**Continious Assignment 2 Report**

Liu Xiangyu G1802061L

1. **Initialization**
   1. Calculate original RGA, RNGA, NI

Original RGA=

1.0986 -0.0876 -0.0110

0.0006 0.1045 0.8949

-0.0992 0.9831 0.1161

Original RNGA=

1.0339 -0.0100 -0.0240

0.0022 0.1035 0.8944

-0.0361 0.9065 0.1296

Original NI=

8.4716

* 1. Do pairing

According to original RGA, RNGA, NI, do pairing based on RNGA. Loop 1-1, 2-3 and 3-2 are chosen to be paird.

* 1. Calculate RGA RNGA NI RARTafter pairing

Paird RGA=

1.0986 -0.0876 -0.0110

-0.0992 0.9831 0.1161

0.0006 0.1045 0.8949

Paird RNGA=

1.0339 -0.0100 -0.0240

-0.0361 0.9065 0.1296

0.0022 0.1035 0.8944

Paird NI=

1.0199

RART =

0.9412 0.1139 2.1739

0.3640 0.9221 1.1163

3.5506 0.9905 0.9994

1. **BLT Decentralized Controller**
   1. Design Steps and Results
2. Find the ultimate gain Ku and ultimate frequency ωu for each diagonal using relay experiment method.

Ultimate gain Ku =

-63.0317

-1.0610

2.9610

Ultimate Frequency Wu =

2.2125

0.1559

1.8845

1. Calculate the expressions of Kci, tIi and tDi according to the formula.
2. Calculate the expressions of G(jw) and Lc(jw).
3. First Iteration Loop, to calculate the optimal F.

F =

1.8400

1. Second Iteration Loop, give the optimal F, to calculate the optimal Fd.

Fd =

16.3900

1. Export parameters to Simulink blocks.

kpBlt =

-15.5711 -0.2621 0.7315

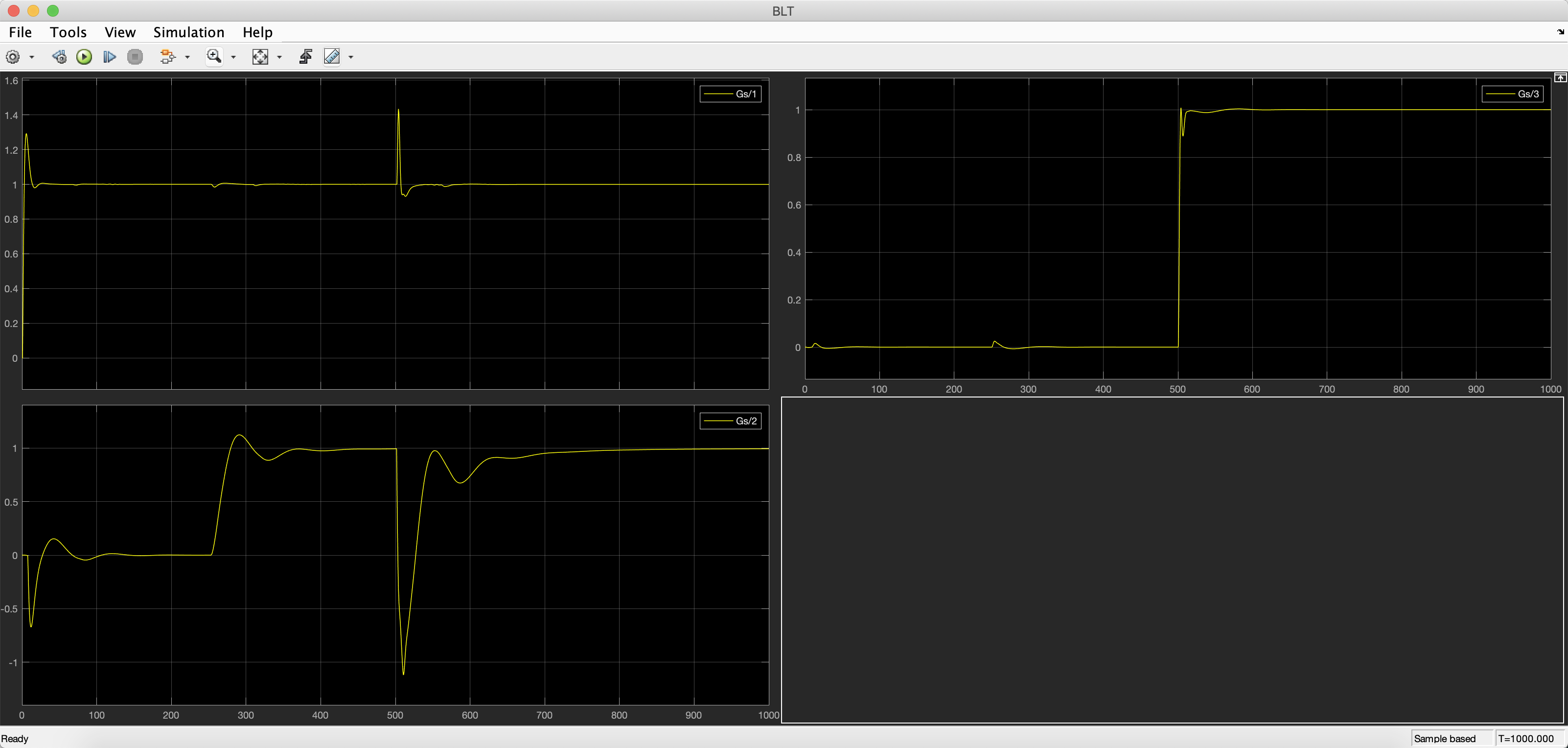
kiBlt =

-3.5760 -0.0042 0.1431

kdBlt =

-0.2764 -0.0660 0.0152

1. Execute simulation.
   1. Figures



1. **ETF Decentralized Controller**
   1. Design Steps and Results
      1. Find ETFs for diagonal loops according to Decentralized ETF rules.
      2. Specify an Am and Calculate PID parameters by GPM Method.

kpDec =

-18.5675 -0.3919 1.0055

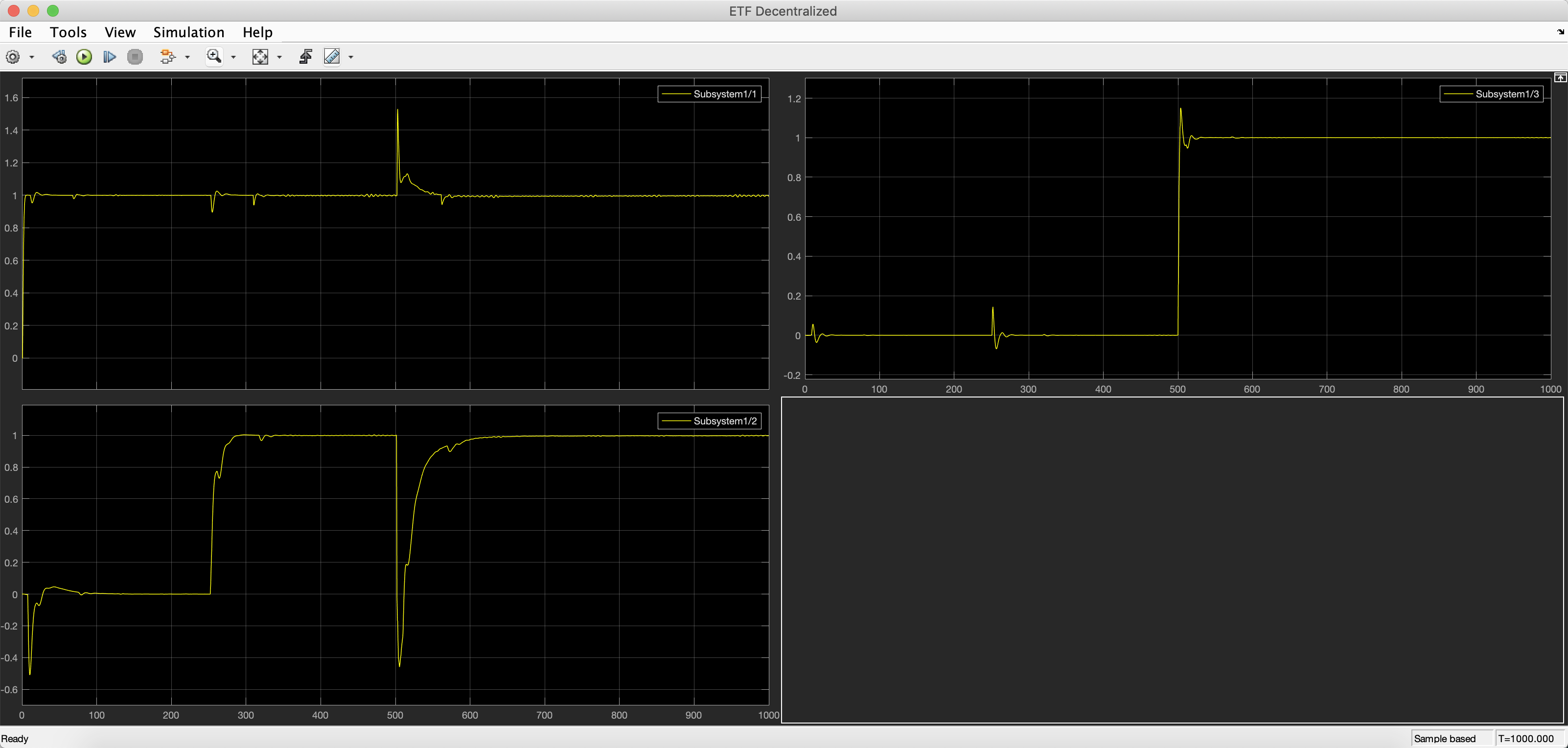
kiDec =

-0.2785 -0.0090 0.3516

kdDec =

0 -4.2603 0.7189

* + 1. Export parameters to Simulink blocks.
  1. Figures



1. **ETF Sparse Controller**
   1. Design Steps and Results
2. Calculate the index matrix

X =

1.0000 0.0096 0.0232

0.0398 1.0000 0.1430

0.0024 0.1157 1.0000

1. Specify a value of epsilon and threshold the index matrix to find which loops to control. Since all off-diagonal index are less than the empirical value 0.15, 0.10 is specified.

G0 =

1.0000 0 0

0 1.0000 0.1430

0 0.1157 1.0000

1. Find ETFs for all loops whose G0 value is non-zero.
2. Calculate PID parameters of ETF.

kpSprs =

-18.5675 0 0

0 -0.3919 -0.8703

0 -0.0332 1.0055

kiSprs =

-0.2785 0 0

0 -0.0090 -0.1828

0 -0.0026 0.3516

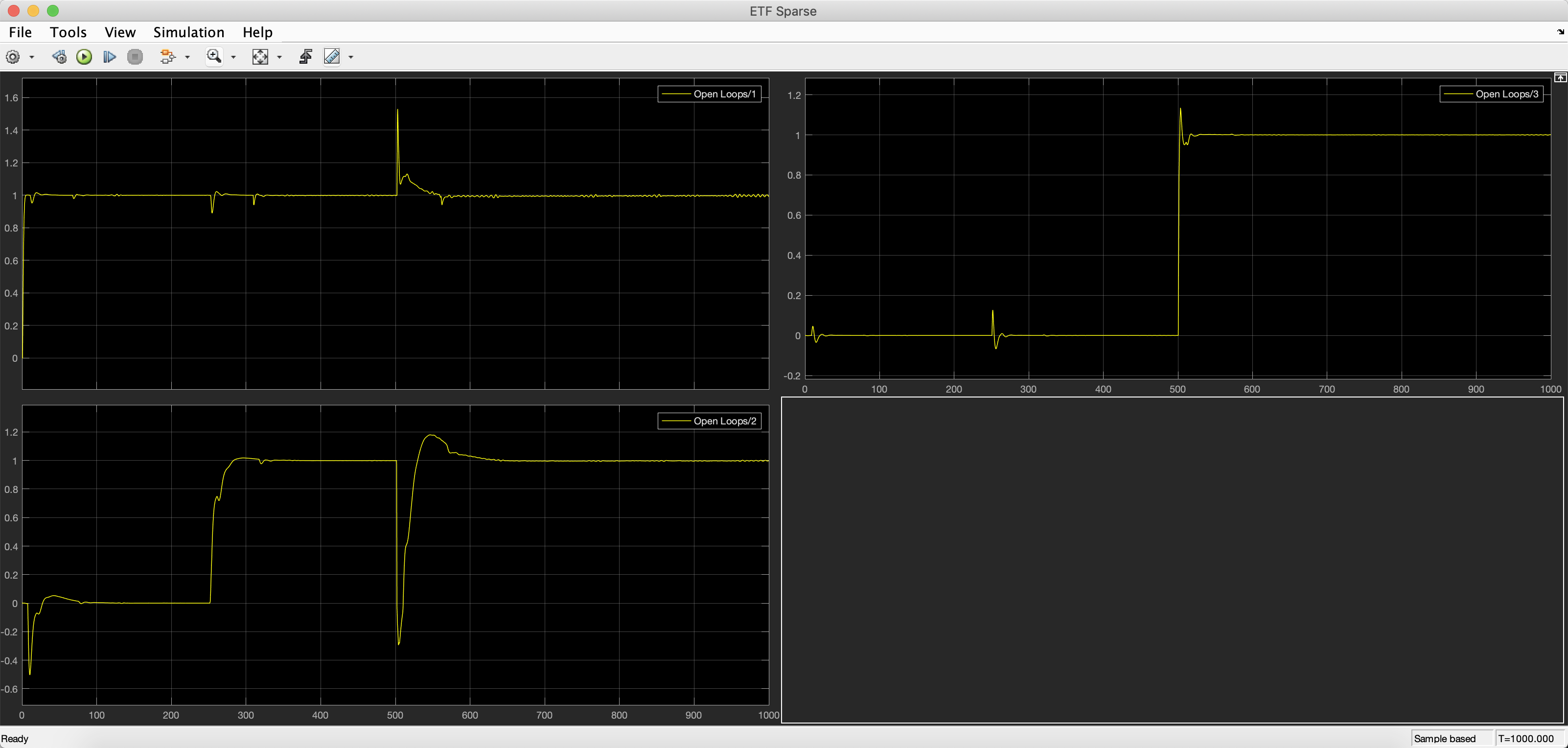
kdSprs =

0 0 0

0 -4.2603 -1.0357

0 0 0.7189

* 1. Figures



1. **ETF Decoupling Controller**
   1. Design Steps and Results
      1. Using Normalized Decoupling Method, find GI = inv(Gs)\*GR, diagonal elements of Gs chosen as GR.
      2. Specify time delay for gRs and gIs to maintain causality.

Calculate PID parameters of ETFs of GR with specified time delay.

kpDecp =

-2.5712

-0.5765

0.3004

kiDecp =

-0.0386

-0.0133

0.1050

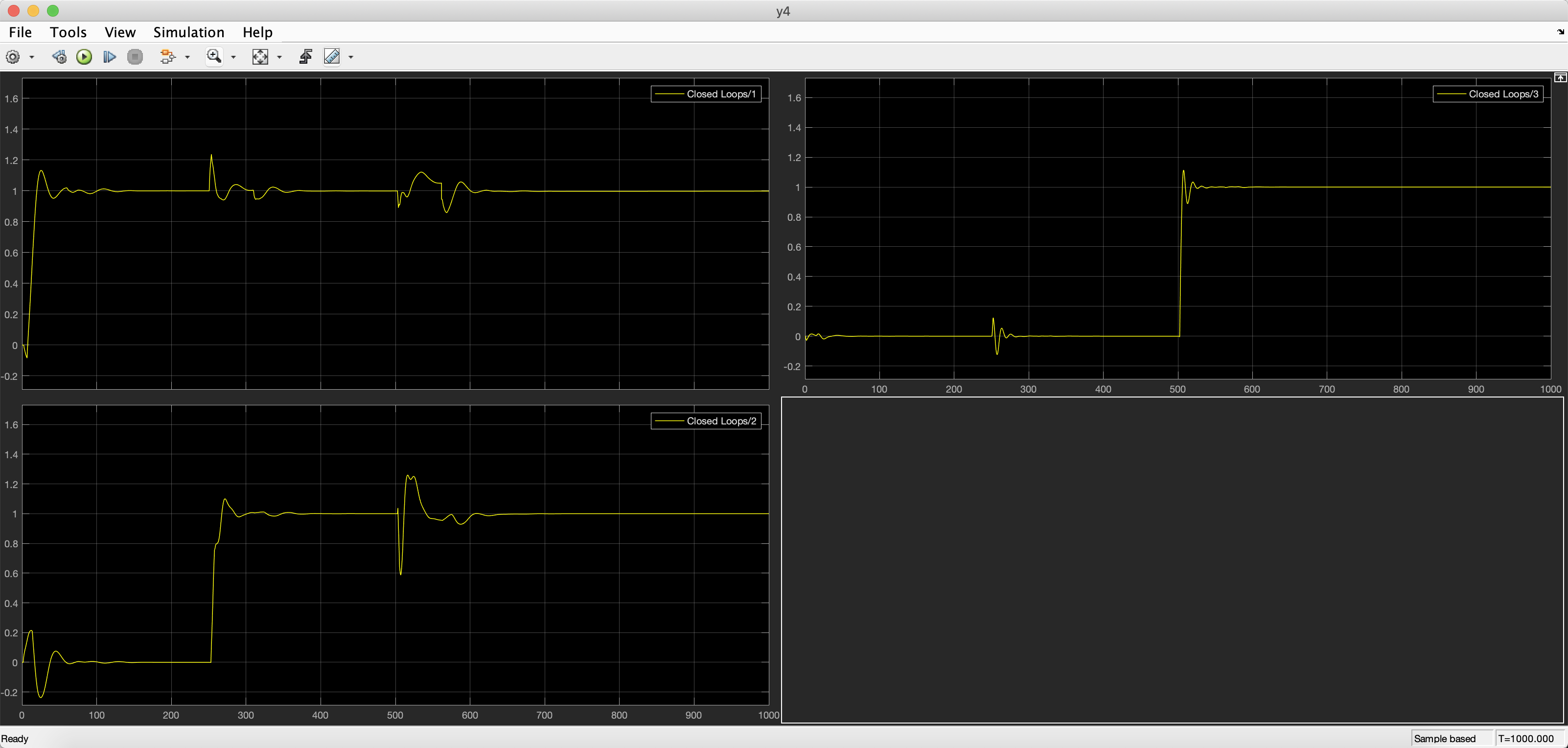
kdDecp =

0

-6.2665

0.2148

* + 1. Export PID parameters and GI transfor functions to Simulink blocks.
  1. Figures



1. **Appendix I: Matlab Scripts**
   1. **BLT.m**

%relay experiment results

h=1;

a(1)=-0.0202;

a(3)=0.43;

a(2)=-1.2;

Pu(1)=18.2552-15.4154;

Pu(3)=16.6099-13.2757;

Pu(2)=101.5974-61.2889;

Wu=zeros(3,1);

Ku=zeros(3,1);

for i=1:3

Wu(i)=2\*pi/Pu(i);

Ku(i)=4\*h/(pi\*a(i));

end

syms s;

Gjw=zeros(3,3);

Kzn=zeros(3,1);

tzn=zeros(3,1);

tdzn=zeros(3,1);

Kc=zeros(3,1);

ti=zeros(3,1);

td=zeros(3,1);

F=0;

Fd=0;

%calculate optimal F

BLT\_1

%calculate optimal Fd

BLT\_2

%get the pid parameters

for i=1:3

kpBlt(i)=Kc(i);

kiBlt(i)=Kc(i)/ti(i);

kdBlt(i)=Kc(i)\*td(i);

end

* 1. **BLT\_1.m**

for F=1:0.01:15

%calculate Kc ti

for i=1:3

Kzn(i)=Ku(i)/2.2;

tzn(i)=2\*pi/(1.2\*Wu(i));

Kc(i)=Kzn(i)/F;

ti(i)=F\*tzn(i);

end

%calculate Gjw Gcjw and Lc

for k=1:100

w=pi/100\*k;

s=w\*j;

for m=1:3

for n=1:3

Gjw(m,n)=K(m,n)/(T(m,n)\*s+1)^O(m,n)\*exp(-L(m,n)\*s);

end

end

for i=1:3

Gcjw(i,i)=Kc(i)\*(1+1/(ti(i)\*j\*w));

end

Wjw=-1+det(I+Gjw\*Gcjw);

Lc\_=20\*log10(abs(Wjw/(1+Wjw)));

if Lc\_<=6

Lc(k)=20\*log10(abs(Wjw/(1+Wjw)));

end

end

Lcm=[Lcm; F max(Lc)];

end

%find the optimal F

[v id]=max(Lcm(:,2));

F=Lcm(id,1)

* 1. **BLT\_2.m**

for Fd=1:0.01:20

for i=1:3

Kzn(i)=Ku(i)/2.2;

tzn(i)=2\*pi/(1.2\*Wu(i));

tdzn(i)=0.125\*Pu(i);

Kc(i)=Kzn(i)/F;

ti(i)=F\*tzn(i);

td(i)=tdzn(i)/Fd;

end

for k=1:100

w=pi/100\*k;

s=w\*j;

for m=1:3

for n=1:3

Gjw(m,n)=K(m,n)/(T(m,n)\*s+1)^O(m,n)\*exp(-L(m,n)\*s);

end

end

for i=1:3

Gcjw(i,i)=Kc(i)\*(1+1/(ti(i)\*j\*w)+td(i)\*j\*w);

end

Wjw=-1+det(I+Gjw\*Gcjw);

Lc\_=20\*log10(abs(Wjw/(1+Wjw)));

if Lc\_<=6

Lc(k)=20\*log10(abs(Wjw/(1+Wjw)));

end

end

Lcm=[Lcm; Fd max(Lc)];

end

[v id]=max(Lcm(:,2));

Fd=Lcm(id,1)

* 1. **ETFDecentralized.m**

AmDec=ones(3,1)\*4;

for i=1:3

for j=1:3

if abs(RGA(i,j))<1

kij=K(i,j)/RGA(i,j);

else

kij=K(i,j);

end

if RART(i,j)>1

Zeta(i,j)=Zeta(i,j)/RART(i,j);

Tij=T(i,j)\*RART(i,j);

Lij=L(i,j)\*RART(i,j);

else

Tij=T(i,j);

Lij=L(i,j);

end

if i==j

if O(i,j)==1

kpDec(i)=pi\*Tij/(2\*AmDec(i)\*Lij\*kij);

kiDec(i)=pi/(2\*AmDec(i)\*Lij\*kij);

kdDec(i)=0;

elseif O(i,j)==2

if kij==K(i,j)

kij=kij\*sign(RGA(i,j));

end

kpDec(i)=pi\*Zeta(i,j)/(AmDec(i)\*wn(i,j)\*Lij\*kij);

kiDec(i)=pi/(2\*AmDec(i)\*Lij\*kij);

kdDec(i)=pi/(2\*AmDec(i)\*wn(i,j)^2\*Lij\*kij);

end

end

end

end

* 1. **ETFDecoupling.m**

AmDecp=[3,3,3];

clear s;

s=tf('s');

phim=pi/4;

%get Gs and GR

Gs\_=s;

GR=[s 0 0;0 s 0;0 0 s];

for i=1:3

for j=1:3

Gs\_(i,j)=K(i,j)/(T(i,j)\*s+1)^O(i,j);

end

GR(i,i)=K(i,i)/(T(i,i)\*s+1)^O(i,i);

end

%calculate GI

GI=inv(Gs\_)\*GR;

%specify L

L\_=L.\*RART;

for i=1:3

LR(i,i)=max(L\_(i,:));

end

for i=1:3

for j=1:3

GI(i,j)=GI(i,j)\*exp(-(LR(i,i)-L\_(i,j))\*s);

end

GR(i,i)=GR(i,i)\*exp(-LR(i,i)\*s);

end

%calculate PID parameters

kpDecp(1)=pi\*T(1,1)/(2\*AmDecp(1)\*LR(1,1)\*K(1,1));

kiDecp(1)=pi/(2\*AmDecp(1)\*LR(1,1)\*K(1,1));

kdDecp(1)=0;

for i=2:3

kpDecp(i)=T(i,i)\*pi/(AmDecp(i)\*LR(i,i)\*K(i,i));

kiDecp(i)=pi/(2\*AmDecp(i)\*LR(i,i)\*K(i,i));

kdDecp(i)=pi\*T(i,i)^2/(2\*AmDecp(i)\*LR(i,i)\*K(i,i));

end

* 1. **ETFSparse.m**

%caculate the index matrix

for i=1:3

for j=1:3

X(i,j)=abs(RNGA(i,j)/RNGA(i,i));

if X(i,j)>1

X(i,j)=1/X(i,j);

end

end

end

eps=0.1;

for i=1:3

for j=1:3

if X(i,j)<eps

G0(i,j)=0;

else

G0(i,j)=X(i,j);

end

end

end

AmSprs=ones(3)\*4;

kpSprs=zeros(3,3);

kiSprs=zeros(3,3);

kdSprs=zeros(3,3);

%calculate PID parameters for each ETFs

for i=1:3

for j=1:3

if abs(RGA(i,j))<1

kij=K(i,j)/RGA(i,j);

else

kij=K(i,j);

end

if RART(i,j)>1

Zeta(i,j)=Zeta(i,j)/RART(i,j);

Tij=T(i,j)\*RART(i,j);

Lij=L(i,j)\*RART(i,j);

else

Tij=T(i,j);

Lij=L(i,j);

end

if G0(i,j)~=0

if O(i,j)==1

kpSprs(j,i)=pi\*Tij/(2\*AmSprs(i,j)\*Lij\*kij);

kiSprs(j,i)=pi/(2\*AmSprs(i,j)\*Lij\*kij);

kdSprs(j,i)=0;

elseif O(i,j)==2

if kij==K(i,j)

kij=kij\*sign(RGA(i,j));

end

kpSprs(j,i)=pi\*Zeta(i,j)/(AmSprs(i,j)\*wn(i,j)\*Lij\*kij);

kiSprs(j,i)=pi/(2\*AmSprs(i,j)\*Lij\*kij);

kdSprs(j,i)=pi/(2\*AmSprs(i,j)\*wn(i,j)^2\*Lij\*kij);

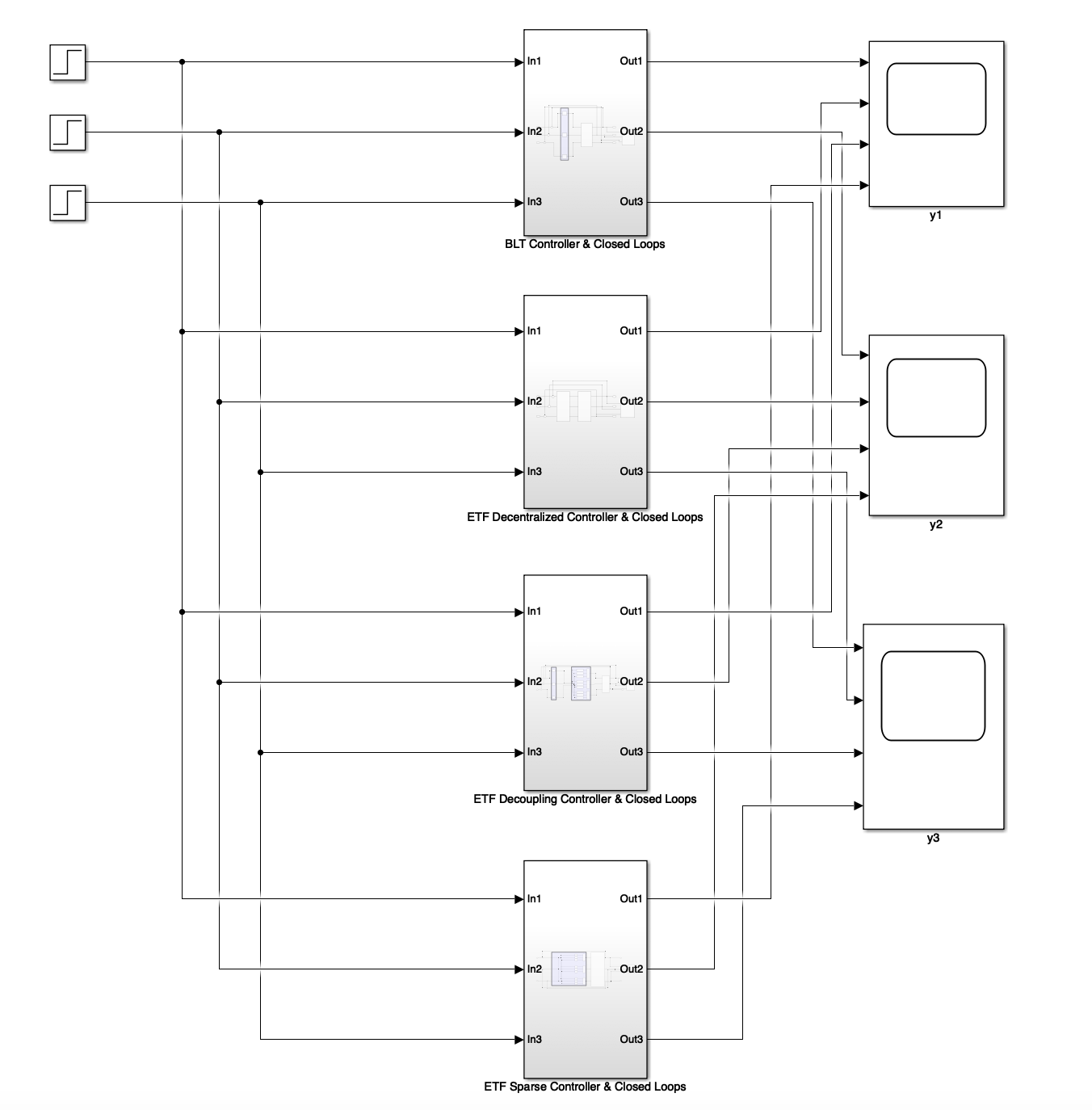
end

end

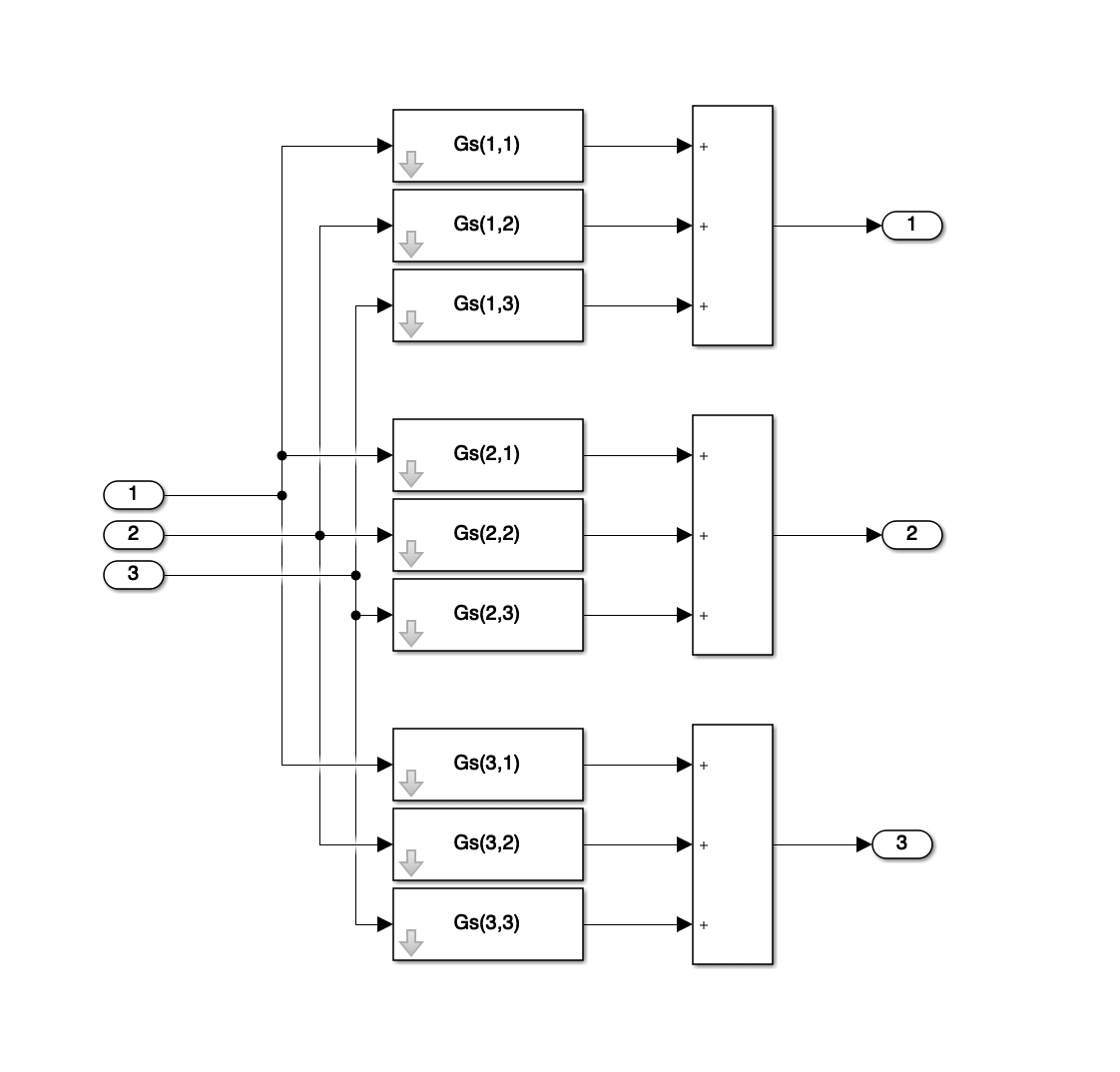
end

end

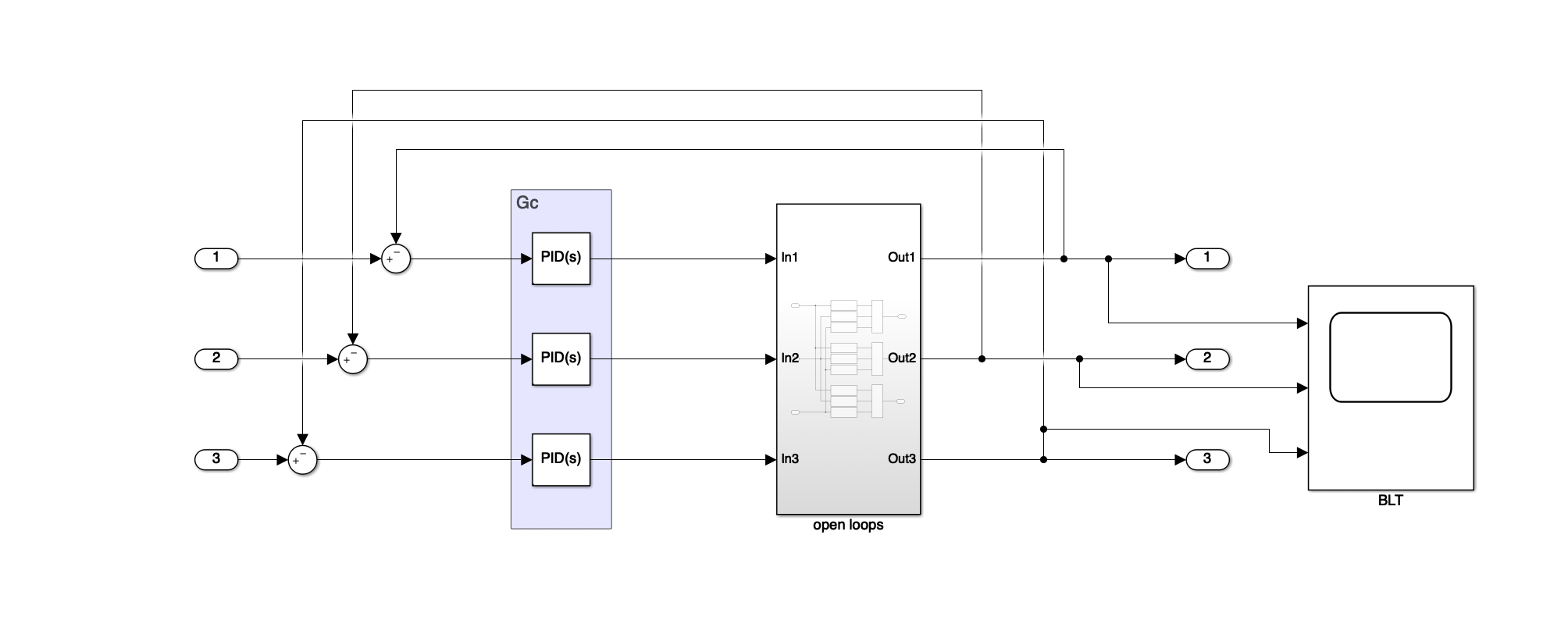
1. **Appendix II: Simulink Models**
   1. **Overall Structure**

****

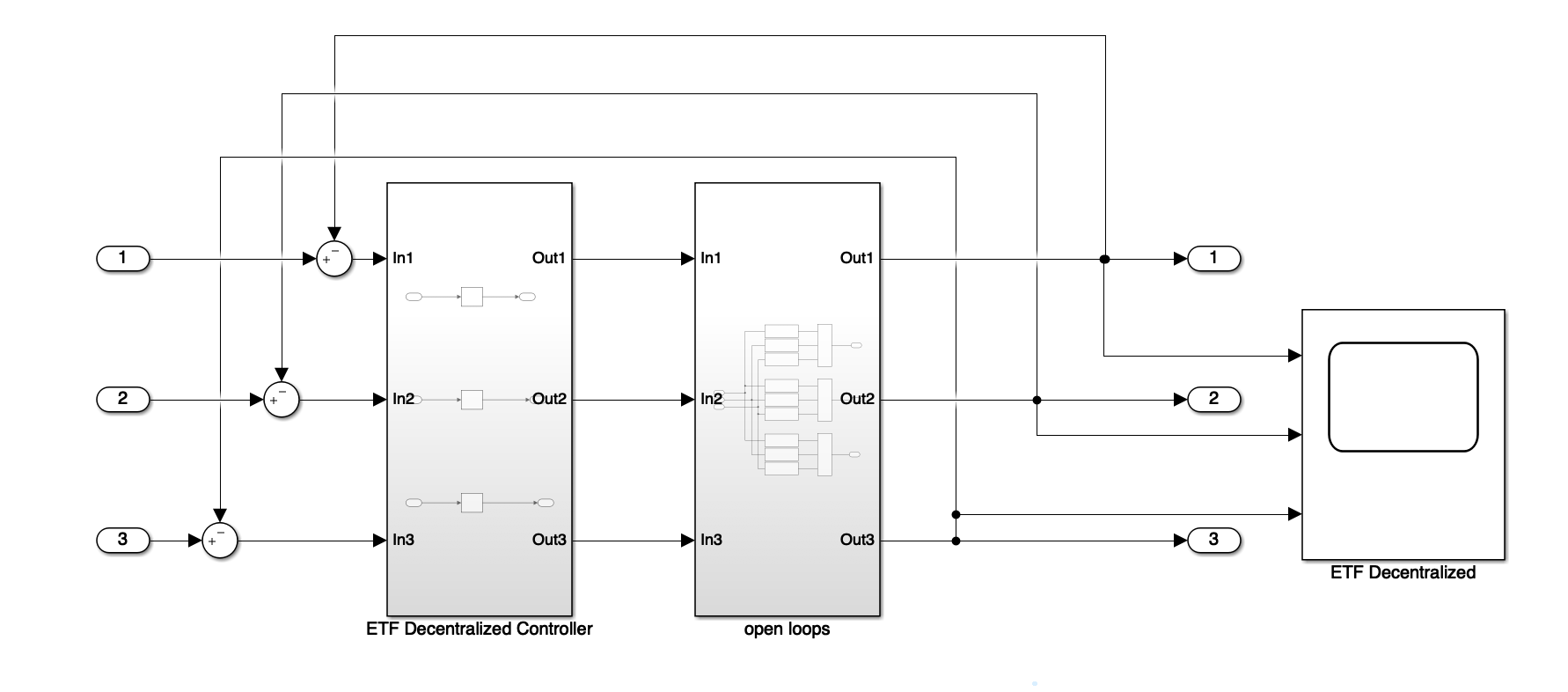
* 1. **Open loops**

****

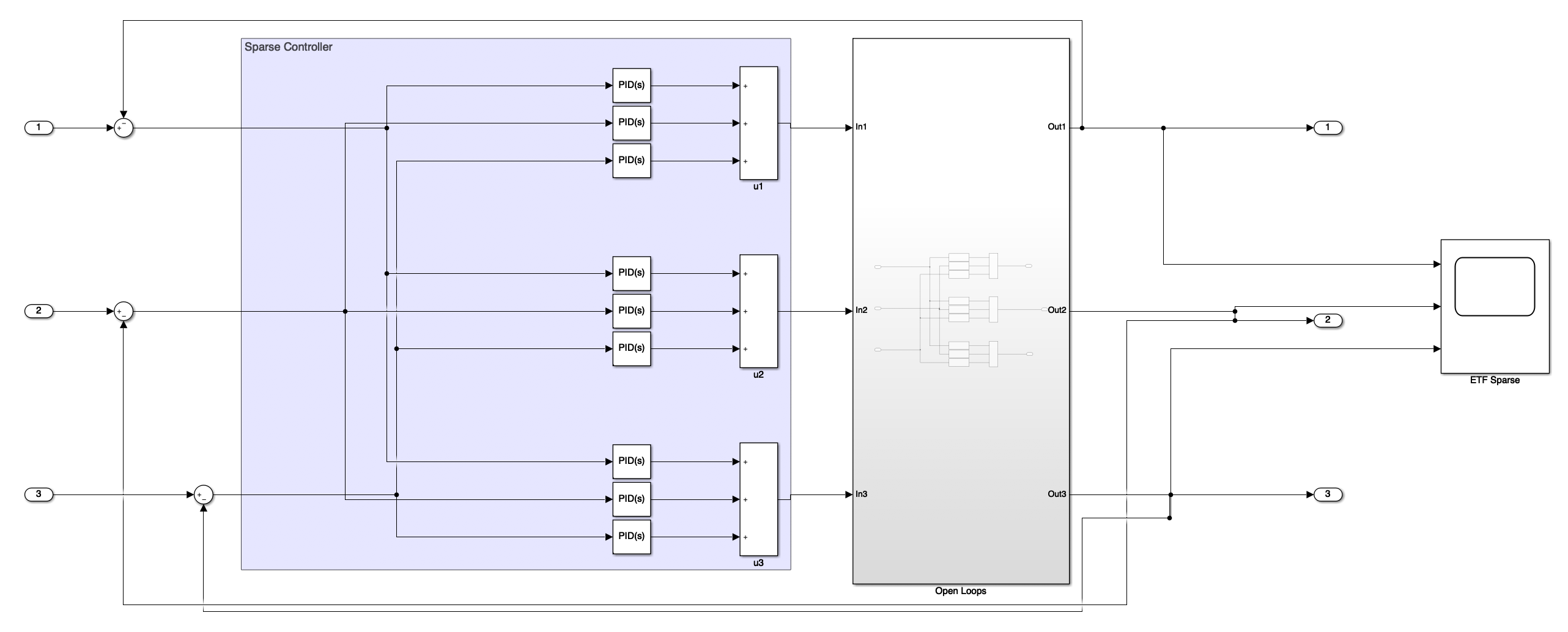
* 1. **BLT**

****

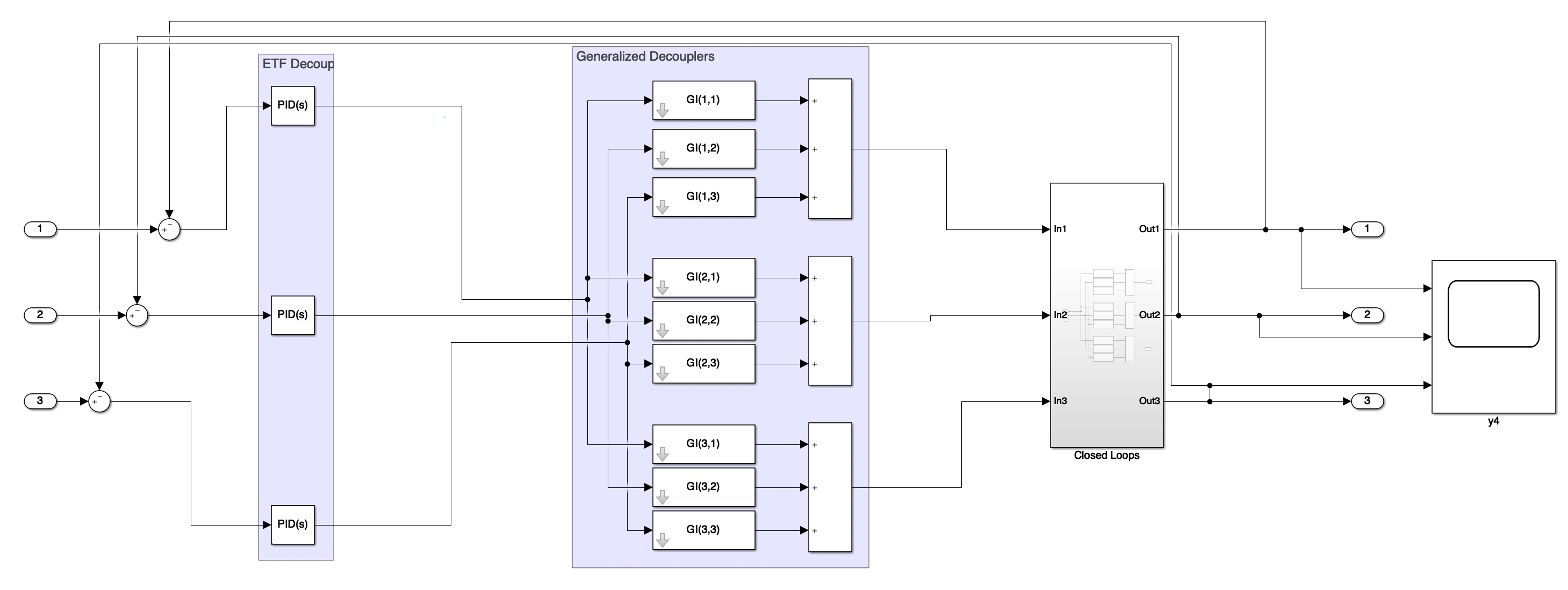
* 1. **ETF Decentralized Controller**

****

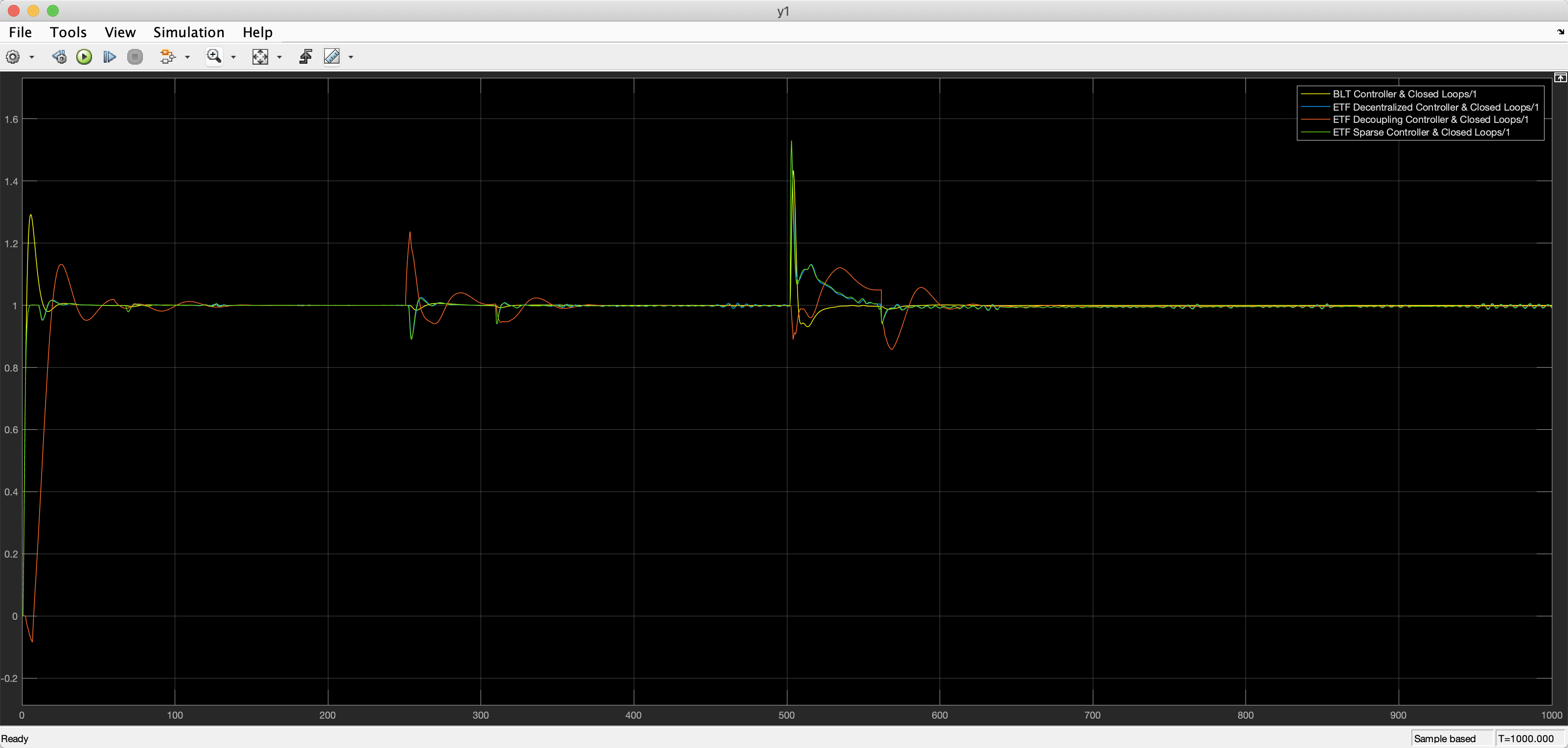
* 1. **ETF Sparse Controller**

****

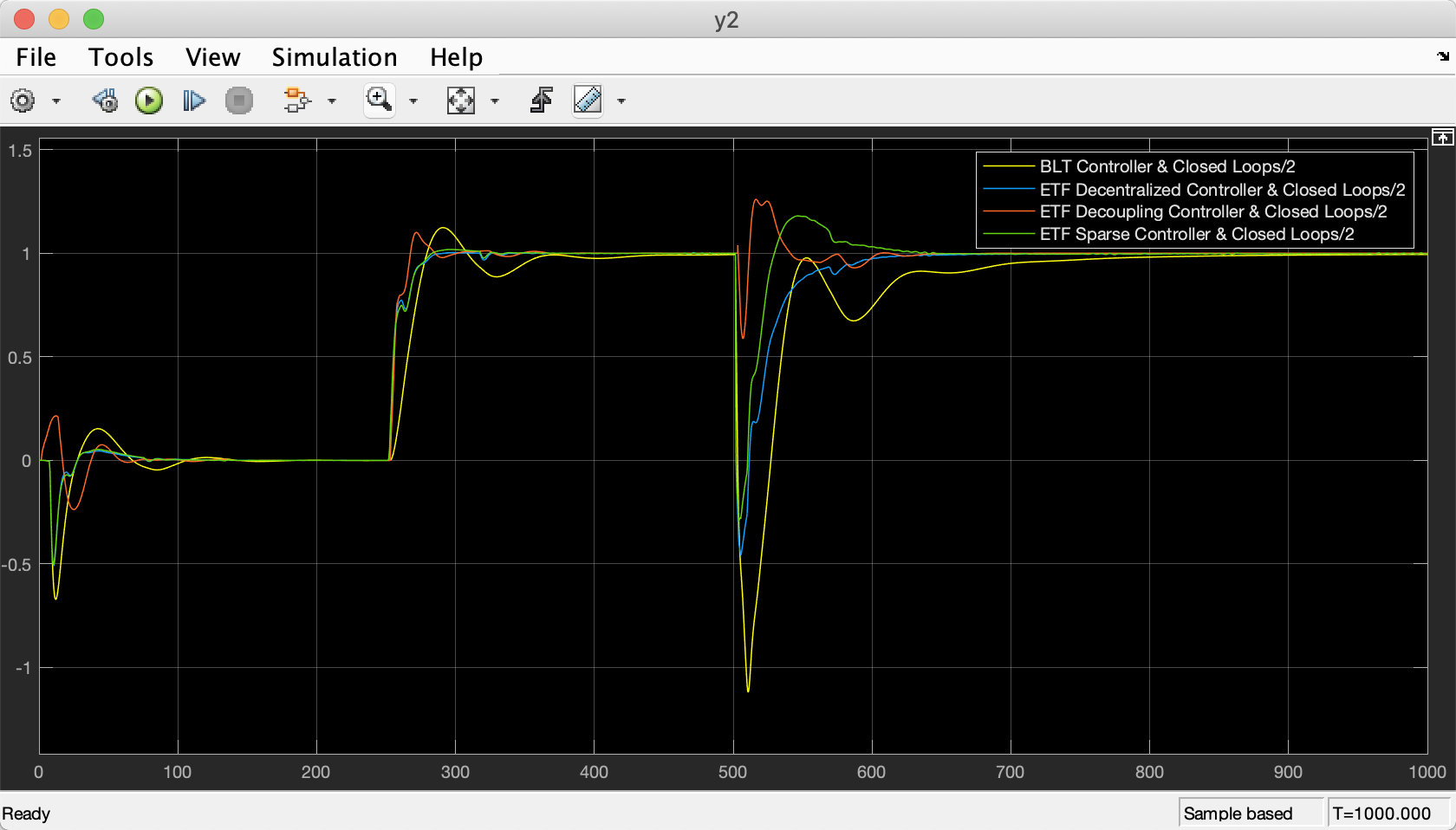
* 1. **ETF Decoupling Controller**

****

1. **Appendix III: Comparing Figures**
   1. **Y1**

****

* 1. **Y2**

****

* 1. **Y3**

****