Optimization-Based Controller Design

Unconstrained optimization with Matlab

- Find a vector x to minimize the objective function $F(x)$

$$\min_{x} F(x)$$

where $x = [x_1, x_2, \cdots, x_n]^T$

Newton's method:

Local linear model of $F(x)$ about x_k:

$$M(x) = F(x_k) + \nabla F(x_k)(x - x_k)$$

Solving for $M(x_{k+1}) = 0$

$$\Rightarrow \quad x_{k+1} = x_k - \nabla F(x_k)^{-1} F(x_k)$$

Matlab function: fminsearch(),

Syntax: [x,f_{opt},key,c]=fminsearch(Fun,x_0,OPT)

Fun is a Matlab function, an inline function, or an anonymous function to optimize

Variable x_0 is the starting point of the optimization solution

OPT contains other control options for the optimization process

Example: Find the minimum of the function: $z = f(x,y) = (x^2 - 2x)e^{-x^2-y^2-xy}$ for the unknown variables x, y.

Let $\boldsymbol{x} = (x_1, x_2)^T = (x, y)^T \;\; \Rightarrow f(\boldsymbol{x}) = ({x_1}^2 - 2x_1)e^{-{x_1}^2-{x_2}^2-x_1x_2}$

Matlab code:

```
F=@(x) [(x(1)^2-2*x(1))*exp(-x(1)^2-x(2)^2-x(1)*x(2))];
x0=[0;0]; x=fminsearch(F,x0)
```

$\Rightarrow$ Optimal solution: $\boldsymbol{x} = (0.6110, -0.3055)^T$

Optimization-Based Controller Design

Constrained optimization with Matlab

- Find a vector x to minimize the objective function $F(x)$ while satisfying all constraints

$$\min_{x \ni \begin{cases} Ax \le B \\ A_{eq} = B_{eq} \\ x_m \le x \le x_M \\ C(x) \le 0 \\ C_{eq}(x) = 0 \end{cases}} F(x)$$

where $x = [x_1, x_2, \cdots, x_n]^T$

Constraints:

Linear inequality constraint: $Ax \le B$

Linear equality constraint: $A_{eq} = B_{eq}$

Upper and lower bounds of optimization: $x_m \le x \le x_M$

Nonlinear inequality and equality constraints: $C(x) \le 0$ and $C_{eq}(x) = 0$

Matlab function: fmincon(),

Syntax: $[x, f_{opt}, key, c] = \text{fmincon}(Fun, x_0, A, B, A_{eq}, B_{eq}, x_m, x_M, CFun, OPT)$

Fun is a Matlab function, an inline function, or an anonymous function to optimize

Variable x_0 is the starting point of the optimization solution

The nonlinear constraints can be described by the Matlab function CFun

Example: Solve the nonlinear programming problem:

$$\min_{x \ni \begin{cases} 8x_1 + 14x_2 + 7x_3 - 56 = 0 \\ x_1, x_2, x_3 \ge 0 \\ {x_1}^2 + {x_2}^2 + {x_3}^2 - 25 = 0 \end{cases}} 1000 - {x_1}^2 - 2{x_2}^2 - {x_3}^2 - x_1 x_2 - x_1 x_3$$

Matlab code:

```
F=@(x) 1000-x(1)^2-2*x(2)^2-x(3)^2-x(1)*x(2)-x(1)*x(3);
x0=[1;1;1]; xm=[0;0;0]; xM=[]; A=[]; B=[]; Aeq=[8,14,7]; Beq=56;
[x,Fopt,c,d]=fmincon(F,x0,A,B,Aeq,Beq,xm,xM,'opt_con')
```

Matlab function for nonlinear constraints:

```
function [c,ceq]=opt_con(x)
c=[];   ceq=x(1)^2+x(2)^2+x(3)^2-25;
```

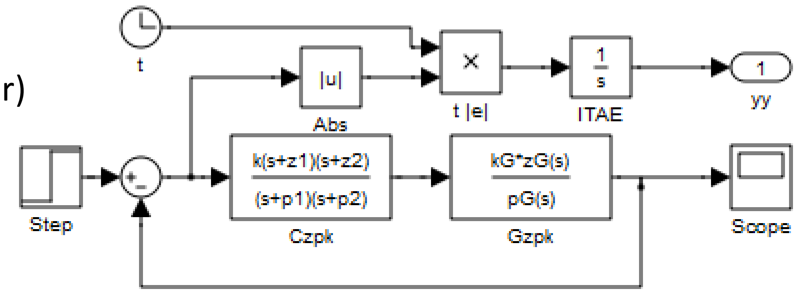
$\Rightarrow$ Optimal solution: $\boldsymbol{x} = (3.5121, 0.2170, 3.5522)^T$, $F_{opt} = 961.7152$

Optimization-Based Controller Design

Example: For system $G(s) = \dfrac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$ design a lead-lag control to minimize the ITAE criterion

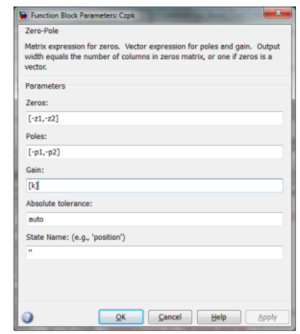
(ITAE = integral of time multiplied by absolute-value of error)

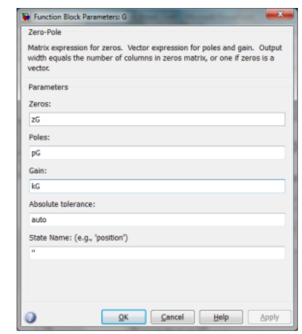
Matlab functions: fminsearch(), assignin(),



Matlab code:

```
clear all, clc,
G=zpk([-1,-0.5],[0,-0.1,-2,-10,-20],4);
[zG,pG,kG]=zpkdata(G,'v');
x0=20*ones(5,1);
[xopt,fopt,flag,iter]=fminsearch('leadlag_opt',x0),
z1=xopt(1), z2=xopt(2), p1=xopt(3), p2=xopt(4), k=xopt(5),
```





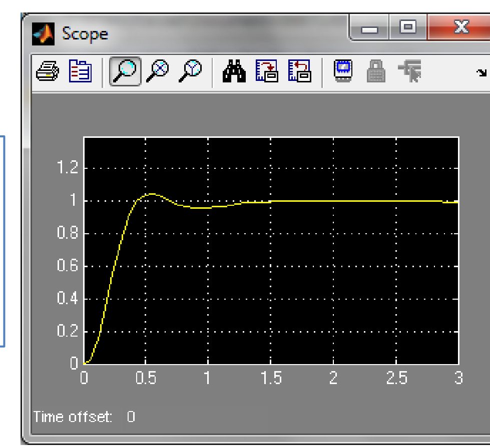
Matlab function for ITAE-based lead-lag:

```
function y=leadlag_opt(x)
assignin('base','z1',x(1));
assignin('base','z2',x(2));
assignin('base','p1',x(3));
assignin('base','p2',x(4));
assignin('base','k',x(5));
[t,xx,yy]=sim('leadlag_sim.mdl',3);
y=yy(end);
```

Optimal lead-lag compensator:

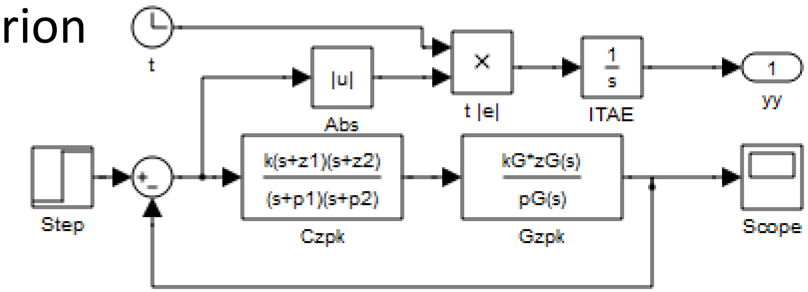
$$G_c(s) = 329.7553\,\frac{(s+6.5815)(s-0.3580)}{(s+7.7848)(s-0.2790)}$$

Unstable controller! (Not good!)



Optimization-Based Controller Design

Example: For system $G(s) = \frac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$ design a lead-lag control with LHP poles/zeros to minimize the ITAE criterion



Matlab code:

```
clear all, clc,
G=zpk([-1,-0.5],[0,-0.1,-2,-10,-20],4);
[zG,pG,kG]=zpkdata(G,'v');
x0=20*ones(5,1);
[xopt,fopt,flag,iter]=fmincon ('leadlag_opt_c',x0,[],[],[],[],zeros(5,1)),
z1=xopt(1), z2=xopt(2), p1=xopt(3), p2=xopt(4), k=xopt(5),
```

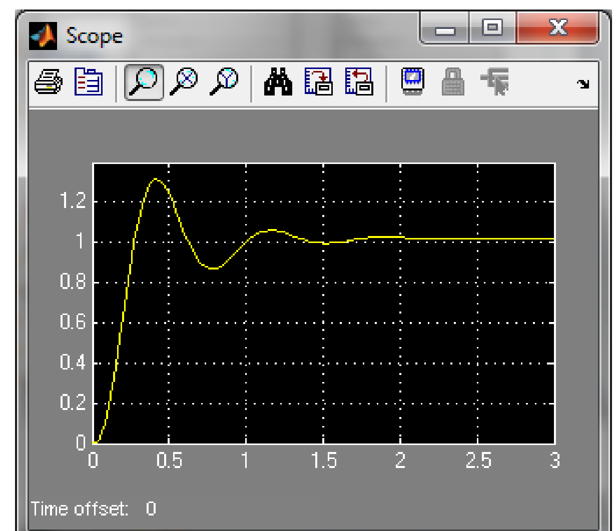
Constrained optimal lead-lag:

$$G_c(s) = 274.7764\frac{(s+164.9907)(s+165.0085)}{(s+124.4027)(s+124.4035)}$$

Stable controller

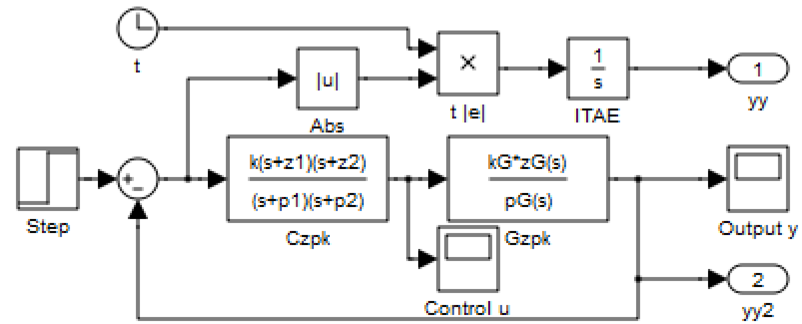
Matlab function for constrained optimal ITAE-based lead-lag:

```
function y=leadlag_opt_c(x)
assignin('base','z1',x(1));
assignin('base','z2',x(2));
assignin('base','p1',x(3));
assignin('base','p2',x(4));
assignin('base','k',x(5));
[t,xx,yy]=sim('leadlag_sim.mdl',3);
y=yy(end);
```



Optimization-Based Controller Design

Example: For $G(s) = \frac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$ design a lead-lag control to minimize the ITAE criterion with overshoot $M_p \le 3\%$



Matlab code:

```
clear all, clc,
G=zpk([-1,-0.5],[0,-0.1,-2,-10,-20],10);
[zG,pG,kG]=zpkdata(G,'v');
x0=20*ones(5,1);
[xopt,fopt,flag,iter]=fmincon ('leadlag_opt_c',x0,[],[],[],[],zeros(5,1)),
z1=xopt(1), z2=xopt(2), p1=xopt(3), p2=xopt(4), k=xopt(5),
```

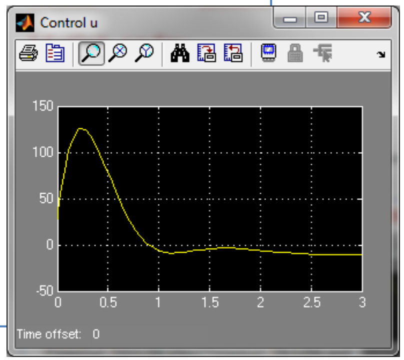
Constrained optimal lead-lag:

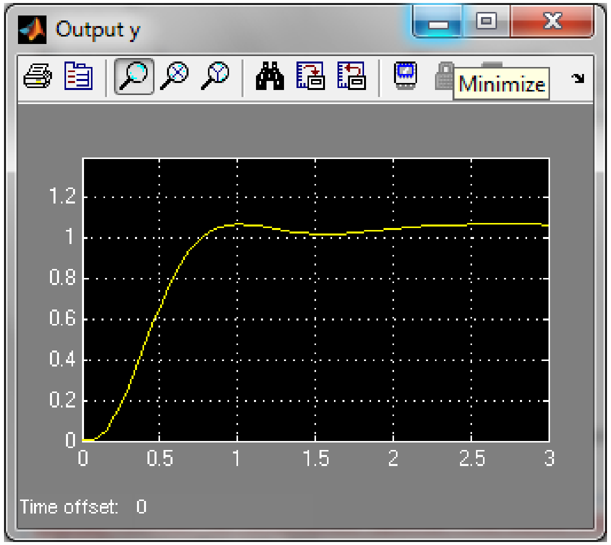
$$G_c(s) = 28.1462\frac{(s+28.0167)(s+28.0183)}{(s+11.9988)(s+11.9988)}$$

Stable controller

Matlab function for constrained optimal ITAE-based lead-lag:

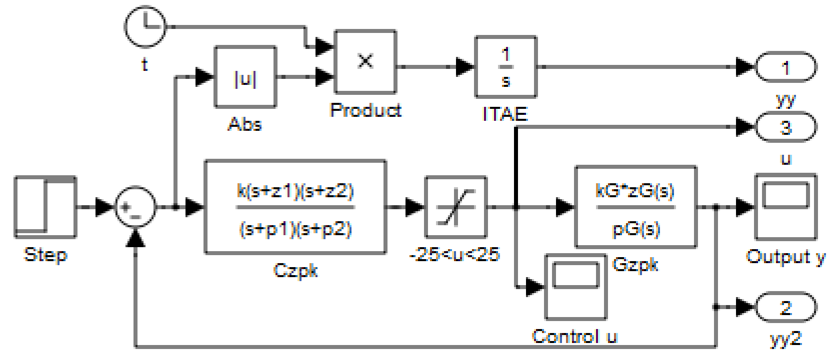
```
function y=leadlag_opt_c(x)
assignin('base','z1',x(1));
assignin('base','z2',x(2));
assignin('base','p1',x(3));
assignin('base','p2',x(4));
assignin('base','k',x(5));
[t,xx,yy]=sim('leadlag_sim.mdl',3); y=yy(end,1);
If max(yy(:,2))>1.03, y=y+2*(max(yy(:,2))-1.03); end
```





Optimization-Based Controller Design

Example: For $G(s) = \frac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$ design a lead-lag control to minimize ITAE, with overshoot $M_p \leq 3\%$ and $|u| < 25$



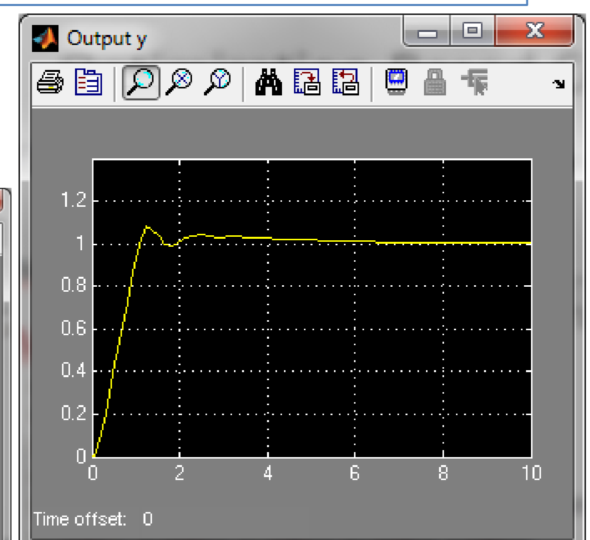
Matlab code:

```
clear all, clc,
G=zpk([-1,-0.5],[0,-0.1,-2,-10,-20],10);
[zG,pG,kG]=zpkdata(G,'v');
x0=20*ones(5,1);
[xopt,fopt,flag,iter]=fmincon ('leadlag_opt_c',x0,[],[],[],[],zeros(5,1)),
z1=xopt(1), z2=xopt(2), p1=xopt(3), p2=xopt(4), k=xopt(5),
```

Constrained optimal lead-lag:

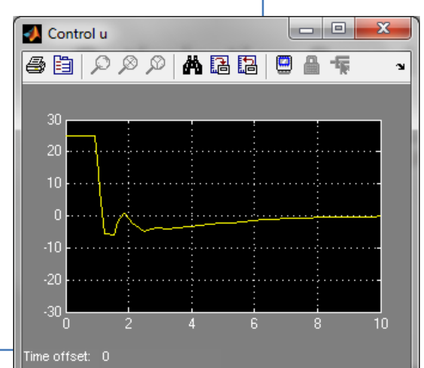
$$G_c(s) = 39.3393 \frac{(s+27.4541)(s+17.7363)}{(s+12.4818)(s+12.5655)}$$

Stable controller



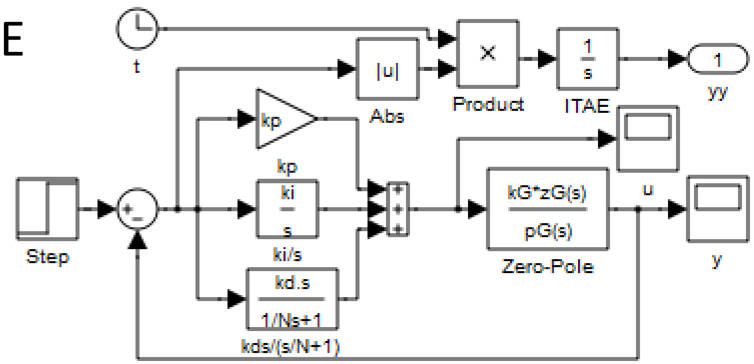
Matlab function for constrained optimal ITAE-based lead-lag:

```
function y=leadlag_sat_opt_c(x)
assignin('base','z1',x(1));
assignin('base','z2',x(2));
assignin('base','p1',x(3));
assignin('base','p2',x(4));
assignin('base','k',x(5));
[t,xx,yy]=sim('leadlag_sat_sim.mdl',10); y=yy(end,1);
If max(yy(:,2))>1.03, y=y+2*(max(yy(:,2))-1.03); end
```



Optimization-Based Controller Design

Example: For system $G(s) = \frac{1}{s(s+1)^4}$ design a PID control, with derivative filter and non-negative pole/zeros, to minimize ITAE



Matlab code:

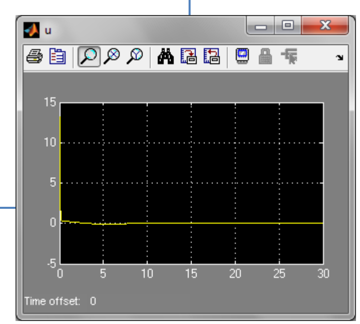
```
clear all, clc,
G=zpk([],[0,-1,-1,-1,-1],1);
[zG,pG,kG]=zpkdata(G,'v');
x0=20*ones(4,1);
[xopt,fopt,flag,iter]=fminsearch ('pid_opt',x0),
kp=xopt(1), ki=xopt(2), kd=xopt(3), N=xopt(4),
```

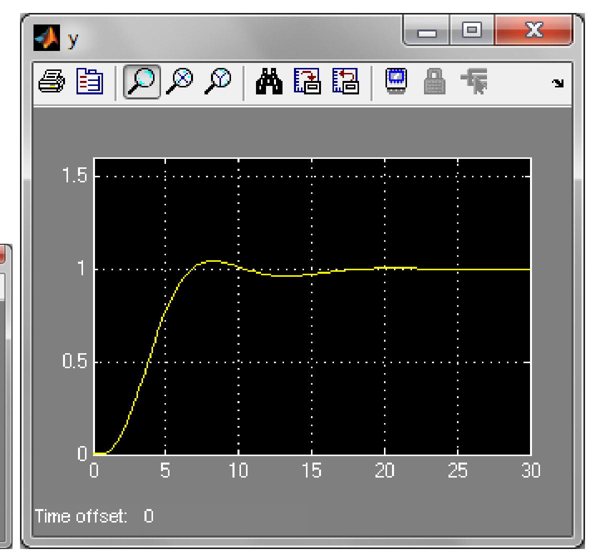
Optimal PID:

$$G_c(s) = 0.2536 + \frac{0.00009}{s} + \frac{0.696\,s}{(\frac{s}{18.5736}+1)}$$

Matlab function for optimal ITAE-based PID:

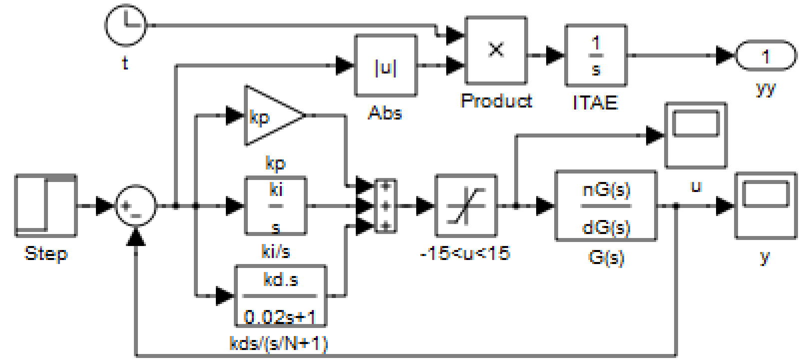
```
function y=pid_opt(x)
assignin('base','kp',x(1));
assignin('base','ki',x(2));
assignin('base','kd',x(3));
assignin('base','N',x(4));
[t,xx,yy]=sim('pid_sim.mdl',30); y=yy(end);
```





Optimization-Based Controller Design

Example: For the unstable system $G(s) = \frac{s+2}{s^4+8s^3+4s^2-s+0.4}$ design a PID control, with $|u| < 15$, to minimize the ITAE criterion



Matlab code:

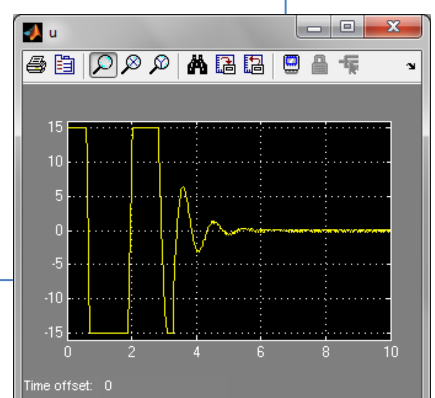
```
clear all, clc,
G=tf([1,2],[1,8,4,-1,0.4]);
[nG,dG]=tfdata(G,'v');
x0=20*ones(4,1);
[xopt,fopt,flag,iter]=fminsearch ('pid_sat_opt',x0),
kp=xopt(1), ki=xopt(2), kd=xopt(3), N=xopt(4),
```

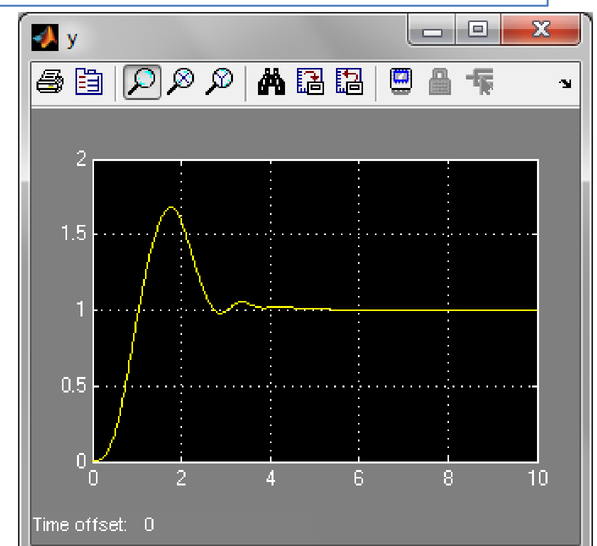
Optimal PID:

$$G_c(s) = 86.5092 + \frac{55.6594}{s} + \frac{62.8729}{(\frac{s}{212.7998}+1)}$$

Matlab function for optimal ITAE-based PID with saturation:

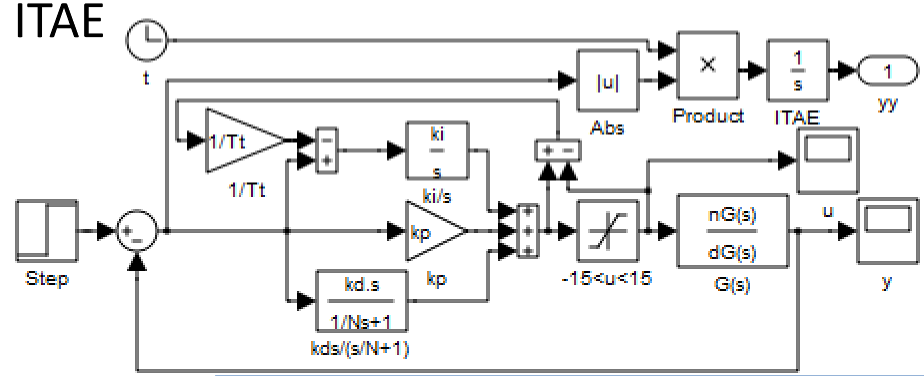
```
function y=pid_sat_opt(x)
assignin('base','kp',x(1));
assignin('base','ki',x(2));
assignin('base','kd',x(3));
assignin('base','N',x(4));
[t,xx,yy]=sim('pid_sat_sim.mdl',10); y=yy(end);
```





Optimization-Based Controller Design

Example: For $G(s) = \frac{10}{s^4+10s^3+35s^2+50s+24}$ design a PID control with anti-wind-up ($T_t = 10$), and $|u| < 15$, to minimize the ITAE



Matlab code:

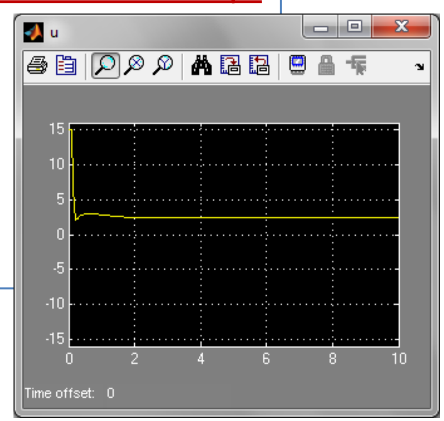
```
clear all, clc,
G=tf([10],[1,10,35,50,24]);
[nG,dG]=tfdata(G,'v');
x0=20*ones(4,1); Tt=10;
[xopt,fopt,flag,iter]=fminsearch ('pid_awu_opt',x0),
kp=xopt(1), ki=xopt(2), kd=xopt(3), N=xopt(4),
```

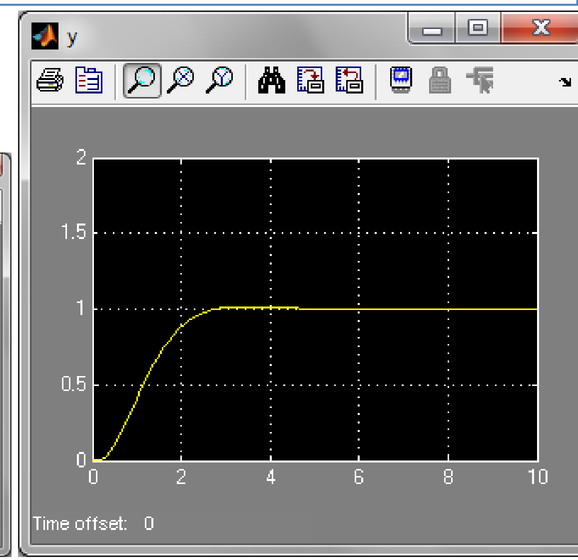
Optimal PID with anti-wind-up (T_t=10):

$$G_c(s) = 33.5125 + \frac{33.5696}{s} + \frac{12.6543}{\left(\frac{s}{30.22488}+1\right)}$$

Matlab function for optimal ITAE-based PID with anti-wind-up:

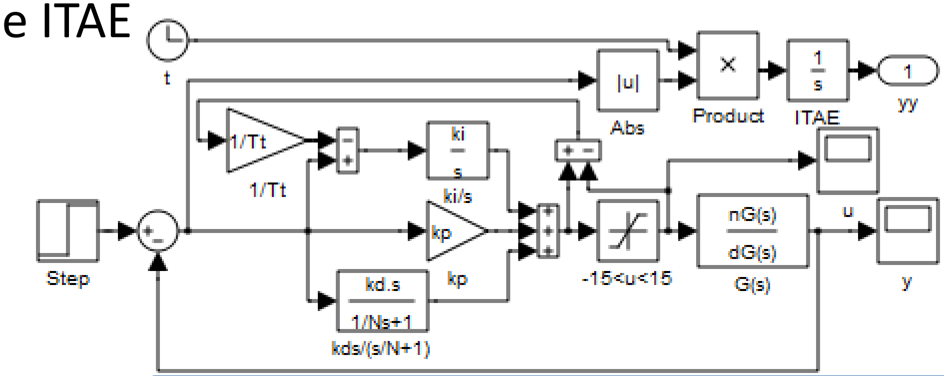
```
function y=pid_awu_opt(x)
assignin('base','kp',x(1));
assignin('base','ki',x(2));
assignin('base','kd',x(3));
assignin('base','N',x(4));
[t,xx,yy]=sim('pid_awu_sim.mdl',10); y=yy(end);
```





Optimization-Based Controller Design

Example: For $G(s) = \frac{10}{s^4+10s^3+35s^2+50s+24}$ design a PID control with tunable anti-wind-up, and $|u| < 15$, to minimize the ITAE



Matlab code:

```
clear all, clc,
G=tf([10],[1,10,35,50,24]);
[nG,dG]=tfdata(G,'v');
x0=20*ones(5,1);
[xopt,fopt,flag,iter]=fminsearch (‘pid_awu2_opt',x0),
kp=xopt(1), ki=xopt(2), kd=xopt(3), N=xopt(4), Tt=xopt(5),
```

Optimal PID with anti-wind-up (T_t=19.0207):

$$G_c(s) = 5.3598 + \frac{2.5662}{s} + \frac{2.5113}{(\frac{s}{18.5949}+1)}$$

Matlab function for optimal ITAE-based PID with tunable anti-wind-up:

```
function y=pid_awu_opt(x)
assignin('base',‘kp',x(1));
assignin('base',‘ki',x(2));
assignin('base',‘kd',x(3));
assignin('base',‘N',x(4));
assignin('base','Tt',x(5));
[t,xx,yy]=sim(‘pid_awu2_sim.mdl',10); y=yy(end);
```