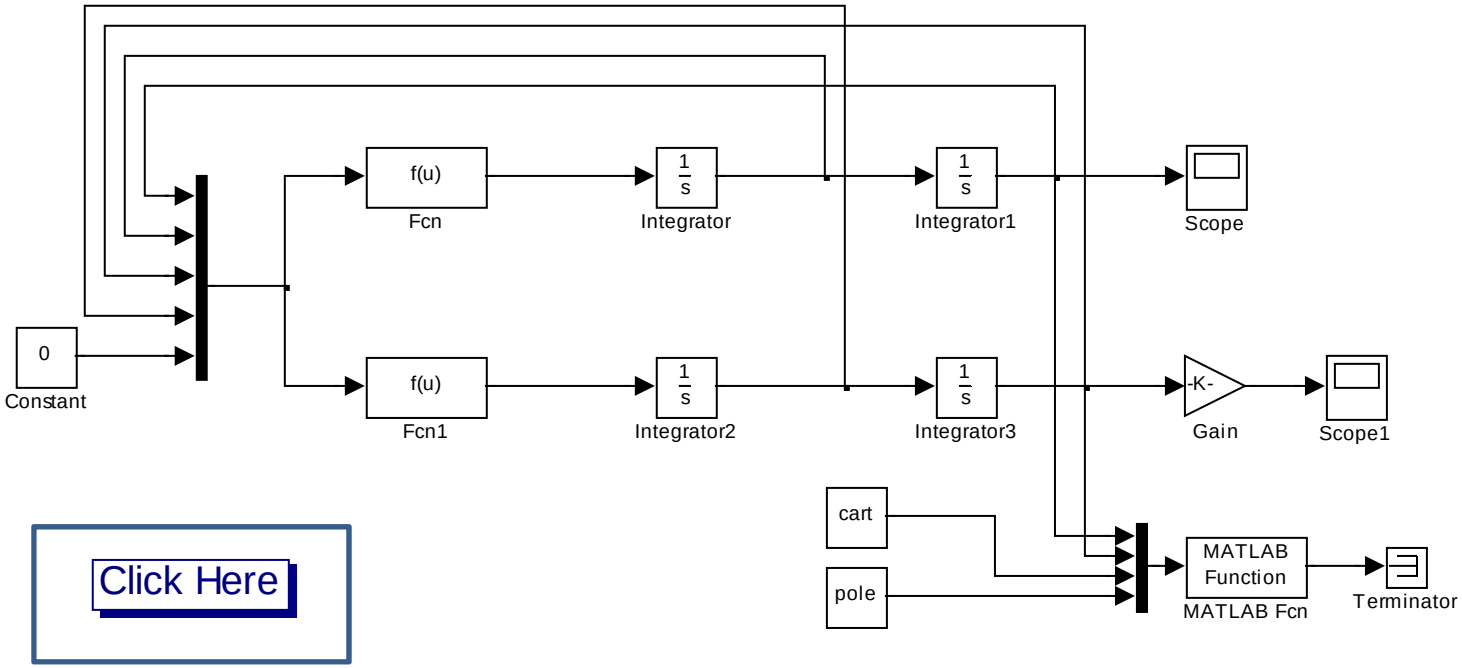
## 애니메이션을 이용한 시뮬레이션



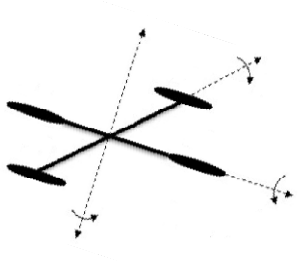

f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

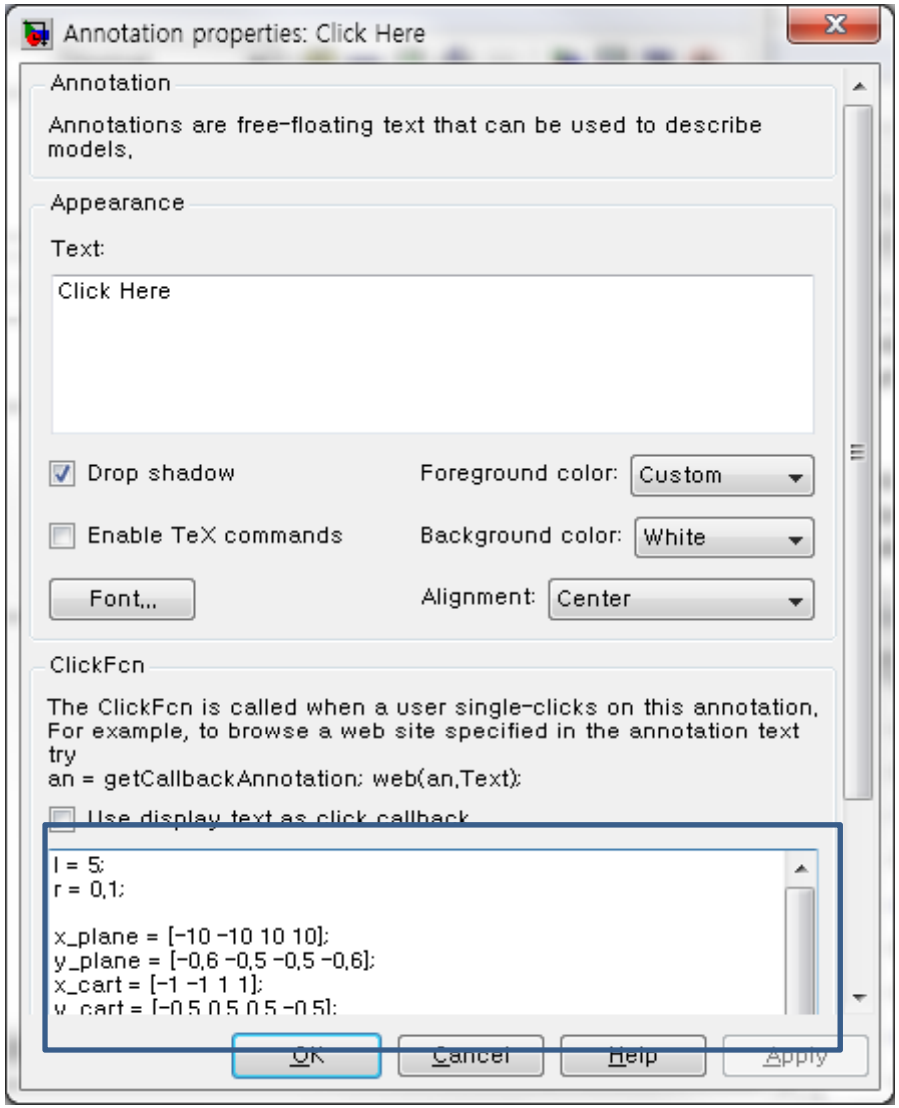
Click Here

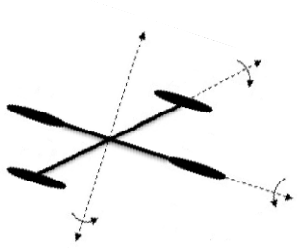



f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

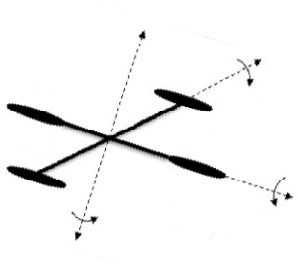
Click Here

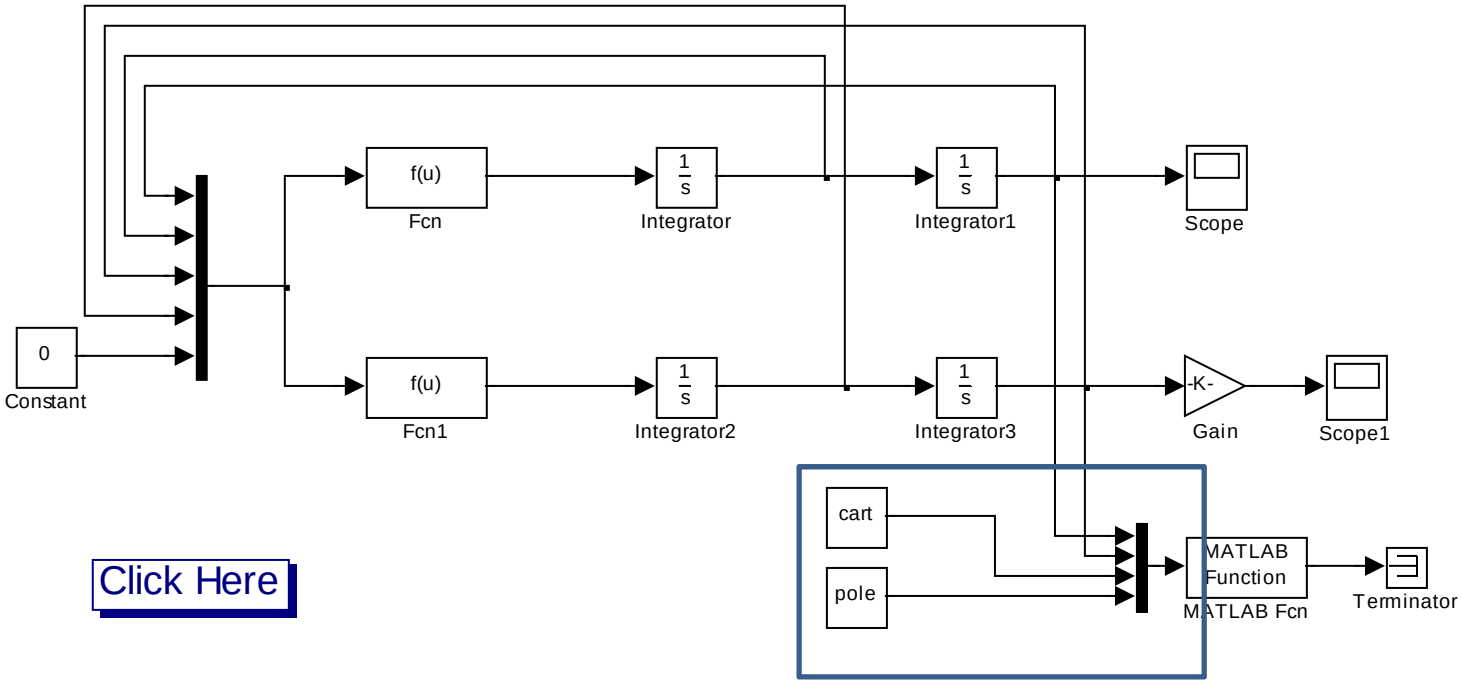



Annotation properties: Click Here
Annotation
Annotations are free-floating text that can be used to describe models.
Appearance
Text:
Click Here
Drop shadow
Foreground color:
Custom
Enable TeX commands
Background color:
White
Font...
Alignment:
Center
ClickFcn
The ClickFcn is called when a user single-clicks on this annotation. For example, to browse a web site specified in the annotation text try
an = getCallbackAnnotation; web(an.Text);
Use display text as click callback
l = 5;
r = 0.1;
x_plane = [-10 -10 10 10];
y_plane = [-0.6 -0.5 -0.5 -0.6];
x_cart = [-1 -1 1 1];
OK
Cancel
Help
Apply

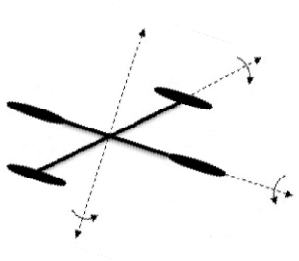


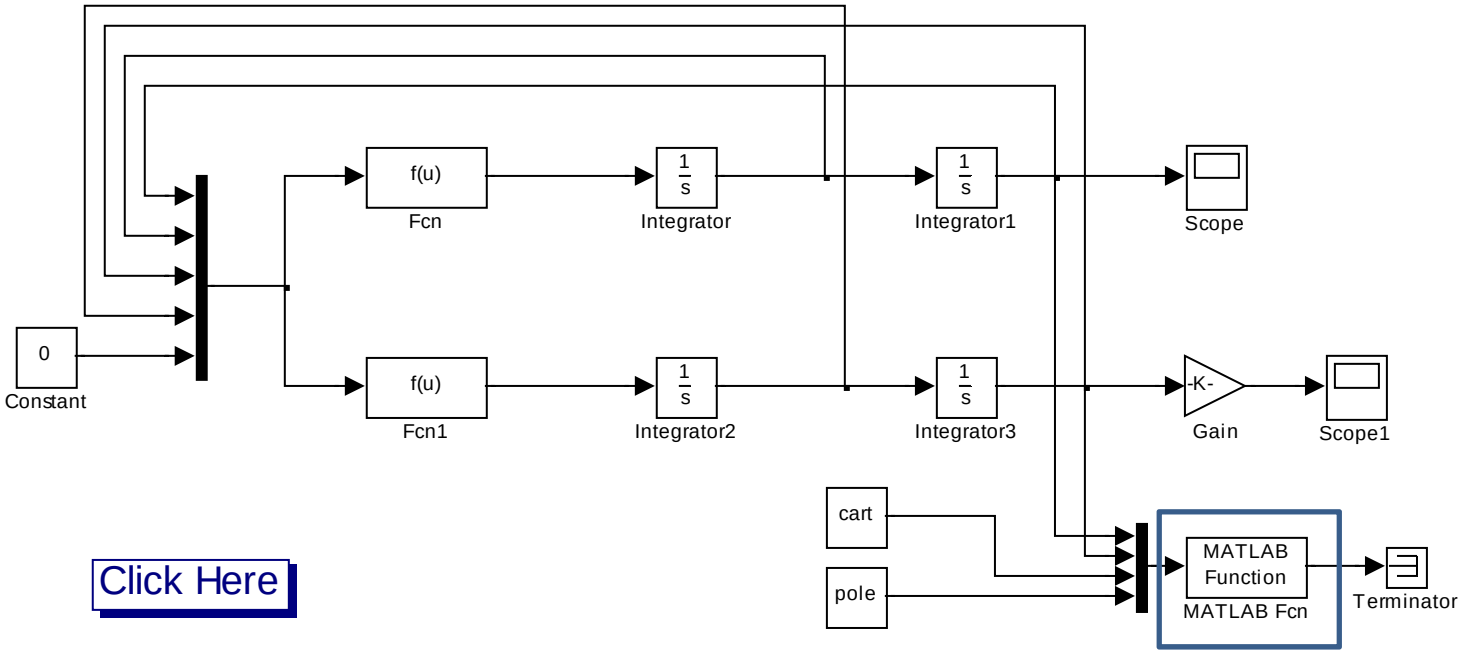
```
l = 5;
r = 0.1;
x_plane = [-10 -10 10 10];
y_plane = [-0.6 -0.5 -0.5 -0.6];
x_cart = [-1 -1 1 1];
y_cart = [-0.5 0.5 0.5 -0.5];
x_pole = [-r -r r r];
y_pole = [-l 0 0 -l];
plane = fill(x_plane, y_plane, 'k');
grid on
hold on
cart = fill(x_cart, y_cart, 'g','EraseMode','normal');
pole = fill(x_pole, y_pole, 'b','EraseMode','normal');
hold off
axis([-10 10 -8 4]);
```



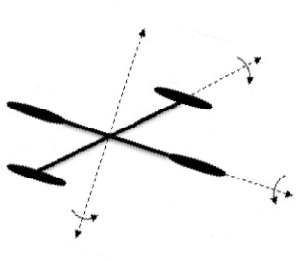

f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

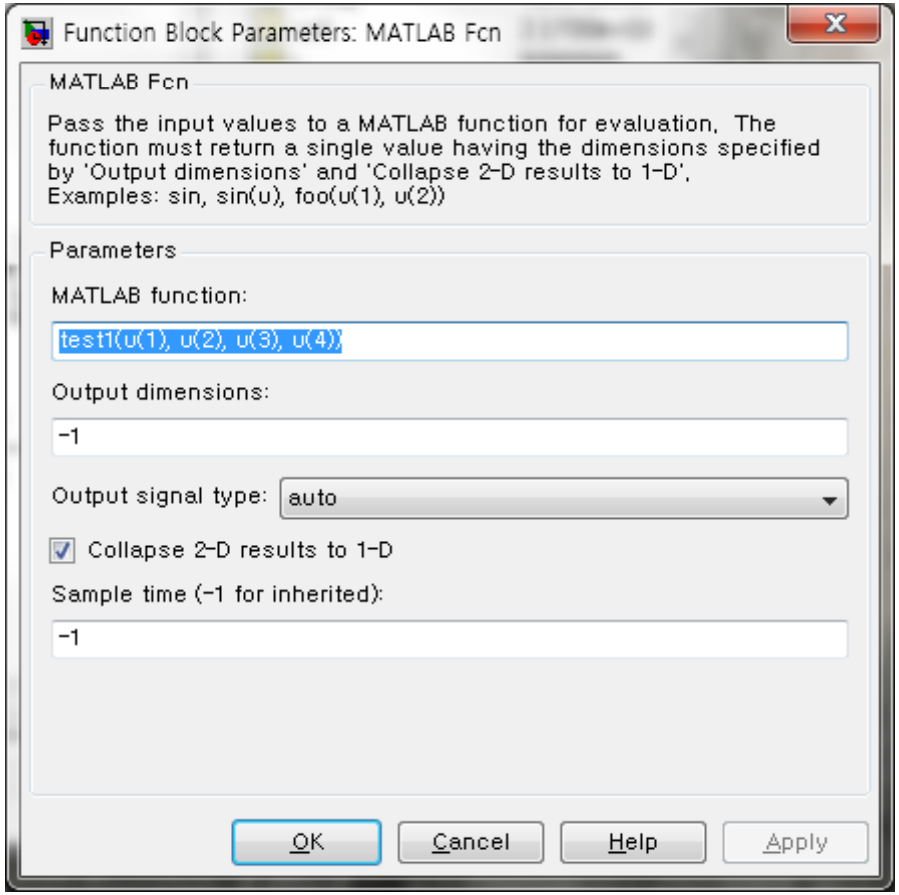
Click Here

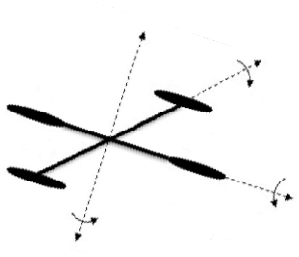



f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
0
Constant
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

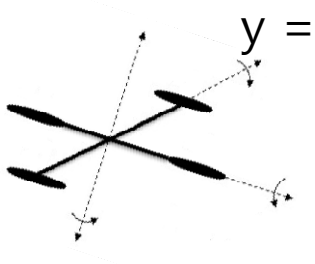
Click Here

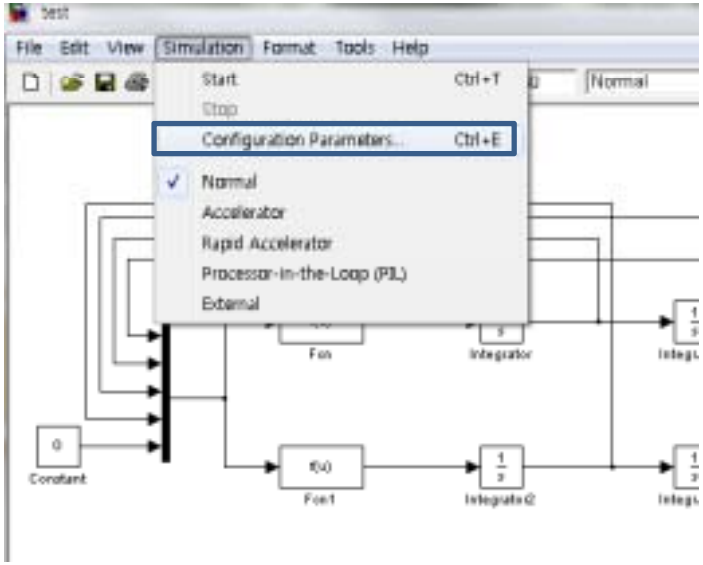


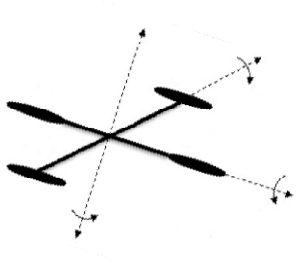

Function Block Parameters: MATLAB Fcn
MATLAB Fcn
Pass the input values to a MATLAB function for evaluation. The function must return a single value having the dimensions specified by 'Output dimensions' and 'Collapse 2-D results to 1-D'.
Examples: sin, sin(u), foo(u(1), u(2))
Parameters
MATLAB function:
test1(u(1), u(2), u(3), u(4))
Output dimensions:
-1
Output signal type: auto
Collapse 2-D results to 1-D
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply

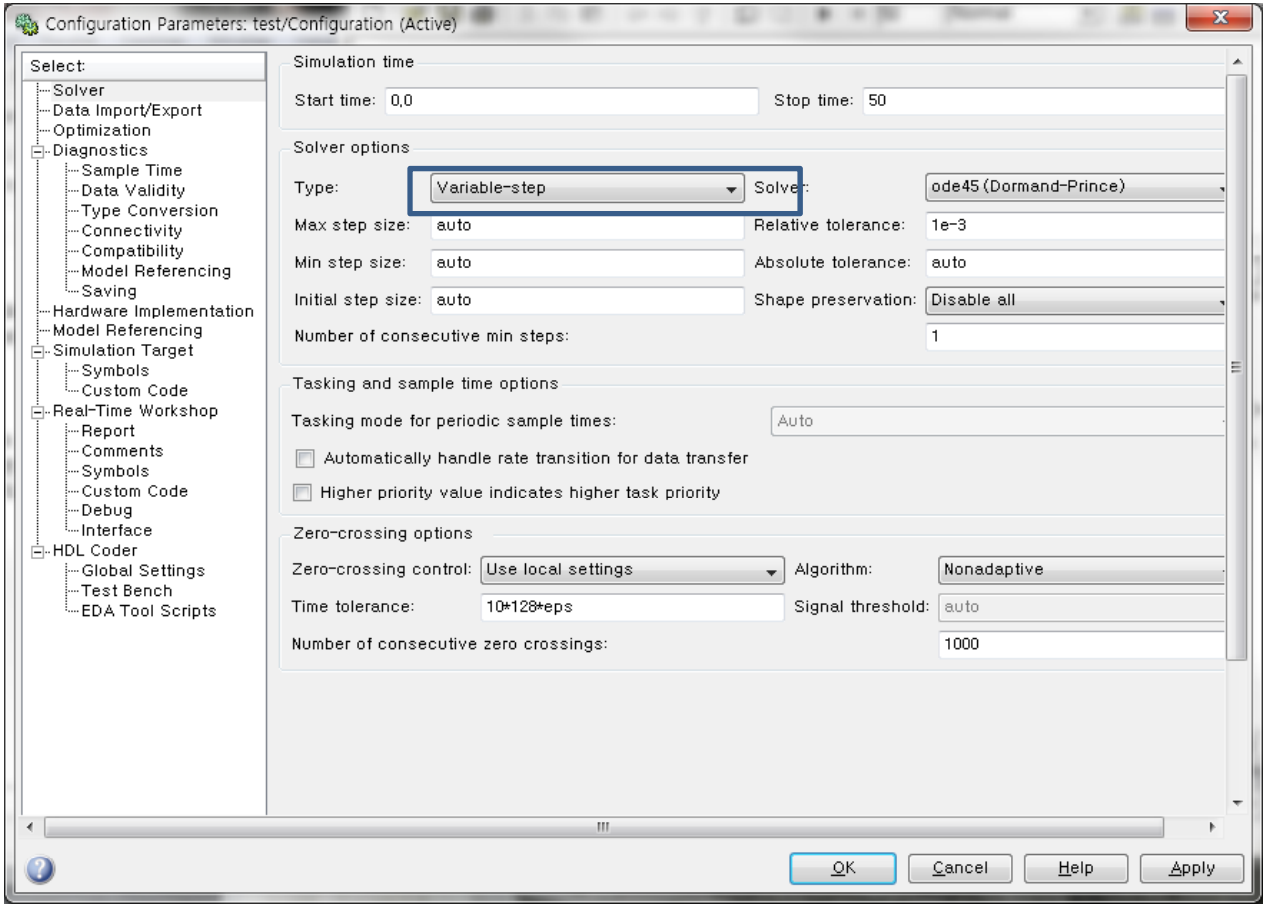


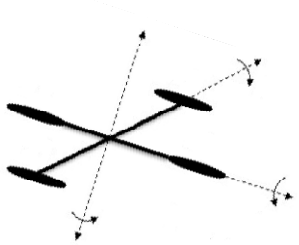
```
function y = test1(pos_cart, theta, cart, pole)

l = 5;
r = 0.1;
x_cart = [-1 -1 1 1];

updatedX_cart = x_cart + pos_cart;
temp_x = [l*cos(3/2*pi+theta-atan(r/l)) r*cos(pi+theta) r*cos(theta)
l*cos(3/2*pi+theta+atan(r/l))];
temp_y = [l*sin(3/2*pi+theta-atan(r/l)) r*sin(pi+theta) r*sin(theta)
l*sin(3/2*pi+theta+atan(r/l))];
updatedX_pole = pos_cart + temp_x;
updatedY_pole = temp_y;
set(cart, 'Xdata', updatedX_cart);
set(pole, 'Xdata', updatedX_pole,'Ydata', updatedY_pole);
drawnow;

y = 1;
```

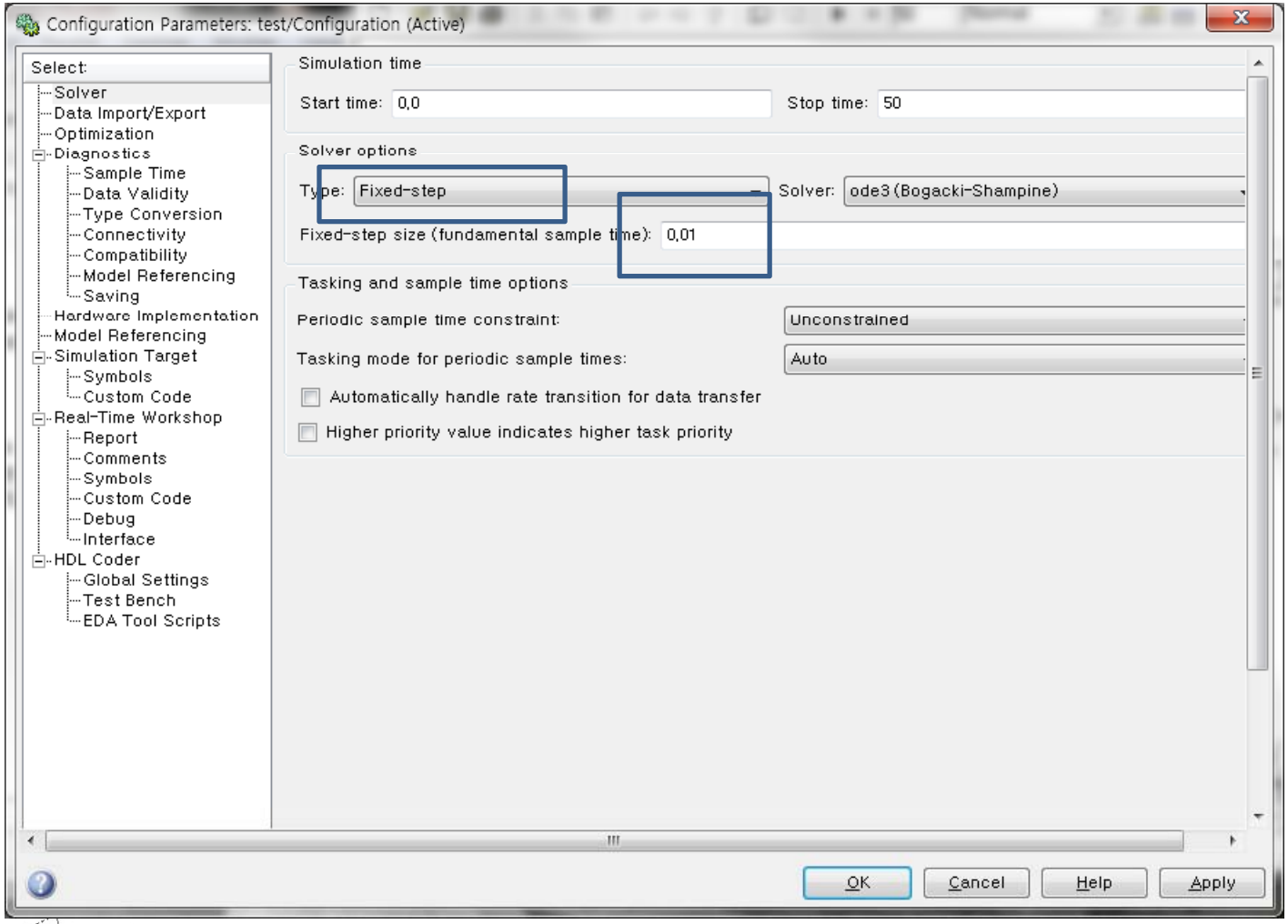


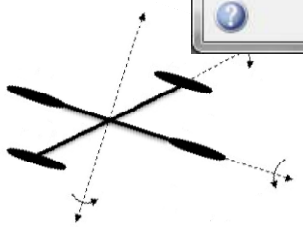

test
File Edit View Simulation Format Tools Help
Normal
Start Ctrl+T
Stop
Configuration Parameters... Ctrl+E
Normal
Accelerator
Rapid Accelerator
Processor-in-the-Loop (PIL)
External
Fcn
Integrator
0
Constant
f(u)
Fcn1
Integrator2




Configuration Parameters: test/Configuration (Active)
Select:
Solver
Data Import/Export
Optimization
Diagnostics
Sample Time
Data Validity
Type Conversion
Connectivity
Compatibility
Model Referencing
Saving
Hardware Implementation
Model Referencing
Simulation Target
Symbols
Custom Code
Real-Time Workshop
Report
Comments
Symbols
Custom Code
Debug
Interface
HDL Coder
Global Settings
Test Bench
EDA Tool Scripts
Simulation time
Start time: 0,0
Stop time: 50
Solver options
Type: Variable-step
Solver: ode45 (Dormand-Prince)
Max step size: auto
Relative tolerance: 1e-3
Min step size: auto
Absolute tolerance: auto
Initial step size: auto
Shape preservation: Disable all
Number of consecutive min steps: 1
Tasking and sample time options
Tasking mode for periodic sample times: Auto
Automatically handle rate transition for data transfer
Higher priority value indicates higher task priority
Zero-crossing options
Zero-crossing control: Use local settings
Algorithm: Nonadaptive
Time tolerance: 10*128*eps
Signal threshold: auto
Number of consecutive zero crossings: 1000
OK
Cancel
Help
Apply

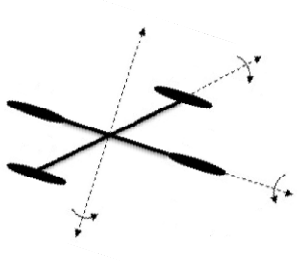



Configuration Parameters: test/Configuration (Active)
Select:
Solver
Data Import/Export
Optimization
Diagnostics
Sample Time
Data Validity
Type Conversion
Connectivity
Compatibility
Model Referencing
Saving
Hardware Implementation
Model Referencing
Simulation Target
Symbols
Custom Code
Real-Time Workshop
Report
Comments
Symbols
Custom Code
Debug
Interface
HDL Coder
Global Settings
Test Bench
EDA Tool Scripts
Simulation time
Start time: 0,0
Stop time: 50
Solver options
Type: Fixed-step
Solver: ode3 (Bogacki-Shampine)
Fixed-step size (fundamental sample time): 0,01
Tasking and sample time options
Periodic sample time constraint: Unconstrained
Tasking mode for periodic sample times: Auto
Automatically handle rate transition for data transfer
Higher priority value indicates higher task priority
OK
Cancel
Help
Apply



# M A T L A B

## LQR 툴을 이용한 제어기 설계

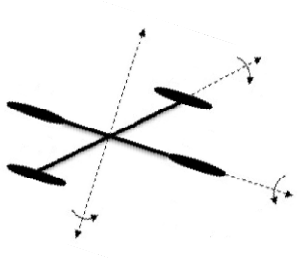


- LQR을 이용한 제어기 설계

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{6.012}\begin{bmatrix} -0.0102\dot{x} - 9.8\theta - 0.01\dot{\theta} + 1.02F \\ -0.01\dot{x} - 58.8\theta - 0.06\dot{\theta} + F \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0017 & -1.6301 & -0.0017 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0017 & -9.7804 & -0.01 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1697 \\ 0 \\ 0.1663 \end{bmatrix} F$$

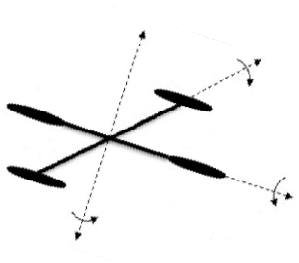
A B



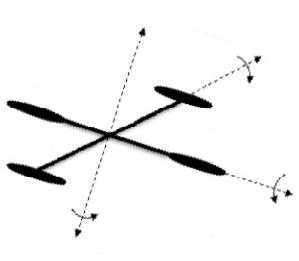
$$A=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & -0.0017 & -1.6301 & -0.0017\\ 0 & 0 & 0 & 1\\ 0 & -0.0017 & -9.7804 & -0.01\end{bmatrix},\ B=\begin{bmatrix}0\\ 0.1697\\ 0\\ 0.1663\end{bmatrix}$$

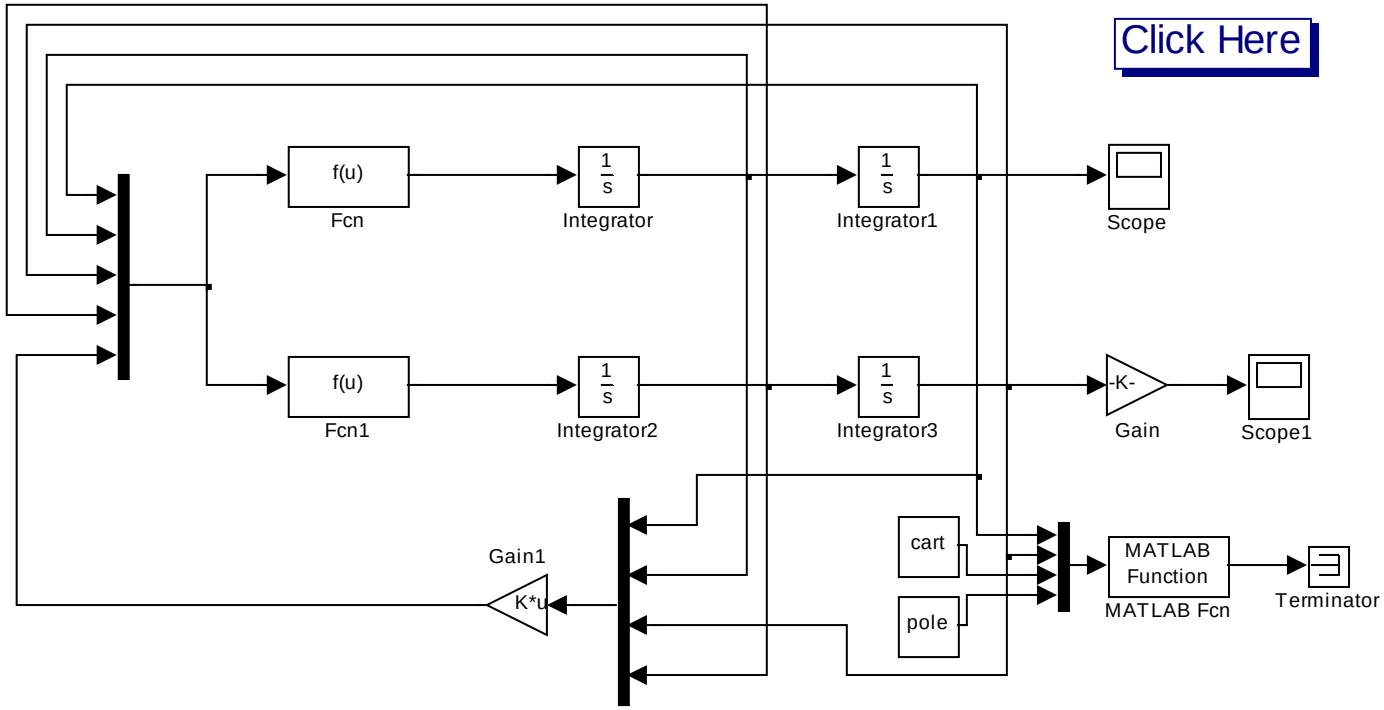
$$Q=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 5 & 0\\ 0 & 0 & 0 & 0\end{bmatrix},\ R=0.01$$

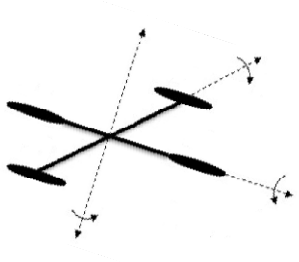
**K = lqr(A,B,Q,R)**

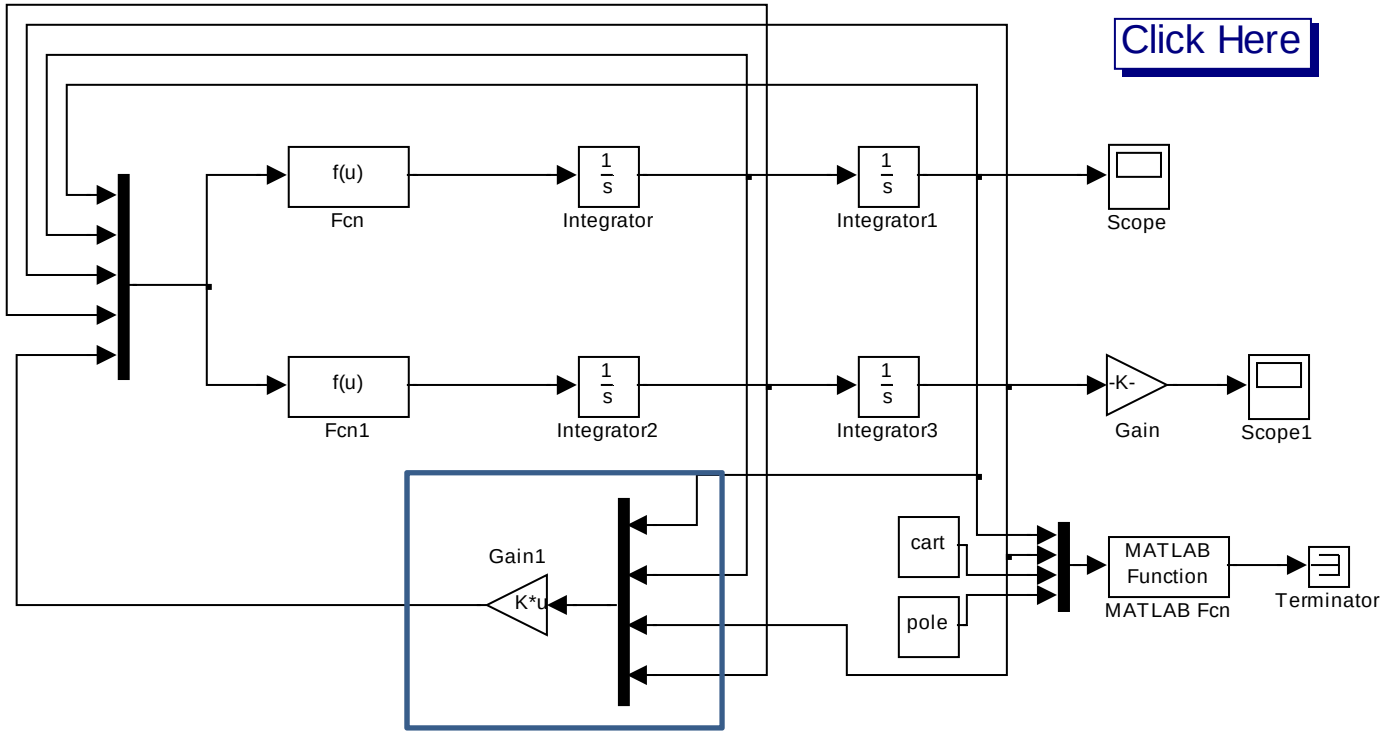


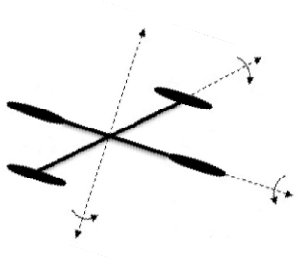
```
>> A = [0 1 0 0; 0 -0.0017 -1.6301 -0.0017; 0 0 0 1; 0 -0.0017 -9.7804 -0.01];
>> B = [0 0.1697 0 0.1663]';
>> Q = [1 0 0 0; 0 0 0 0; 0 0 5 0; 0 0 0 0];
>> R = 0.01;
>> K=lqr(A,B,Q,R)

K =

   10.0000   13.0808   12.7367    3.2320
```

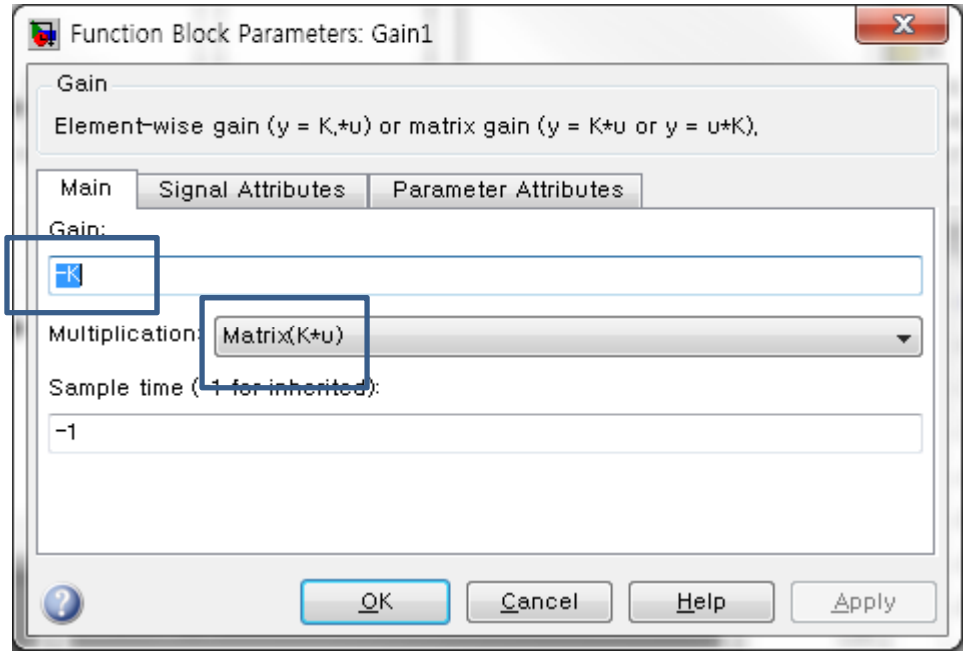


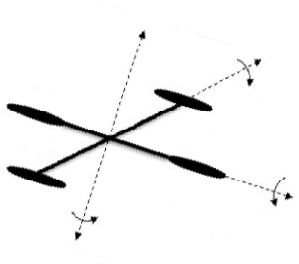

Click Here
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
Gain1
K*u
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

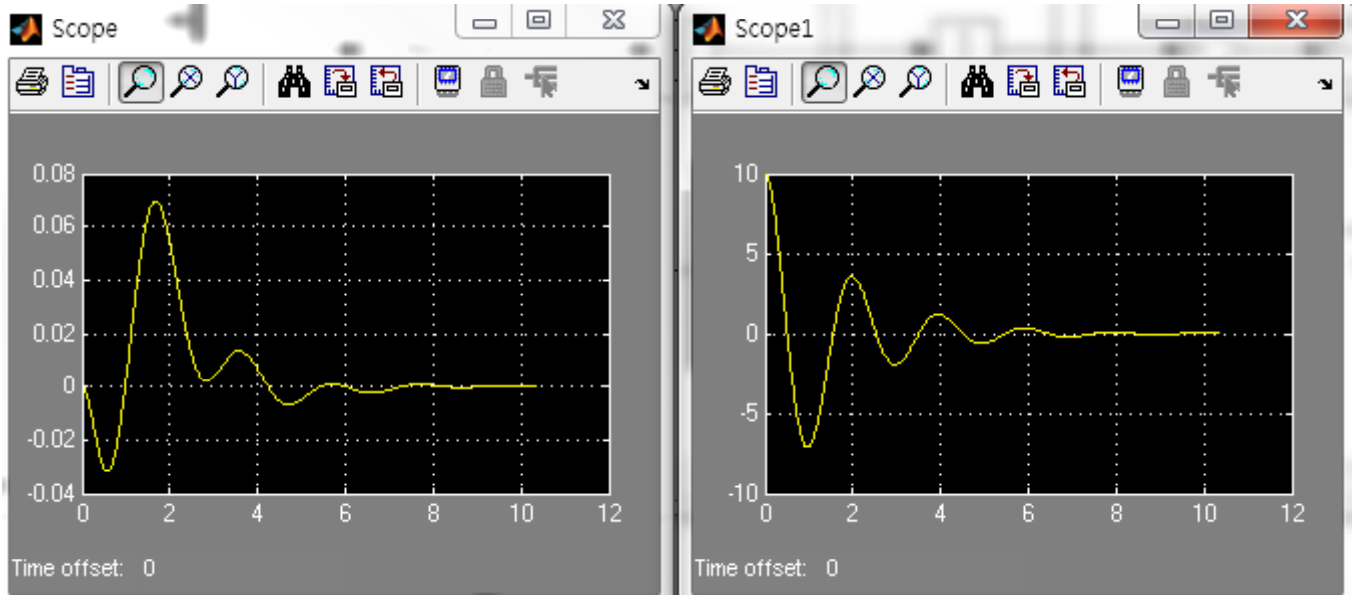


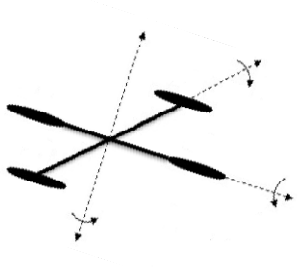

Click Here
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
Gain1
K*u
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

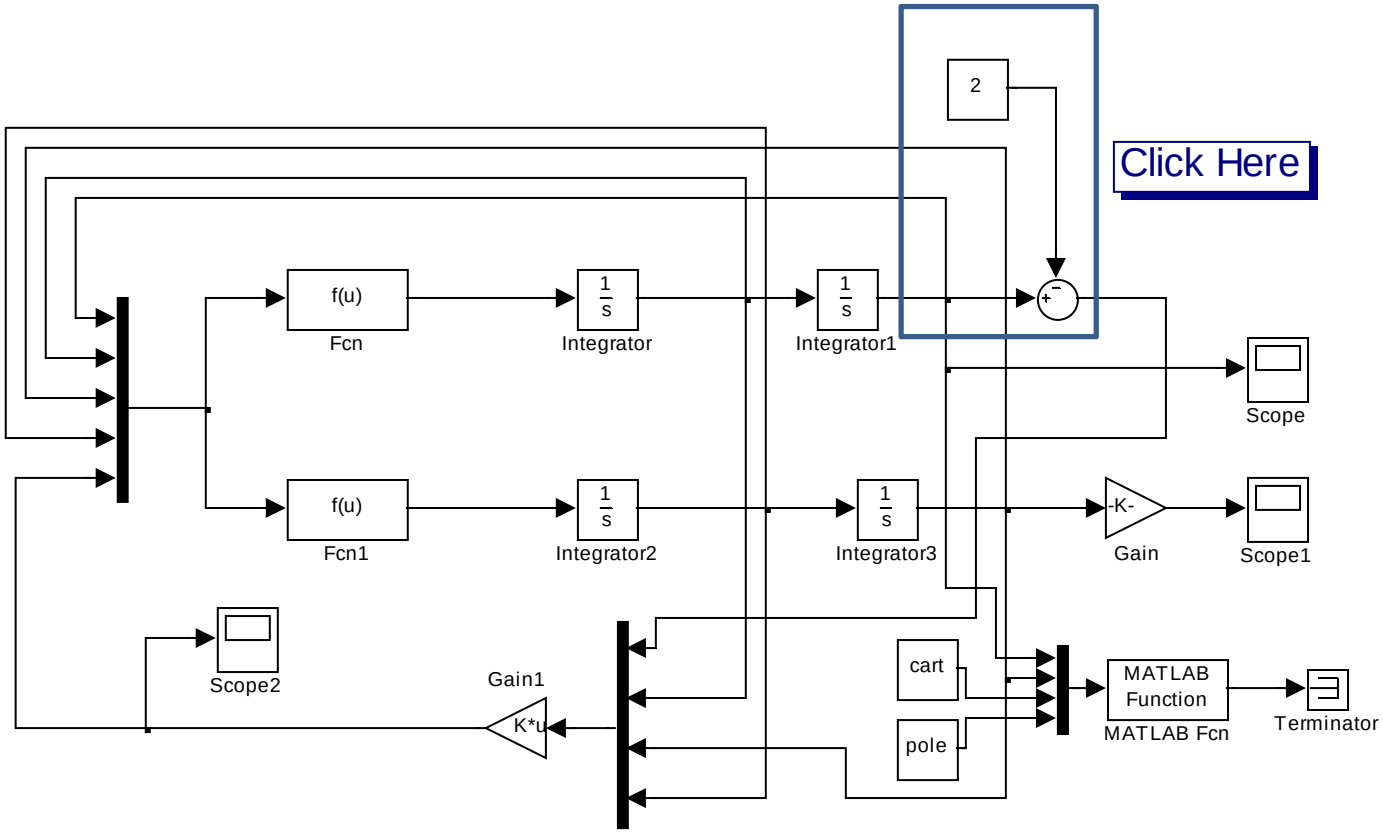


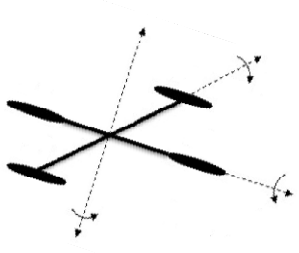

Function Block Parameters: Gain1
Gain
Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Main
Signal Attributes
Parameter Attributes
Gain:
-K
Multiplication:
Matrix(K*u)
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply

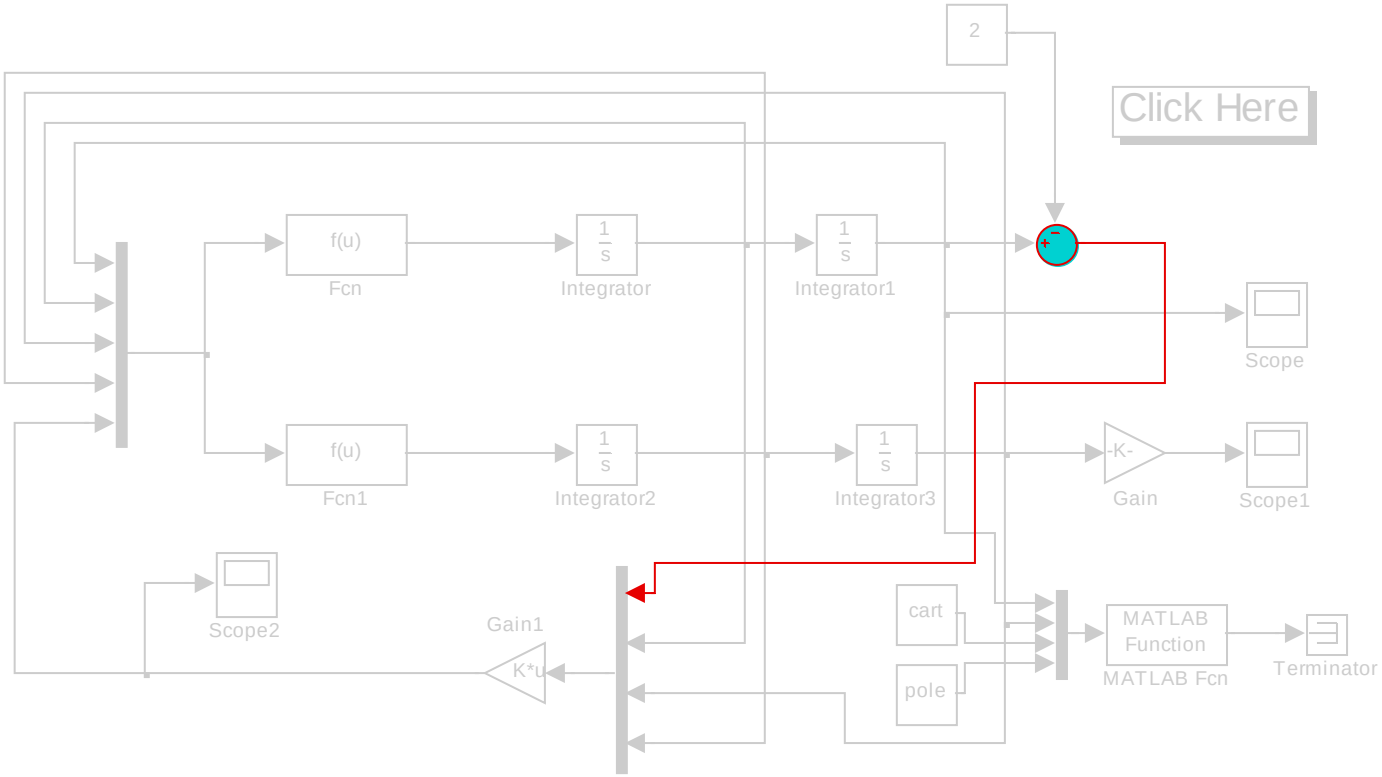


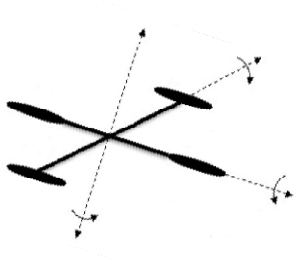

Scope
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
2
4
6
8
10
12
Time offset: 0
Scope1
10
5
0
-5
-10
0
2
4
6
8
10
12
Time offset: 0

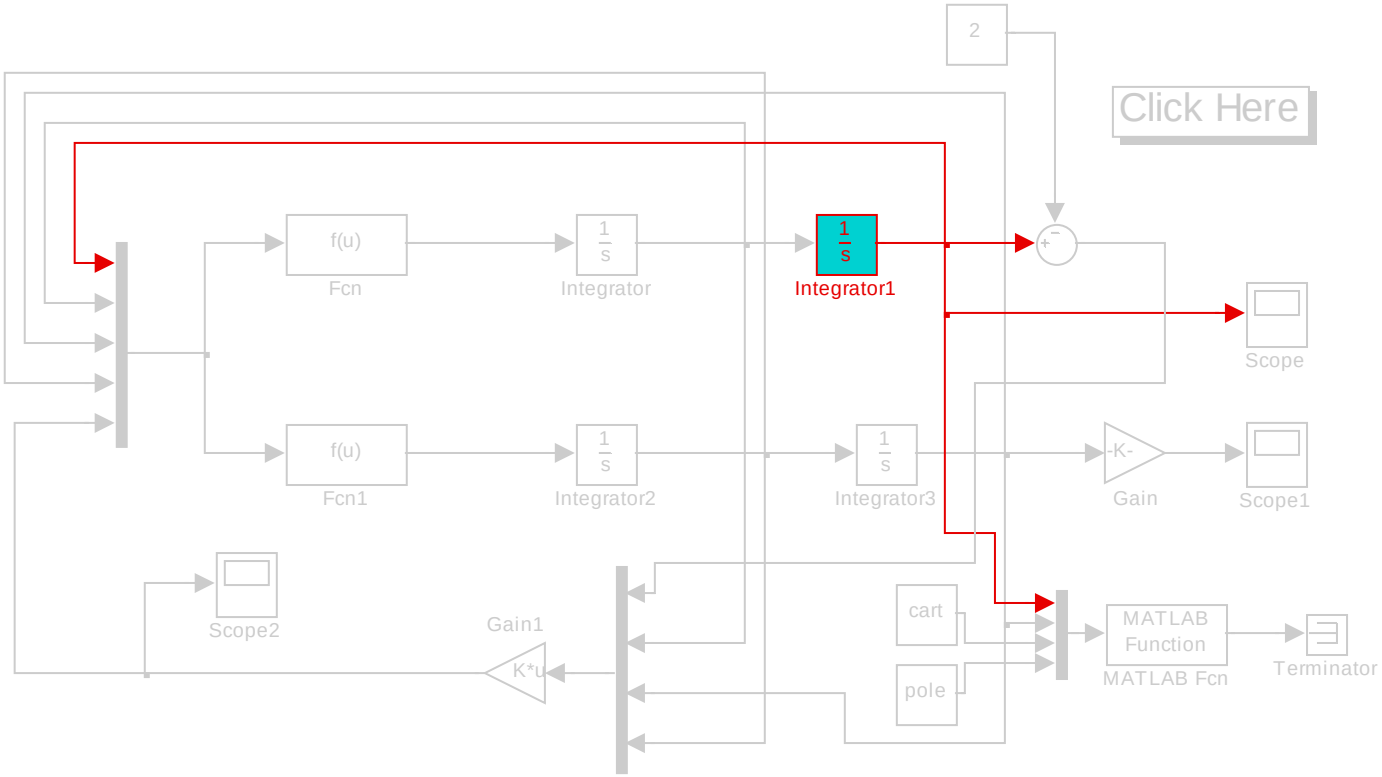


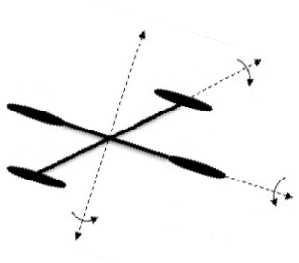

2
Click Here
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
Scope2
Gain1
K*u
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

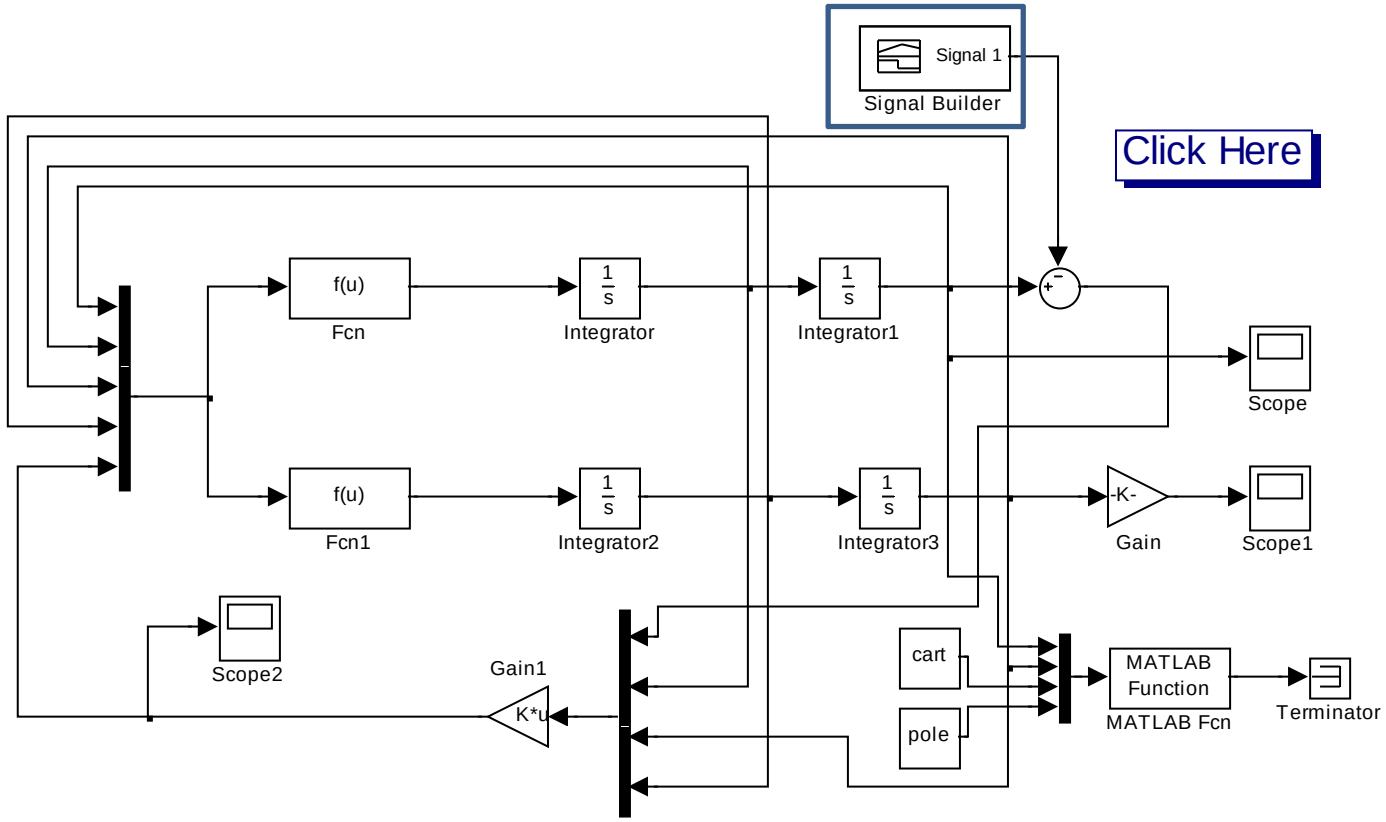


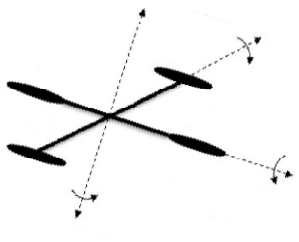

2
Click Here
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
Scope2
Gain1
K*u
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

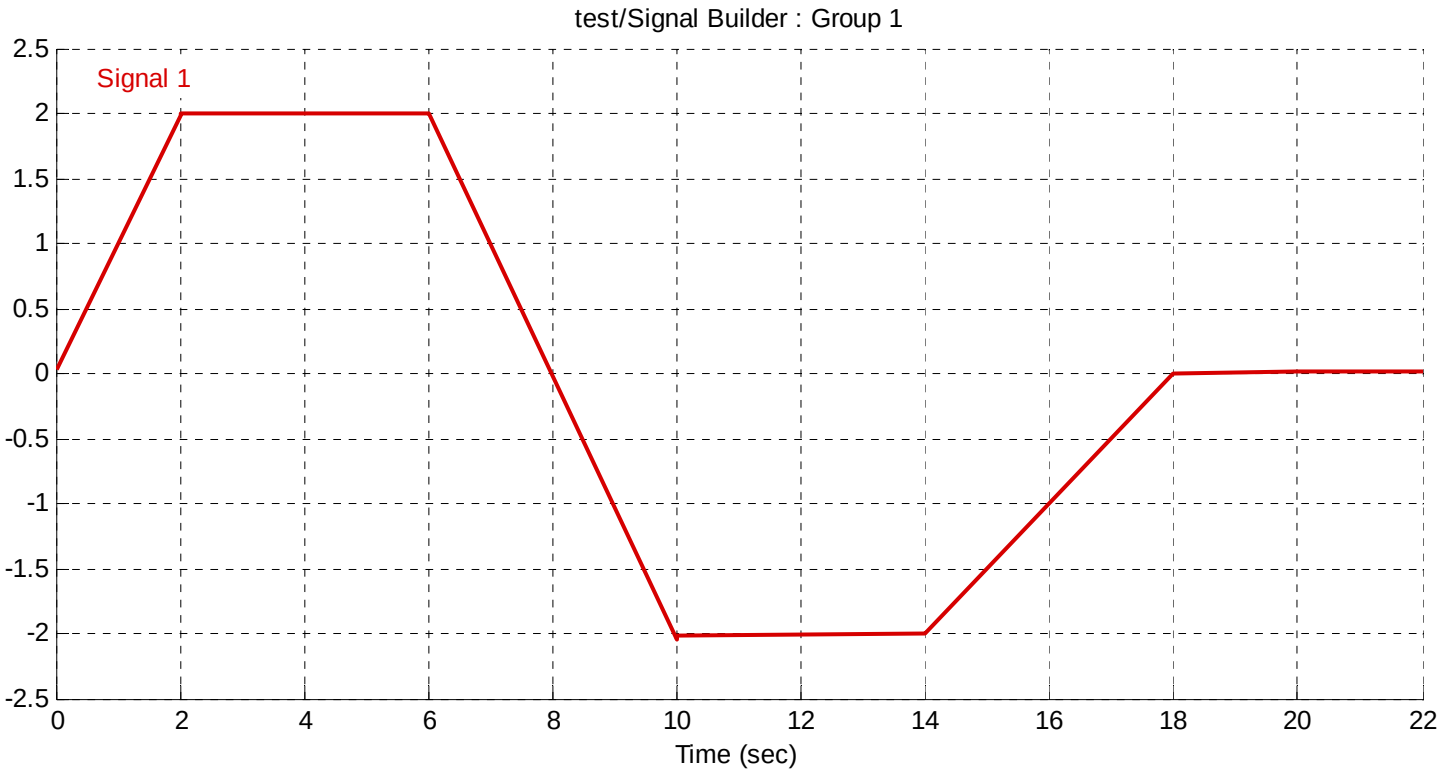


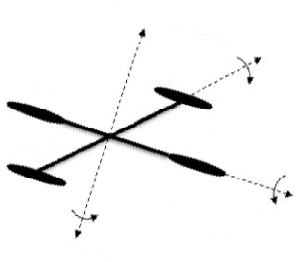

2
Click Here
f(u)
Fcn
1
s
Integrator
1
s
Integrator1
Scope
f(u)
Fcn1
1
s
Integrator2
1
s
Integrator3
-K-
Gain
Scope1
Scope2
Gain1
K*u
cart
pole
MATLAB
Function
MATLAB Fcn
Terminator

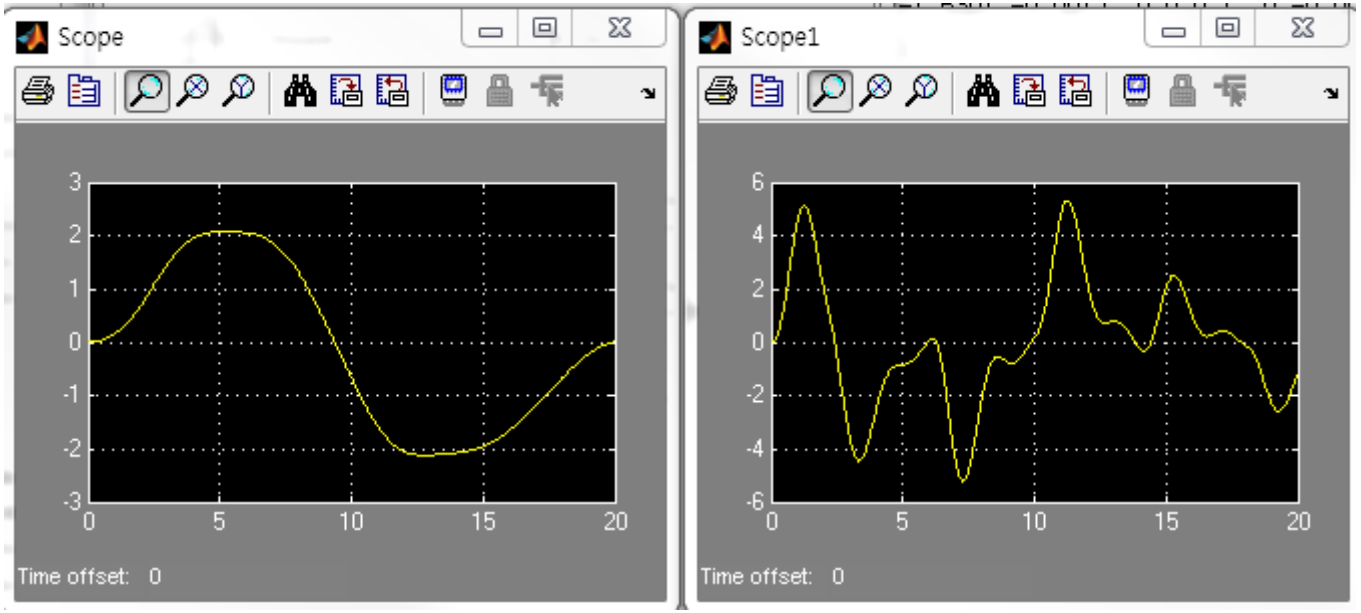



Signal 1
Signal Builder
Click Here
f(u)
Fcn
1/s
Integrator
1/s
Integrator1
Scope
f(u)
Fcn1
1/s
Integrator2
1/s
Integrator3
-K-
Gain
Scope1
Scope2
Gain1
K*u
cart
pole
MATLAB Function
MATLAB Fcn
Terminator




test/Signal Builder : Group 1
Signal 1
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
2
4
6
8
10
12
14
16
18
20
22
Time (sec)




Scope
3
2
1
0
-1
-2
-3
0
5
10
15
20
Time offset: 0
Scope1
6
4
2
0
-2
-4
-6
0
5
10
15
20
Time offset: 0