UtahStateUniversity
COLLEGE OF ENGINEERING

CSOIS
center for
self-organizing and
intelligent systems

Fractional Order Controls
A Beginner's Point of View

Zhuo Li

Zhuo.li@aggiemail.usu.edu

5/15/2012

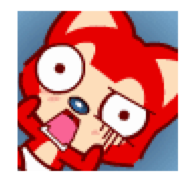
Outlines

- Key Techniques
- Backgrounds
- Distributed Order
- Approximation and Discretization
- Tools in Matlab
- IOPID, FOPI, FO[PI], DOPI and DO[PI]
- My ECE5320 MAD Report
- My Views on the 3 Tuning Equations
- Proposed Approach
- Some Results

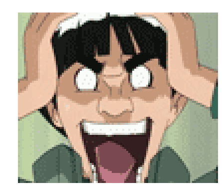
Weird Things to Me!

- For me, these are the barriers between a beginner and TD, FO, DO.
- e^{-sT} How can the "s" climb up to e? (9/2011)

- $\frac{D^{0.5}}{Dt^{0.5}} = s^{0.5} = (j\omega)^{0.5}$ First time in my life heard about FOC. (10/2011)



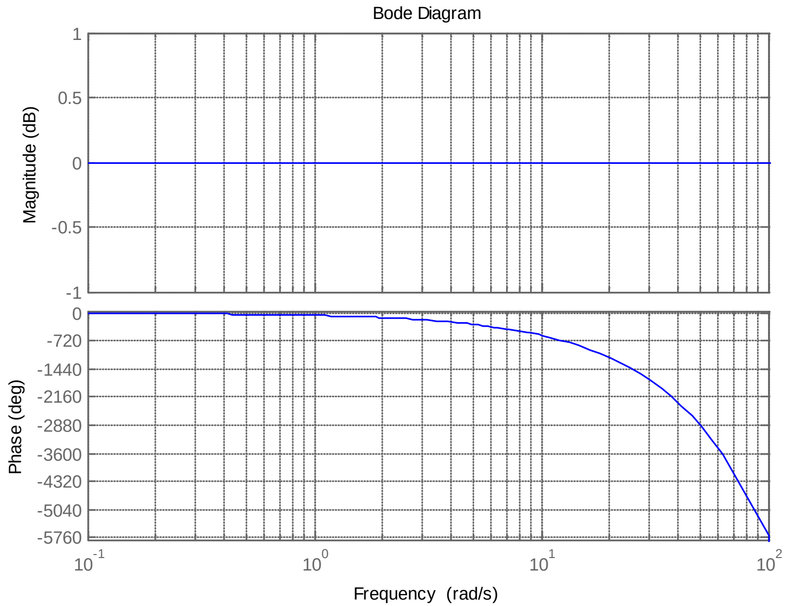
- $\int_a^b \frac{1}{s^\alpha} d\alpha = \frac{s^{-a} - s^{-b}}{\ln(s)}$ What is this? (5/2012)



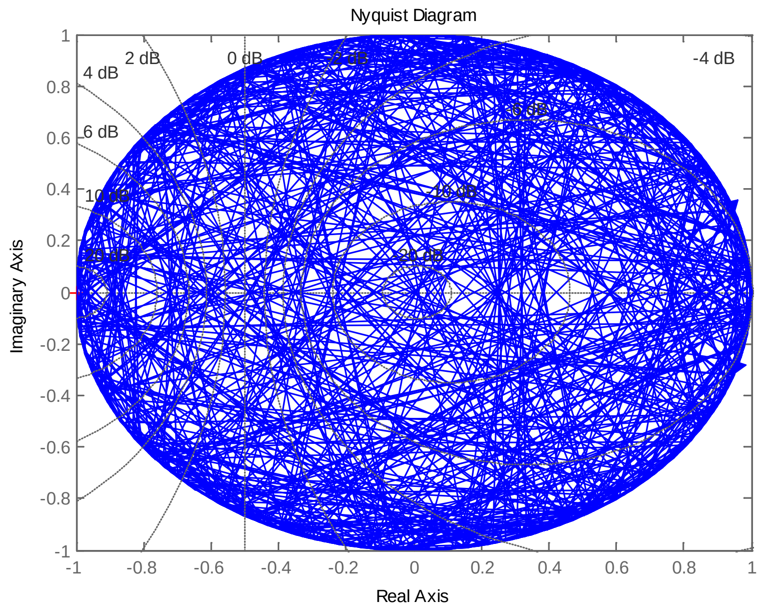
- $\ln(j\omega)$ Complex logarithm. Never heard! (5/2012)

Time Delay

Bode Plot of
$G(s) = e^{-s}$



NyquistCurve of
$G(s) = e^{-s}$



Key techniques

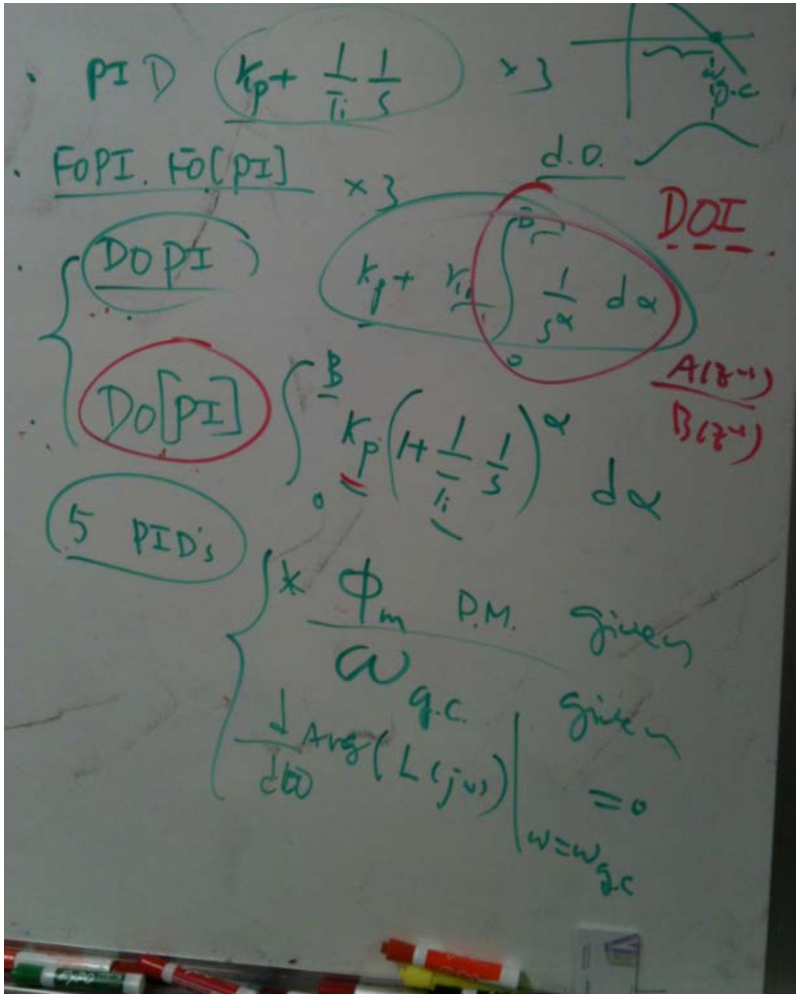
- Recall Euler's formula

$$e^{j\omega} = \cos(\omega) + jsin(\omega)$$

- $e^{s} = e^{j\omega}$ Now, we can deal with it.
- $j = \cos\left(\frac{\pi}{2}\right) + jsin\left(\frac{\pi}{2}\right) = e^{\frac{\pi}{2}j}$
- $(j)^{\alpha} \triangleq e^{\frac{\pi}{2}j\alpha} = \cos\left(\frac{\pi}{2}\alpha\right) + jsin\left(\frac{\pi}{2}\alpha\right)$(5/2012)
- $\ln(j\omega) = \ln(\omega) + \ln(e^{\frac{\pi}{2}j})$(5/2012)

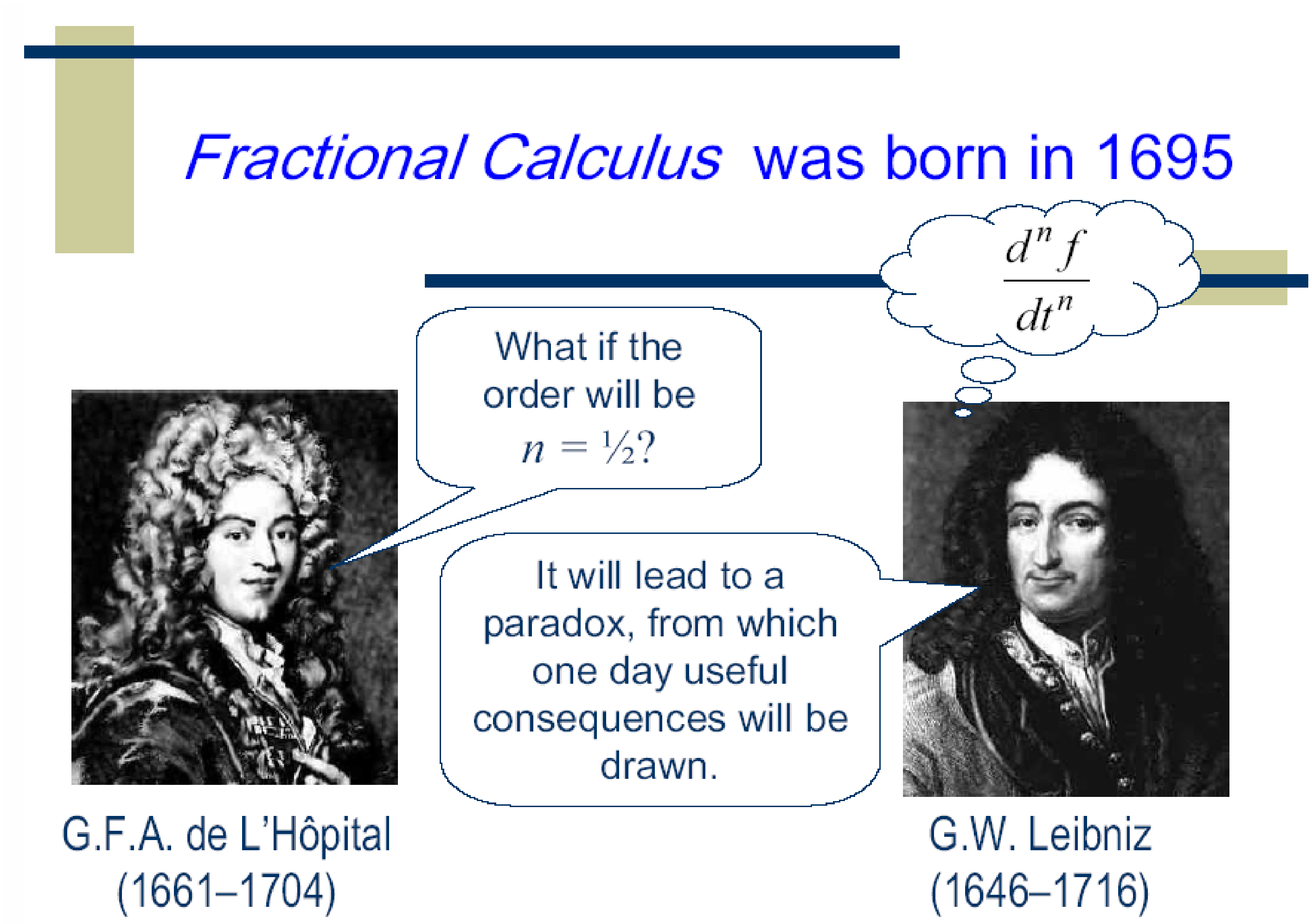
Control Engineer's View

- With these basic techniques, we can further deal with the analysis in frequency domain.
- IOPID
- FOPI^{α}, FO$[PI]^{\alpha}$
- DOPI^{α}, DO$[PI]^{\alpha}$



Board writing by Dr. Y.Q. Chen

Background



Slide from: Dr. Yangquan Chen and Dr. Igor Podlubny.

Fact

- FOC depicts most physical phenomenon more precisely.
- Superior to Integer order in a lot of cases

Three Common Definitions

- Riemann-Liouville derivative

$$ {}_aD_t^{-\alpha} f(t) = {}_aI_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau $$

- Grünwald-Letnikov derivative

$$ \mathbb{D}^q f(x) = \lim_{h \to 0} \frac{1}{h^q} \sum_{0 \le m < \infty} (-1)^m \binom{q}{m} f(x - mh). $$

- Caputo fractional derivative

$$ {}_a^C D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau) d\tau}{(t-\tau)^{\alpha+1-n}} $$

Two Important Functions

Gamma function

Here we should mention the most important function used in fractional calculus - Euler's *gamma* function, which is defined as

$$\Gamma(n) = \int_0^\infty t^{n-1} e^{-t} \, dt.$$

This function is a generalization of the factorial in the following form:

$$\Gamma(n) = (n-1)!$$

In Matlab,
y=gamma(x)

Mittag-Leffler function

Another function, which plays a very important role in the fractional calculus, was in fact introduced in 1953. It is a two-parameter function of the *Mittag-Leffler* type defined as (Podlubny, 1999):

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \qquad (\alpha > 0, \quad \beta > 0). \tag{3}$$

```
[e]=mlf(alf,bet,c,fi)
f=gml_fun(a,b,c,x,eps0)
```

Source, "Engineering Education and Research Using MATLAB", INTECK, editor Ali H. Assi, Chapter10, Ivo Petráš, 2011.

Approximating FO using IO

- Discretization

  z-transform

$$
\begin{tikzcd}
f(t) \arrow[rr, "\text{采样 } f(nT)"] \arrow[dd, "\text{拉氏变换}"', "\int_0^{\infty} f(t)e^{-sT}dt"] & & f^*(t) \arrow[rr, "\sum_{n=0}^{\infty} f(nT)z^{-1}"] \arrow[dd, "\text{采样信号拉氏变换}"', "\sum_{n=0}^{\infty} f(nT)e^{-nsT}"] & & {} \arrow[dd, "\text{Z变换}"] \\
& & & & \\
F(s) \arrow[rr, "\frac{1}{T}\sum_{-\infty}^{+\infty} F(s-jn\omega)", "\text{这个叫星号变换，也可用留数定理求得}"'] & & F^*(s) \arrow[rr, "e^{sT}=z", "\text{差值法（零阶保持，一阶保持）}"'] & & F(z)
\end{tikzcd}
$$

Methods	$s \to z$ conversion
Euler	$s^{\alpha} \approx \left[\frac{1-z^{-1}}{T}\right]^{\alpha}$
Al-Alaoui	$s^{\alpha} \approx \left[\frac{8}{7T}\frac{1-z^{-1}}{1+z^{-1}/7}\right]^{\alpha}$
Tustin	$s^{\alpha} \approx \left[\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right]^{\alpha}$
Simpson	$s^{\alpha} \approx \left[\frac{3}{T}\frac{(1+z^{-1})(1-z^{-1})}{1+4z^{-1}+z^{-2}}\right]^{\alpha}$

[♠]

[♠] Y.Q. Chen, Ivo Petr´aˇs and D.Y. Xue, “Fractional Order Control - A Tutorial”, ACC, 2009.

Discrete Approximation - PSE

- Generating function

$$\omega(z^{-1}) = 2\frac{1-z^{-1}}{1+z^{-1}}$$

$$D^{\pm\alpha}(z) = \frac{Y(z)}{F(z)} = T^{\mp\alpha}\ \mathrm{PSE}\left\{\left(1-z^{-1}\right)^{\pm\alpha}\right\} \simeq T^{\mp\alpha} P_p(z^{-1}) \tag{26}$$

Performing the PSE of the function $\left(1-z^{-1}\right)^{-\alpha}$ leads to the formula given by Lubich for the fractional integral/derivative of order α [37]:

$$\nabla_T^{\pm\alpha} f(nT) = T^{\pm\alpha} \sum_{k=0}^{\infty} (-1)^k \binom{\mp\alpha}{k} f((n-k)T). \tag{30}$$

Grünwald-Letnikov method

Another possibility for the approximation is the use of the trapezoidal rule, that is, the use of the generating function and then the PSE

$$\omega(z^{-1}) = 2\frac{1-z^{-1}}{1+z^{-1}} \tag{31}$$

The **forward** method is not suitable for cause problems

Discrete Approximation - CFE

- The continued fraction expansion (CFE) of

$$\left(\omega(z^{-1})\right)^{\pm\alpha} = \left(2\frac{1-z^{-1}}{1+z^{-1}}\right)^{\pm\alpha}$$

$$d_0(x) = f(x)$$

$$d_{k+1}(x) = \frac{x - x_k}{d_k(x) - d_k(x_k)}, \quad k = 0,1,2\ldots$$

Function f is approximated by

$$f(x) = d_0(x_0) + \cfrac{x - x_0}{d_1(x_1) + \cfrac{x - x_1}{d_2(x_2) + \cfrac{x - x_2}{d_3(x_3) + \cfrac{x - x_3}{\cdots}}}}$$

User and programmer manual, Ninteger v. 2.3 Fractional control toolbox for MatLab. Universidade Técnica De Lisboa Instituto Superior Técnico, 2005

Discrete Approximation – irid

- Impulse-response invariant discretization (IRID) method
- The approximated discrete time model matches the impulse response of the DO model precisely at every sampling point.

Realization

- FIR, IIR
- e.g.

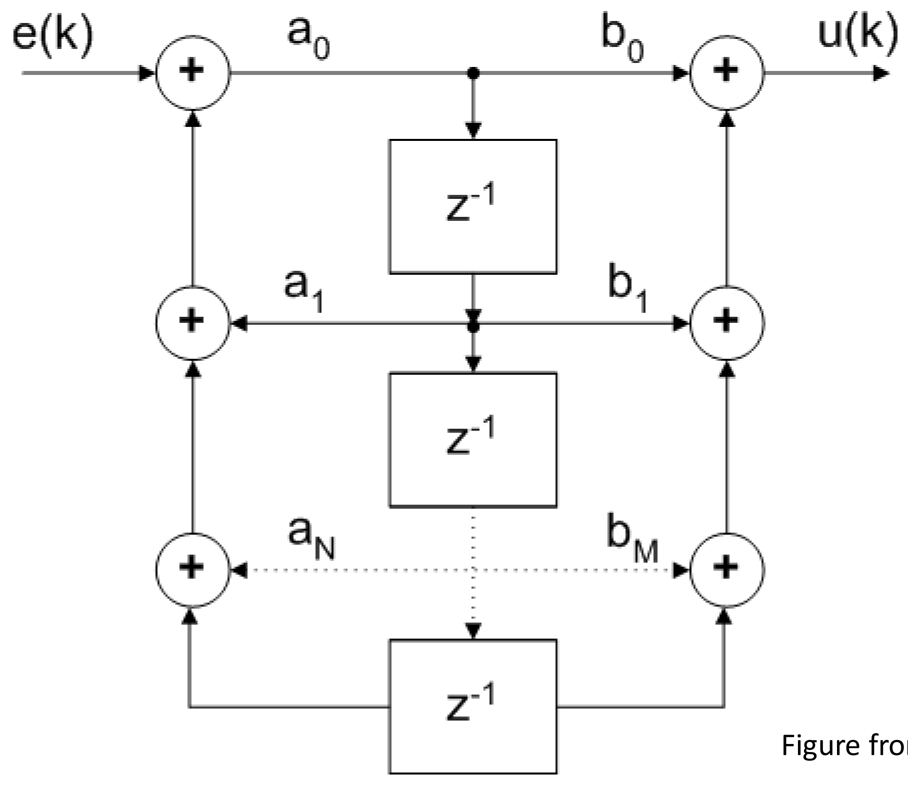


Figure from [*]

[*] Y.Q. Chen, Ivo Petr´aˇs and D.Y. Xue, “Fractional Order Control - A Tutorial”, ACC, 2009.

Distributed Order

- Definition of Distributed Order FO Integrator/differentiator [*]

$$\int_a^b \frac{1}{s^\alpha} d\alpha = \frac{s^{-a} - s^{-b}}{ln(s)}$$

- Analytical Solution [**]

$$\mathcal{L}^{-1}\left\{\int_a^b \frac{1}{s^\alpha} d\alpha\right\} = \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} e^{st}\frac{s^{-a} - s^{-b}}{ln(s)} ds$$

[*] H. Sheng, Y.Q. Chen, T.S. Qiu, "Fractional Processes and Fractional-order Signal processing", Springer, 2012.
[**] F.Y. Zhou, Y. Zhao, Y. Li, Y.Q. Chen, "An Implementation of Distributed Order PI Controller and Its Applications to the Wheeled Service Robot", ACC13 Submission.

Available Tools in Matlab

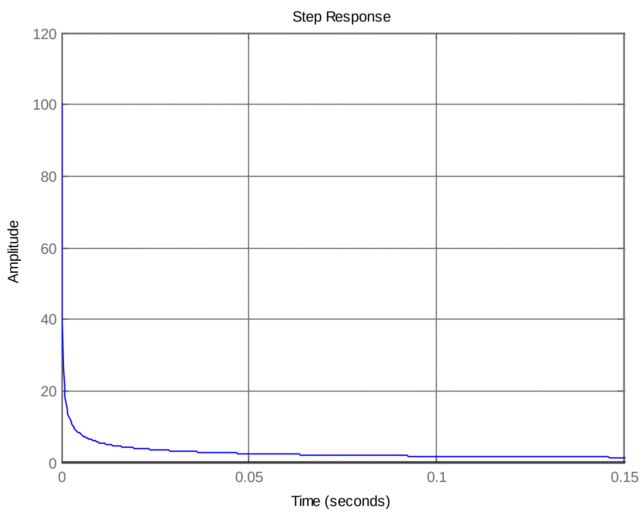
Toolbox Names	Authors
Ninteger Toolbox	Duarte Valério
Crone Toolbox	CRONE Team
@fotf Toolbox	Dr. Dingyu Xue
Irid_doi(), gml_fun, FOCP, nilt (non-integer order inverse Laplace) ora_foc(), etc.	Dr. Yangquan Chen, Hu Sheng, and Christophe Tricaud, et al.
mlf(), etc.	Dr. Igor Podlubny
dfod1(), dfod1(), dfod1(), DFOC, etc.	Dr. Ivo Petras
fderiv()	F. M. Bayat
etc.	etc.

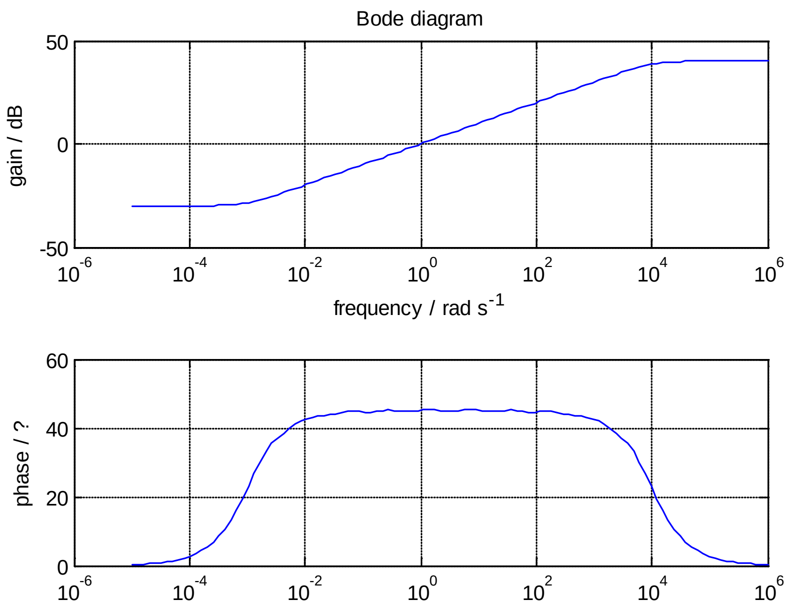
Demos

Half order differentiator $s^{0.5}$

10dB/dec

```
%% ninteger Toolbox
G_nid = nid(1, 0.5, [1e-3 1e4], 10, 'crone')
figure(5)
step(G_nid)
figure(6)
bodeFr(G_nid)
```



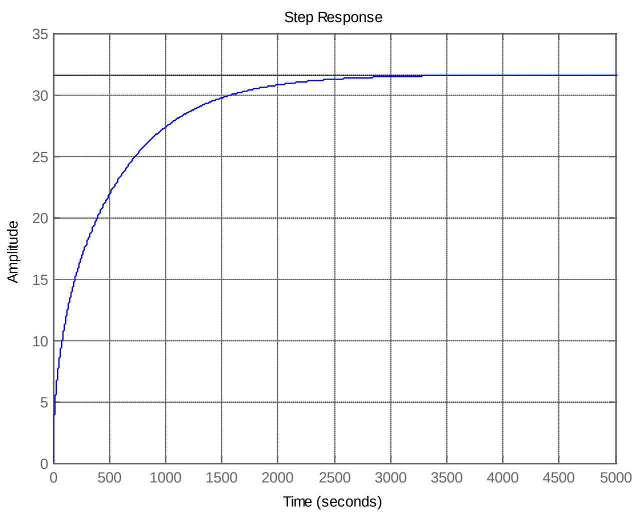


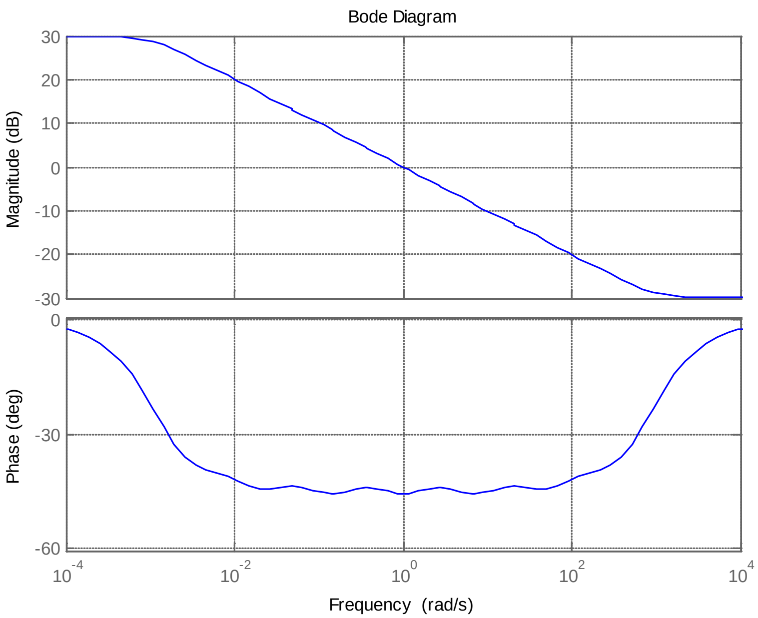
Demos

Half order itegrator $\frac{1}{s^{0.5}}$

-10dB/dec

```
%% ora_foc() by Christophe Tricaud
sys=ora_foc(-0.5,3,0.01,100)
figure(3)
step(sys)
figure(4)
bode(sys)
```

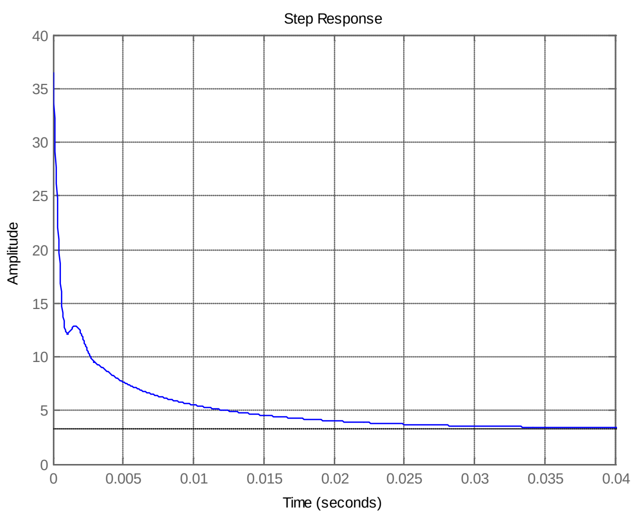


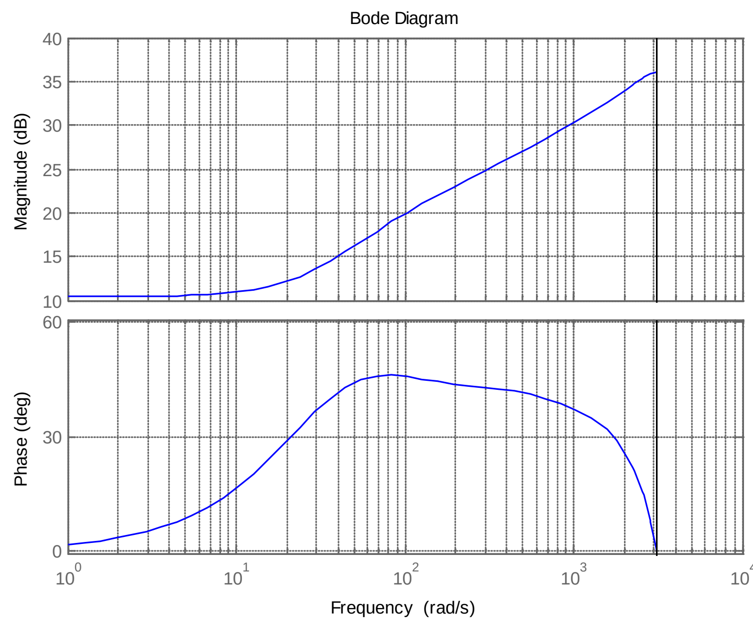


Demos

Half order differentiator $s^{0.5}$

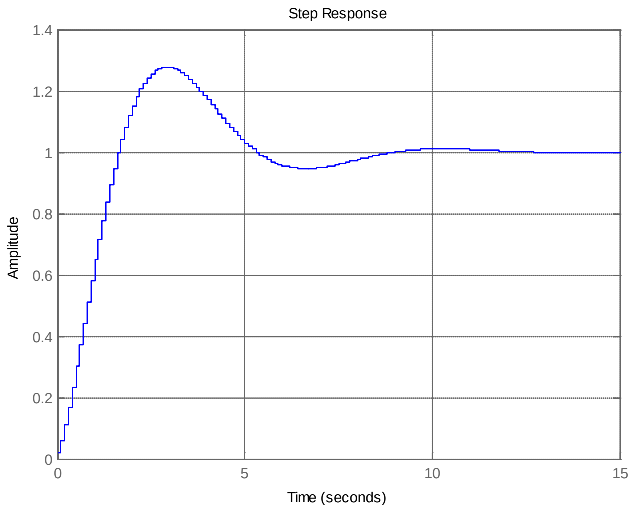
```
%% dfod1() by Dr. Ivo Petras
sys=dfod1(50, 0.001, 1/3, 0.5)
figure(1)
bode(sys)
sys_c=d2c(sys)
figure(2)
step(sys_c)
```

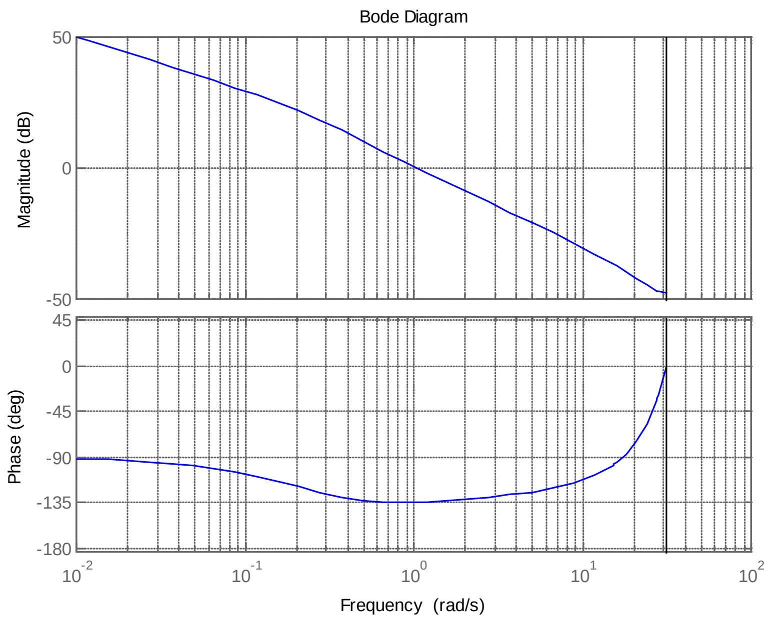




Demos

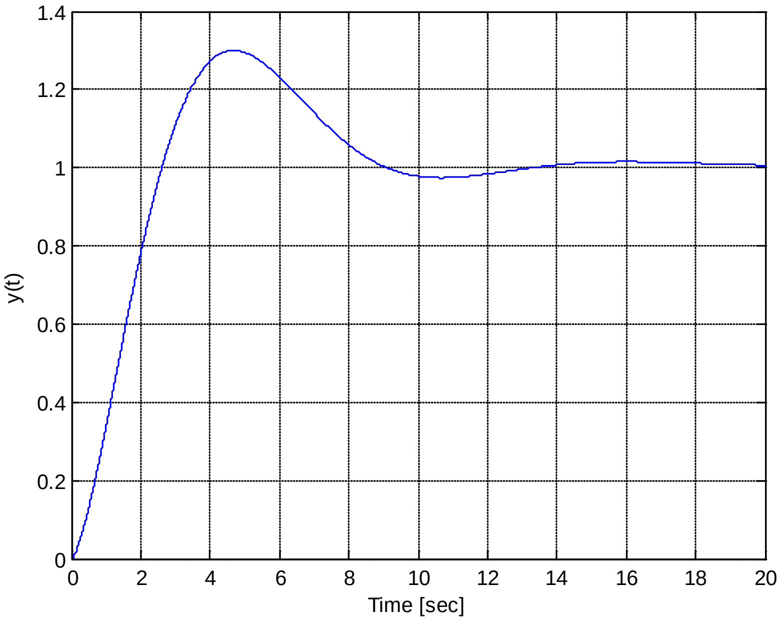
```
%% dfod1(),  Ivo's demo
T=0.1; a=1/3;
z=tf('z',T,'variable','z^-1');
Hz=((1+a)/T)*((1-z^-1)/(1+a*z^-1));
Gz_DCM=0.08/(Hz*(0.05*Hz+1));   % DC modtor plant
Cz=0.625*dfod1(5,T,a,0.5)+12.5*dfod1(5,T,a,-0.5);      % controller
Gz_close=(Gz_DCM*Cz)/(1+(Gz_DCM*Cz));
figure(6); step(Gz_close,15);grid;
Gz_open=(Gz_DCM*Cz);
figure(7); bode(Gz_open);
[Gm,Pm] = margin(Gz_open);
```





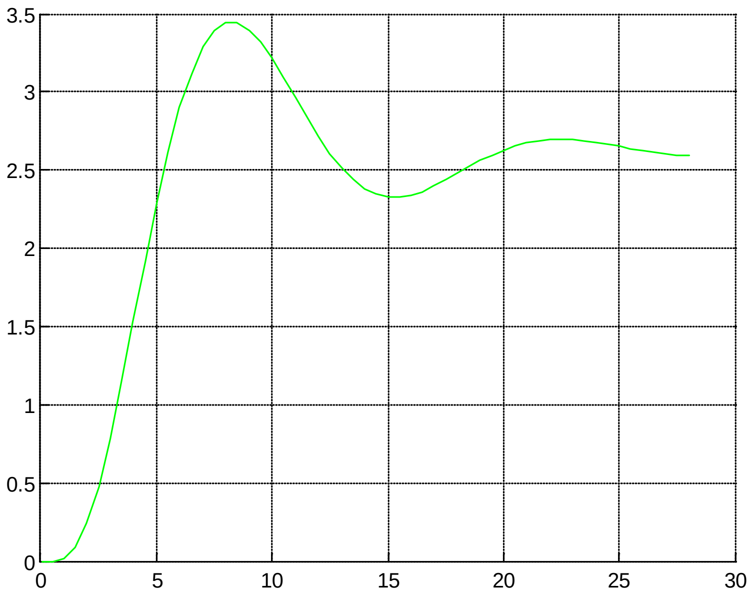
Demos

```
%% mlf(), time domain solution using
Mittag-leffler, Dr.Igor Podlubny
clear all; close all;
a=2; b=1; alpha=1.5;
t=0:0.001:20; % for time step 0.001 and
computational time 20 sec
y=(1/a)*t.^(alpha).*mlf(alpha,alpha+1,((-
b/a)*t.^(alpha)));
figure(5)
plot(t,y);
xlabel('Time [sec]');
ylabel('y(t)');grid;
```



Demos

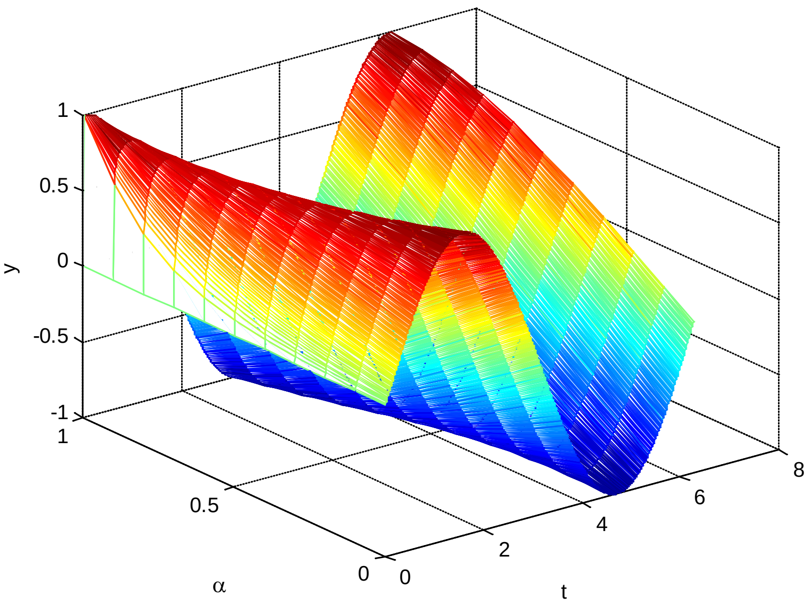
```
%% fotf()  Dr.Dingyu Xue
b = [-2 4];
a = [2 3.8 2.6 2.5 1.5];
nb = [0.63 0];
na =[3.5 2.4 1.8 1.3 0];
t = 0:0.5:28;
G = fotf(a,na,b,nb)

yi = step(G,t);
figure(6); hold on;
plot(t,yi,'g');
```



Demos

%% FO derivative sine wave, using fderiv() by Bayat

```
h=0.01; t=0:h:2*pi;
y=sin(t);
order=0:0.1:1;
for i=1:length(order)
yd(i,:)=fderiv(order(i),y,h);
end
[X, Y]=meshgrid(t,order);
mesh(X,Y,yd)
xlabel('t'); ylabel('\alpha'); zlabel('y')
```



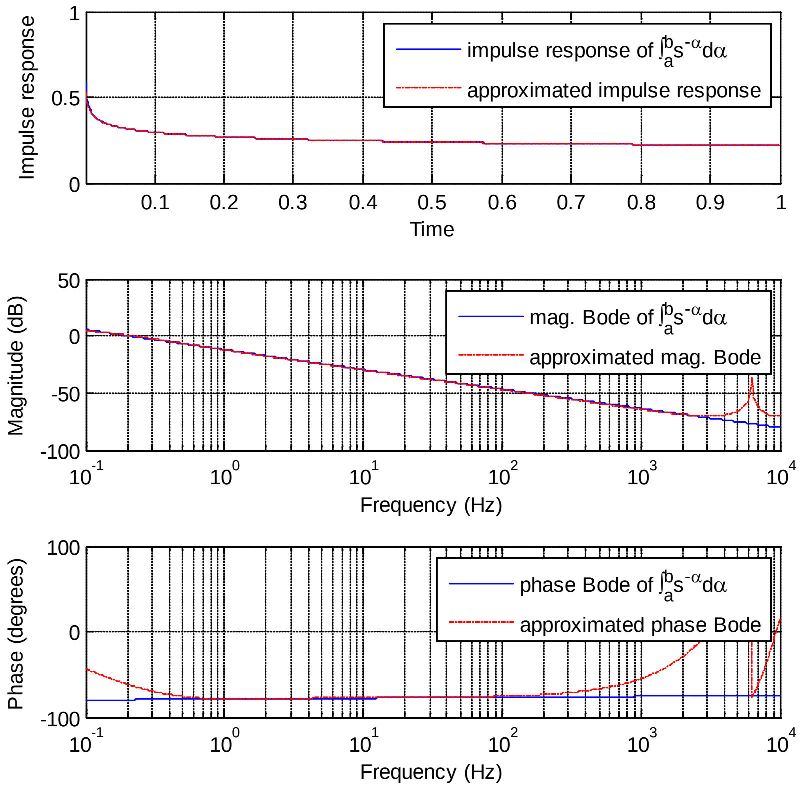
Demos

%% irid_doi(), By Dr.Y Li, Dr.YQ Chen Impulse response invariant discretization of distributed order integrator

doi=irid_doi(0.01,0.75,1,5,5);

$$\int_a^b \frac{1}{s^{\alpha}} d\alpha,$$

$\mathrm{a,b} \in (0.5, 1), a < b$



Three-parameter-tunable Controllers

- IOPID $$\mathrm{C(s)} = \mathrm{K_p} + \frac{K_i}{s} + \mathrm{K_d}\mathrm{s}$$
- FOPI $$\mathrm{C(s)} = \mathrm{K_p}(1 + \frac{K_i}{s^{\alpha}})$$
- FO[PI] $$\mathrm{C(s)} = (\mathrm{K_p} + \frac{K_i}{s})^{\alpha}$$
- DOPI $$\mathrm{C(s)} = \mathrm{K_p} + \mathrm{K}_i \int_0^b \frac{1}{s^{\alpha}} d\alpha = \mathrm{K_p} + \mathrm{K}_i \frac{1 - s^{-b}}{\ln(s)}$$
- DO[PI] $$\mathrm{C(s)} = \int_0^b \mathrm{K_p}(1 + \frac{1}{T_i s})^{\alpha} d\alpha$$

Tuning Rules

- Idea: given ϕ_{m}, ω_c, obtain flat phase at ω_c
- $\angle \mathrm{G}(\mathrm{j}\omega) = \angle \mathrm{C}(\mathrm{j}\omega_{\mathrm{c}}) + \angle \mathrm{P}(\mathrm{j}\omega_{\mathrm{c}}) = -\pi + \phi_{\mathrm{m}}$
- $|\mathrm{G}(\mathrm{j}\omega)| = |\mathrm{C}(\mathrm{j}\omega_c)\mathrm{P}(\mathrm{j}\omega_c)| = 0dB$
- $\left.\frac{d\angle \mathrm{G}(\mathrm{j}\omega)}{d\omega}\right|_{\omega=\omega_c} = 0$

Procedure

- Separate the real and imaginary part of $\mathrm{P}(\mathrm{j}\omega)$ and$\mathrm{C}(\mathrm{j}\omega)$, (for each controller $\mathrm{C}(\mathrm{j}\omega)$)
  – using the techniques on slide 4
- Calculate the gain and phase
- Solve the 3 equations numerically
  – Get Kp, Ki, α
- Digital simulation and implementation
  – Using the tools on slide 16.

My ECE5320 MAD Report

Introduction:

The heat-flow experiment (HFE) system, shown in figure 1, consists of a duct equipped with a heater and a blower at one end and three temperature sensors located along the duct [1]. The power of the heater and the speed of the fan are controlled using analog signals. Fast settling platinum temperature transducers are used to measure the temperature, and a tachometer is used to measure the fan speed.

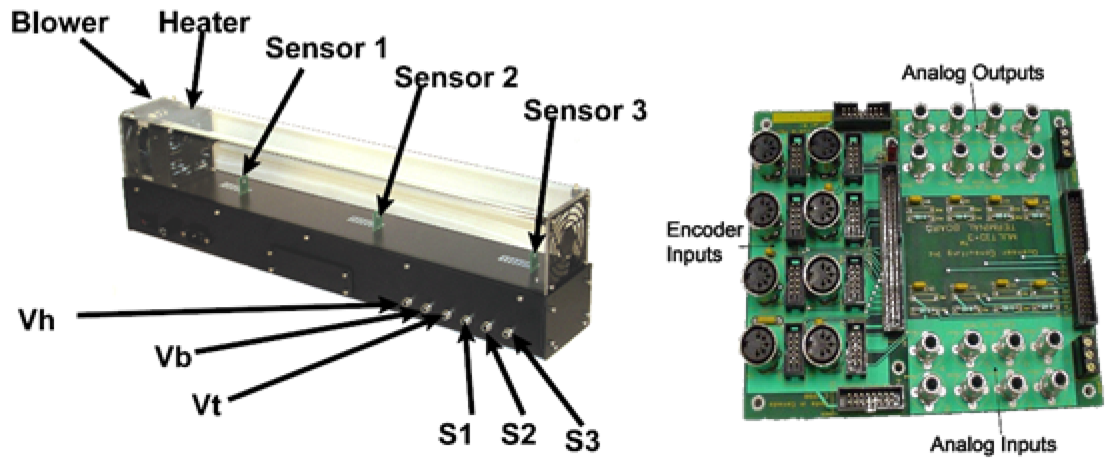


Fig1. Left: The heat-flow apparatus by Quanser Academic [1];
Right: The Q4 terminal board for connecting HFE to computers [2].

The equipment is interfaced with the computer via a Q4 terminal board shown in figure 1.

Control Objectives:

The control objective of this equipment is to make the temperature settle at the reference value fast and as smooth as possible by using various PI controllers.

My ECE5320 MAD Report

System ID by Prediction Error estimation (**PEM**) is carried out to obtain the integer order model. Since time is limited; data collection procedure is not performed. The data used here is from the MADHeatFlowQ8.zip file [2], which is measured by the temperature sensor 1 and plotted in figure 2. Figure 3 shows the comparison of the original data with the simulation data of the identified model. The parametric model along with its step response is shown in figure 4.

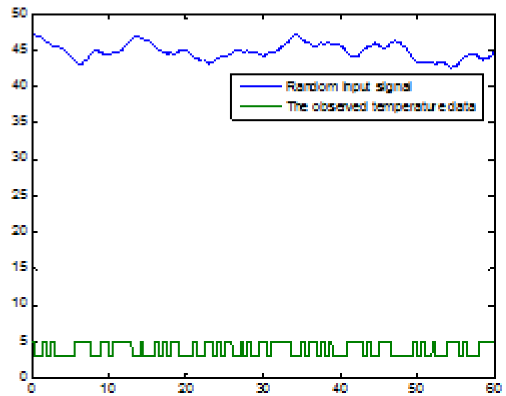


Fig2. Response of T1 to random input VQ. Note that the data is collected after running the experiment for 4 minutes in order to remove trends.

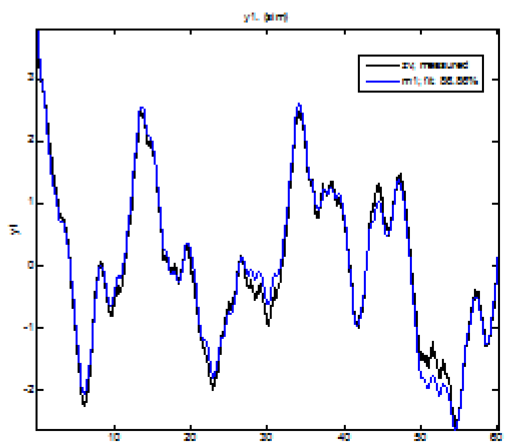
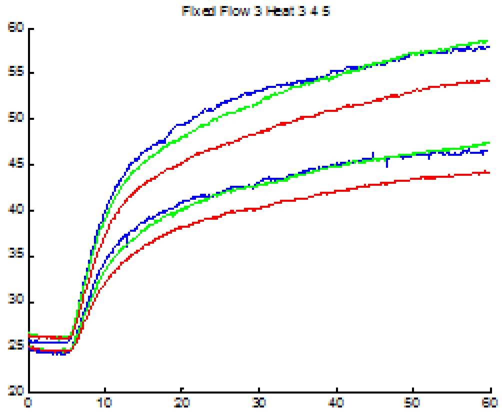


Fig3. Comparison of the original data with the simulation data of the identified model.

```
ze = dtrend(ze);
m1 = pem(ze);
zv = HF(Nstart:Nend);
zv = dtrend(zv);
figure(1)
compare(zv,m1);
```

My ECE5320 MAD Report

- First order model by curve fitting



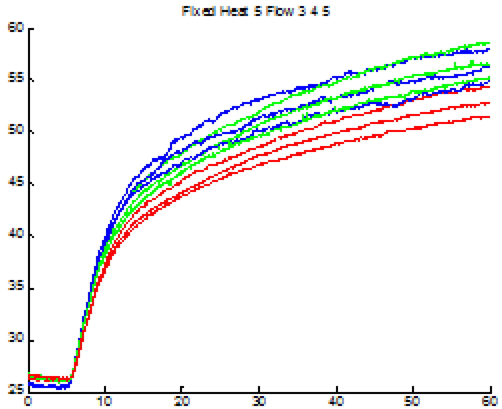


Fig5. 2 sample plots of the step response for different VQ with different flow rate voltage.

The experimental data in the zip file is plotted in figure 5. From these data, the first order model of the heat transfer at temperature **sensor 1** can be abstracted as [2]:

$$\frac{T}{V_Q} = \frac{6.4}{6.9s+1}$$

- First order plus time delay

This model takes the form below. It can be identified via nonlinear **least squares fitting** approach, which is presented in Lab2.

$$P(s) = \frac{K}{Ts+1}e^{-Ls}$$

The parameters for the model of temperature **sensor 3** is presented in [3] as $K = 38.86$, $T = 23.43$, $L = 3 \times 10^{-11}$.

For the data I collected, the parameters are estimated to be $K = 9.5$, $T = 350$, $L = 12$. The step response of this model is plotted in figure 8.

My ECE5320 MAD Report

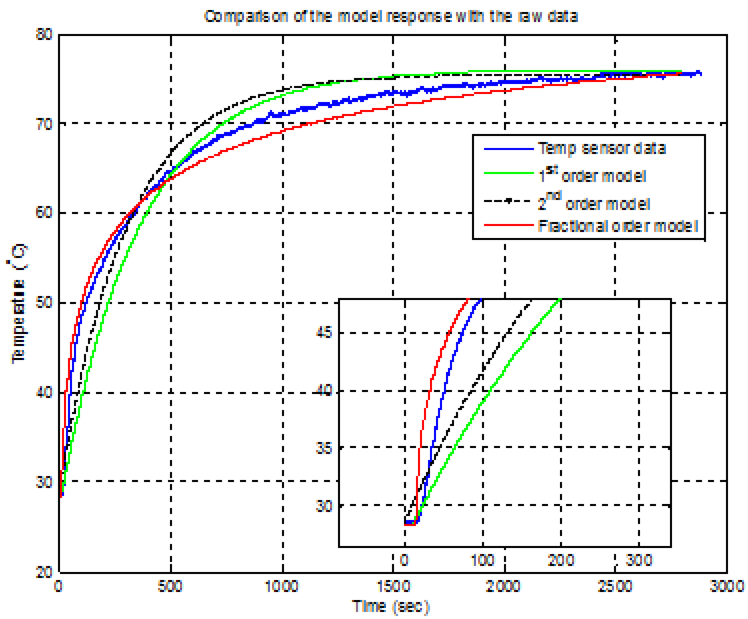


Fig8. Plot of the IO model and FO model response and the raw data

To summarize, the 3 models that we have obtained are:

Second order model	$\frac{-s+47}{s^2+300s+1}$	(1)
First order + time delay	$\frac{9.5}{350s+1}e^{-12s}$	(2)
Fractional order + time delay	$\frac{12}{20s^{0.5}+1}e^{-15s}$	(3)

My ECE5320 MAD Report

- Different controllers

With the above technique, solving the equations (4~6) will give the parameters for either integer order or fractional order controller. Computation is not performed here in detail. Instead, controllers with parameters calculated in [3] are listed below to illustrate the concept. Note: the given phase margin at cross frequency, $\omega_c = 0.08\ rad$, is $\phi_m = 50^\circ$.

Integer order PID	Kp = 0.3276, Ki = 0.6641, Kd = 52.54	$C(s) = K_p + \frac{K_i}{s} + K_d s$
Fractional order PI^α	Kp = 0.8812, Ki = 0.3153, α = 1.082	$C(s) = K_p(1 + \frac{K_i}{s^\alpha})$
Fractional order $[PI]^\alpha$	Kp=0.782, Ki=0.2927, α = 1.097	$C(s) = (K_p + \frac{K_i}{s})^\alpha$

The fractional order controller takes the form in the above table respectively. The implementation of the fractional order PI^α and $[PI]^\alpha$ controller in Simulink is provided by CRONE Toolbox as well. The Simulink connection for experiment is shown in figure 11.

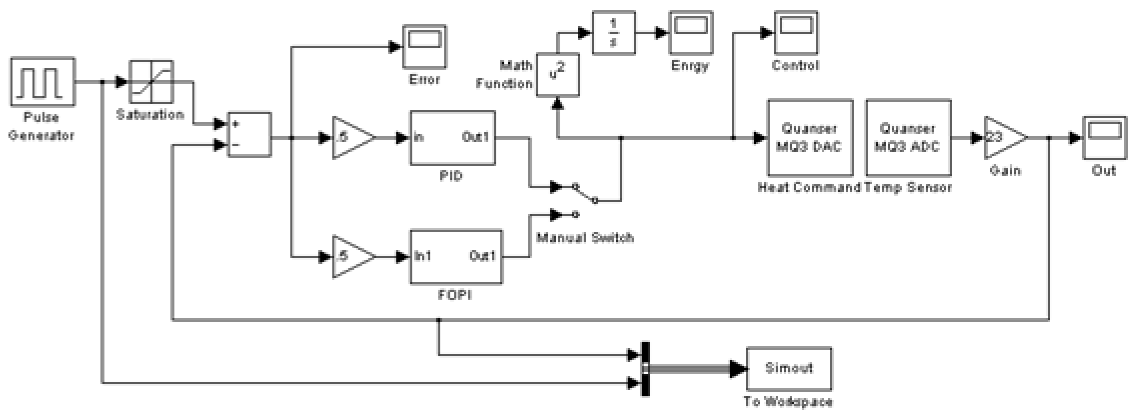
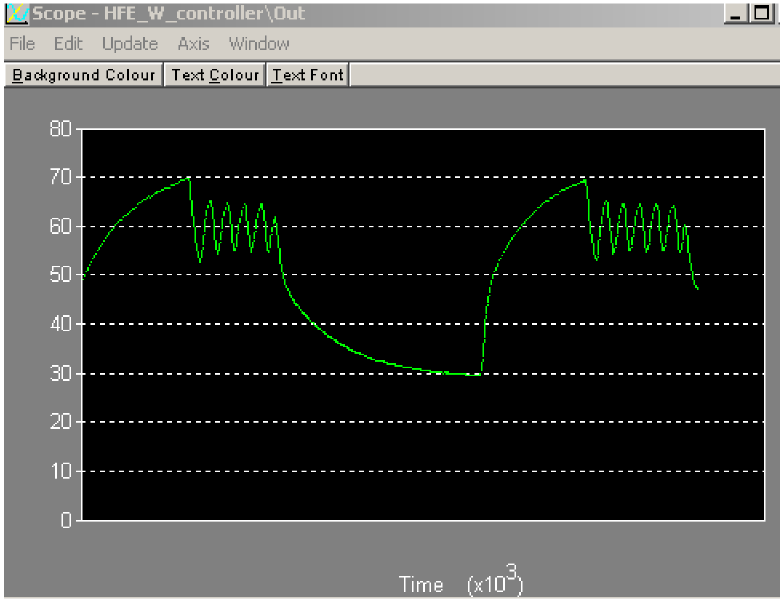
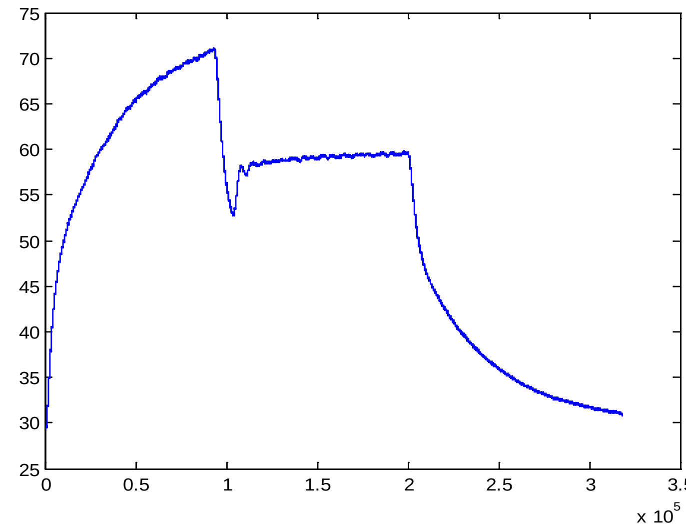


Fig11. Simulink connection of the HFE controlled by PID controller and FOPI controller

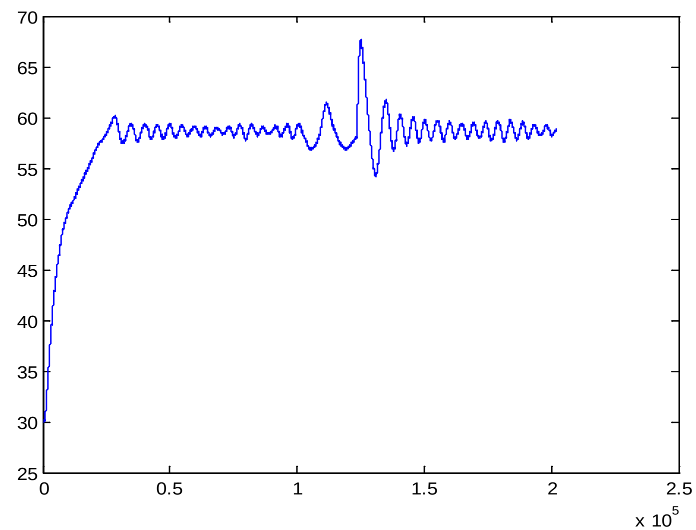
My ECE5320 MAD Report



PID



FOPI



FO[PI]

My views on the 3 Eq.

- Allowing users to specify the Gain cross-over frequency ω_c gives the user too much freedom. Barely using anyone of the above five controllers cannot always satisfy this strong design constraint.

  More specifically, **in most cases**, for sure there is an ω_c, and there is an ω_d at which the phase plot is flat.

  $$\frac{d\angle G(j\omega)}{d\omega} = 0.$$

  By adjusting the 3 tunable parameters, these two frequencies can be aligned, meaning that they match each other. However, they cannot be aligned at ***arbitrary*** frequency. As we can see from the Bode plot of the plant, the time delay pulls down the phase plot too quickly at high frequency, which gives us less time to regulate the plant. At the meanwhile, all of the 5 controllers provide limit phase compensation at high frequency, which is powerless to bring the phase back from $-\infty$. This can be seen clearly from the Bode plot of the controllers. Hence, if the desired ω_c is assigned "too late", obtaining a "flat phase" at ω_c becomes impossible.

My views on the 3 Eq.

- The above statement has shown that freely defining ω_c by user is already a "luxurious service". In spite of this, allowing users to specify the phase margin ϕ_m is another strong requirement.

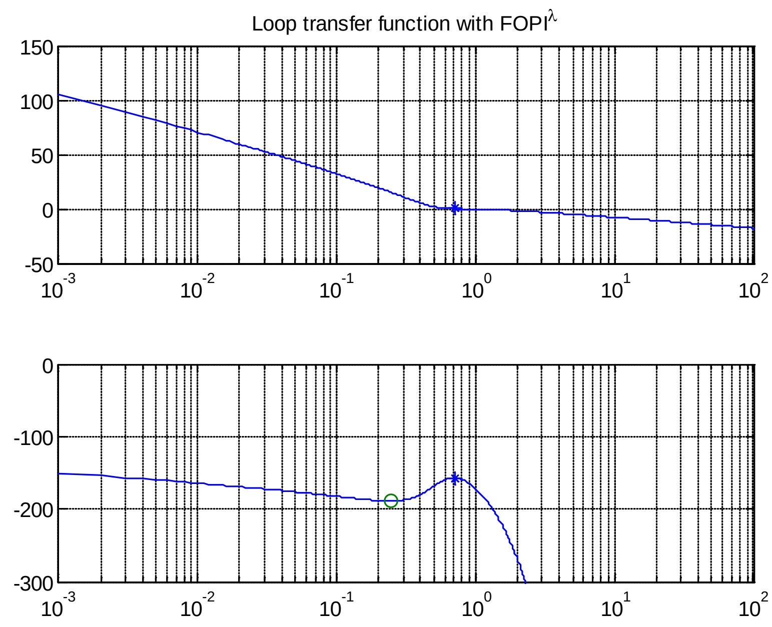
  The point where ω_c and ω_d are aligned is not guaranteed to provide the desired phase margin. The Bode plot of a given plant is fixed. The ***shape*** of the phase plot of a given controller is also determined. As an example, let's see fig 1 and fig 2. For $\lambda \in (0\sim1)$, the FOPI is able to provide -90° phase compensation ***at the most***. If the ϕ_m is assigned to be 50° at $\omega < 0.01\ rad/s$, then, it is obvious the demand cannot be meet, unless using $\lambda > 1$.

My views on the 3 Eq.

- Either one of the above 2 design specifications may make the 3 tuning equations unsolvable. Combining them together, the 3 tuning equations is possible to become **contradictory**.
- For some combinations of plant and controller, there may not exist a "flat phase".

Proposed Approach

- instead of specifying ω_c, ϕ_m, and exactly solving the 3 equations, we can try to align ω_d with ω_c first, then manually examine the phase margin.



Some Results

Fig1. The FO plant

$$P(s) = \frac{K}{Ts^a+1}e^{-Ls}$$

K =66.16;

T =12.73;

L = 1.93; % time delay

a=0.5;   % alpha

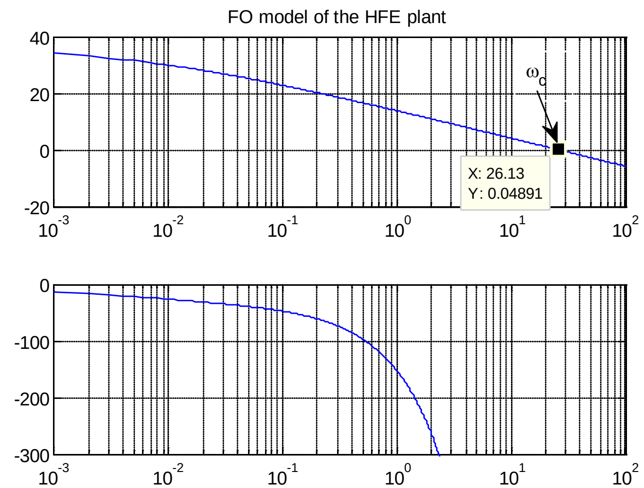
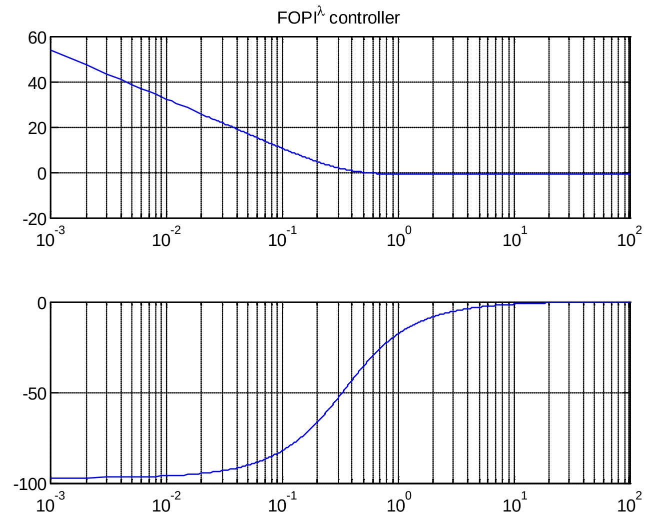


Fig2. FOPI controller with initially guessed parameters

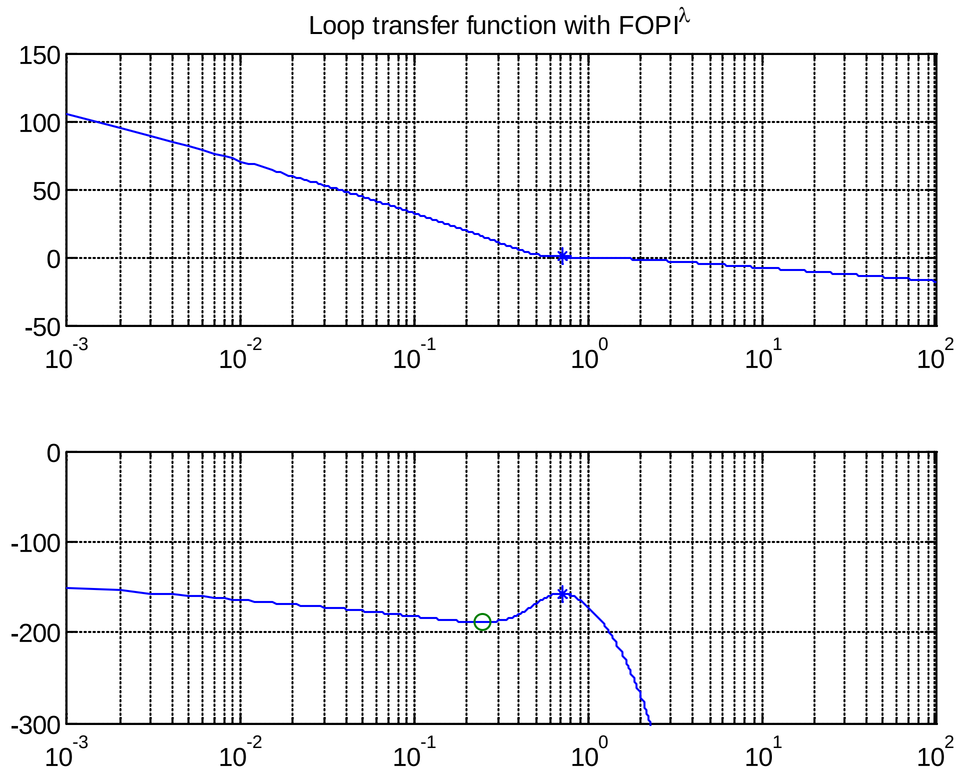
Kp_fopi=0.8812;

Ki_fopi=0.3153;

lambda = 1.082;



Some Results

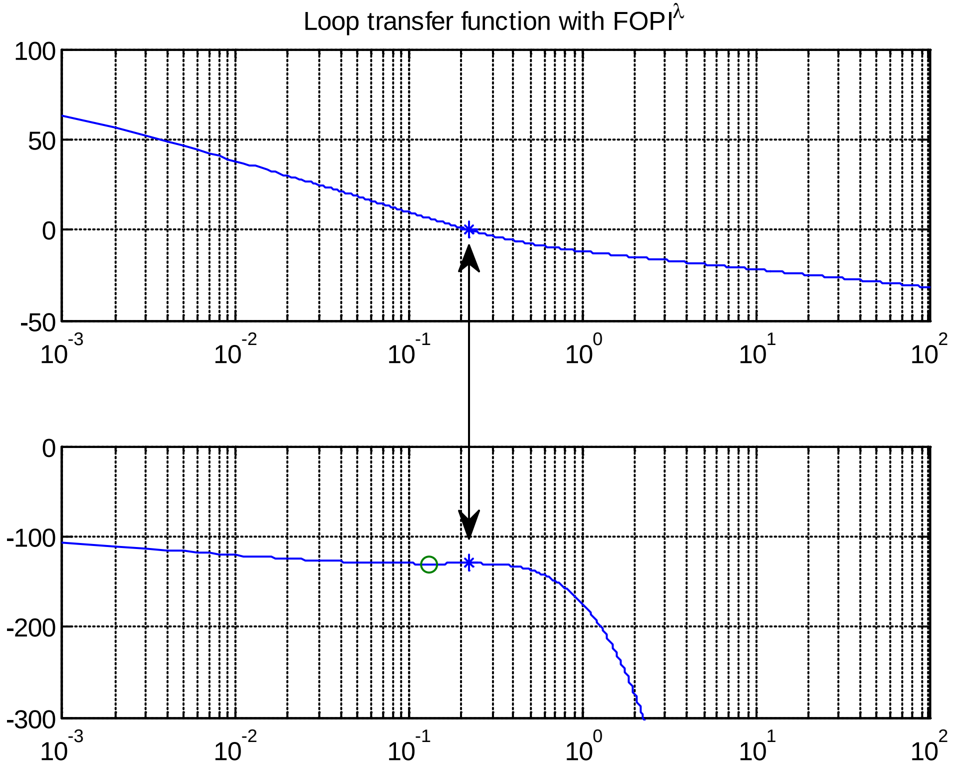


These set of parameters are obtained **without** specifying phase margin.

Kp = 0.2535 Ki = 0.3630 λ = 1.5338

Pm_fopi = 22.9815

Some Results

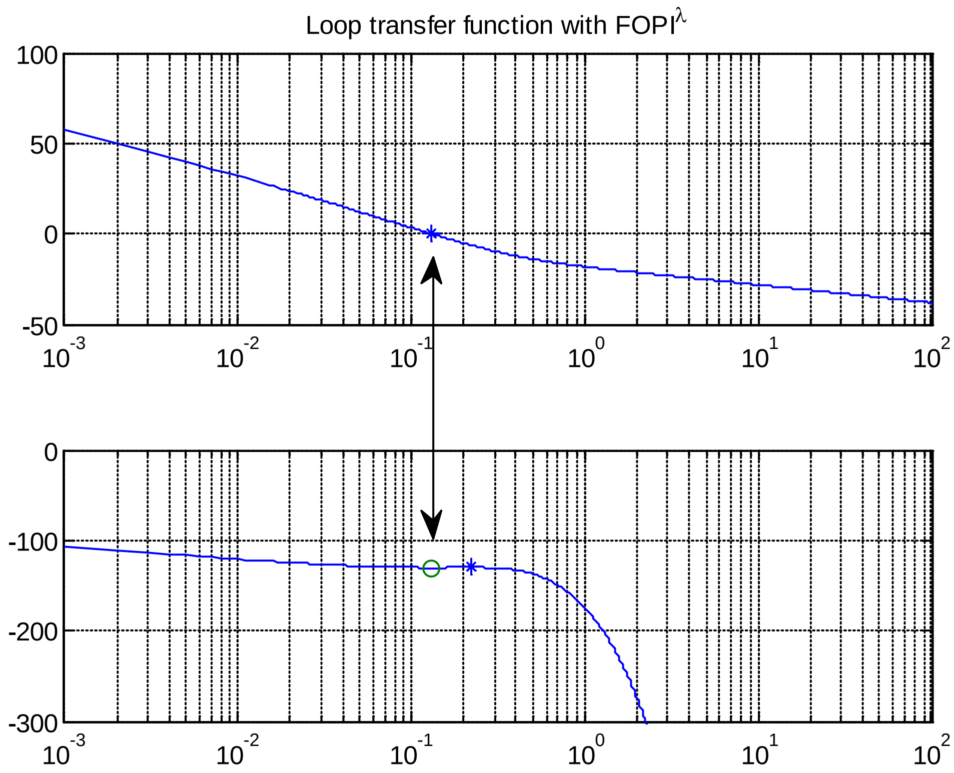


These set of parameters are obtained **with**specifying phase margin to be $\phi_m = 50^{\circ}$.

Kp = 0.0501   Ki = 0.3815 $\lambda = 1.0618$

Pm_fopi = 50.0000°.

Some Results



These set of parameters are obtained **with**specifying phase margin to be $\phi_m = 50^{\circ}$. And aligning wc with phase_min.

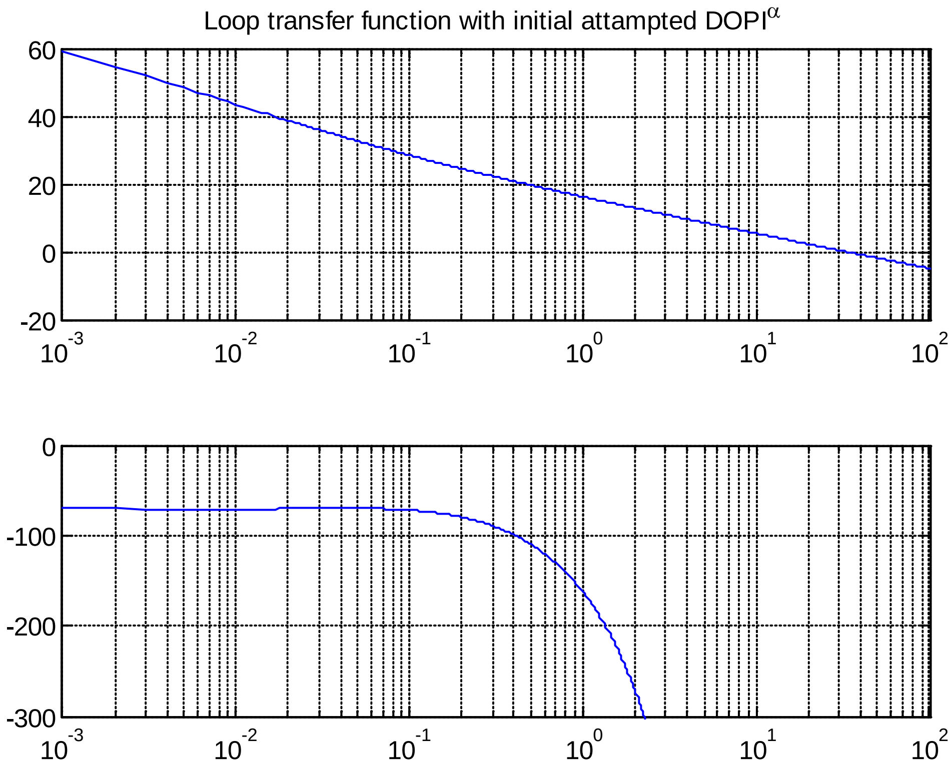
Kp = 0.0241 Ki = 0.3821 λ =1.0614

Pm_fopi = 50.0000°.

Some Results

without phase
margin constraint

Some Results



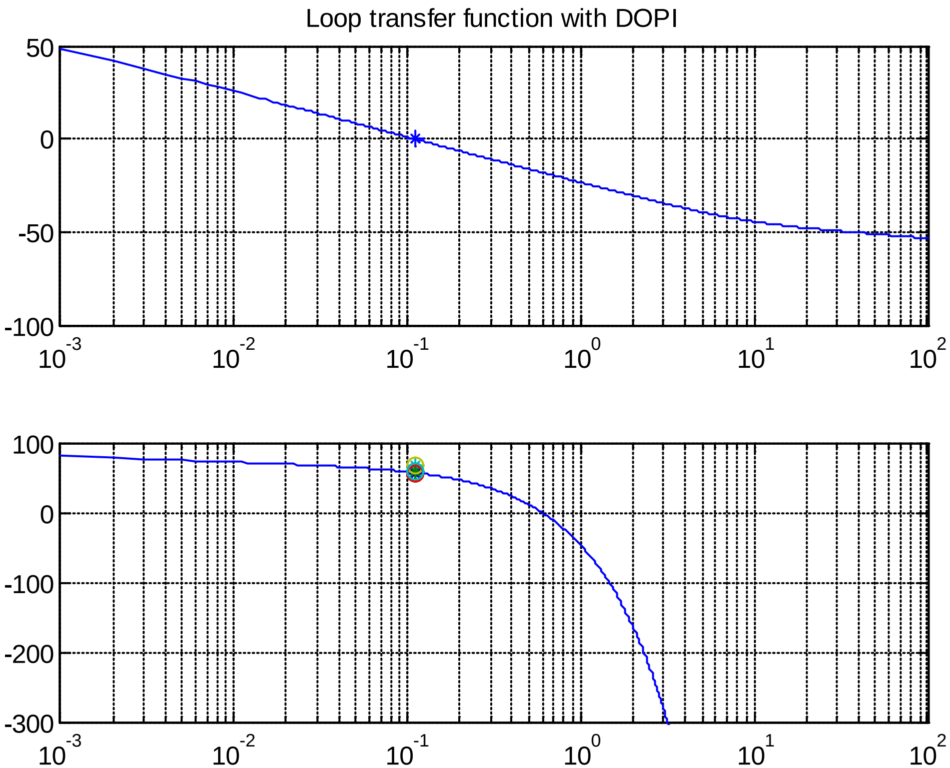
Bode plot of loop transfer function with an arbitrary guessed **DOPI**

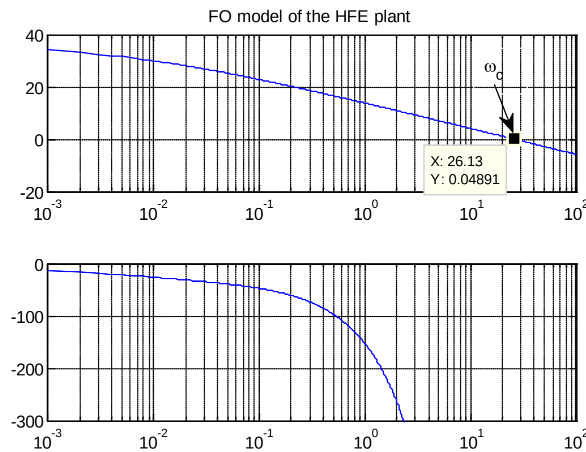
Kp_dopi=1;

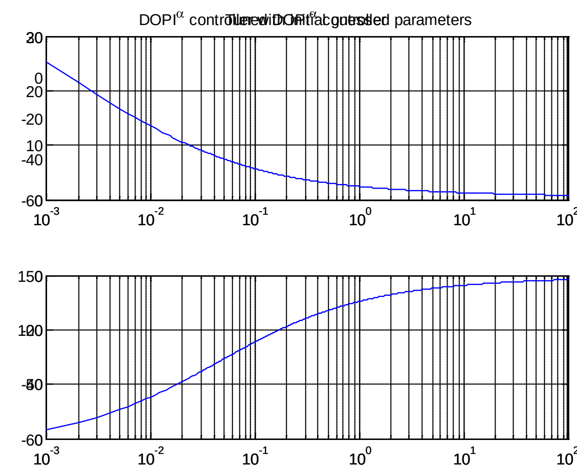
Ki_dopi=0.5;

b_dopi=0.8;

Some Results



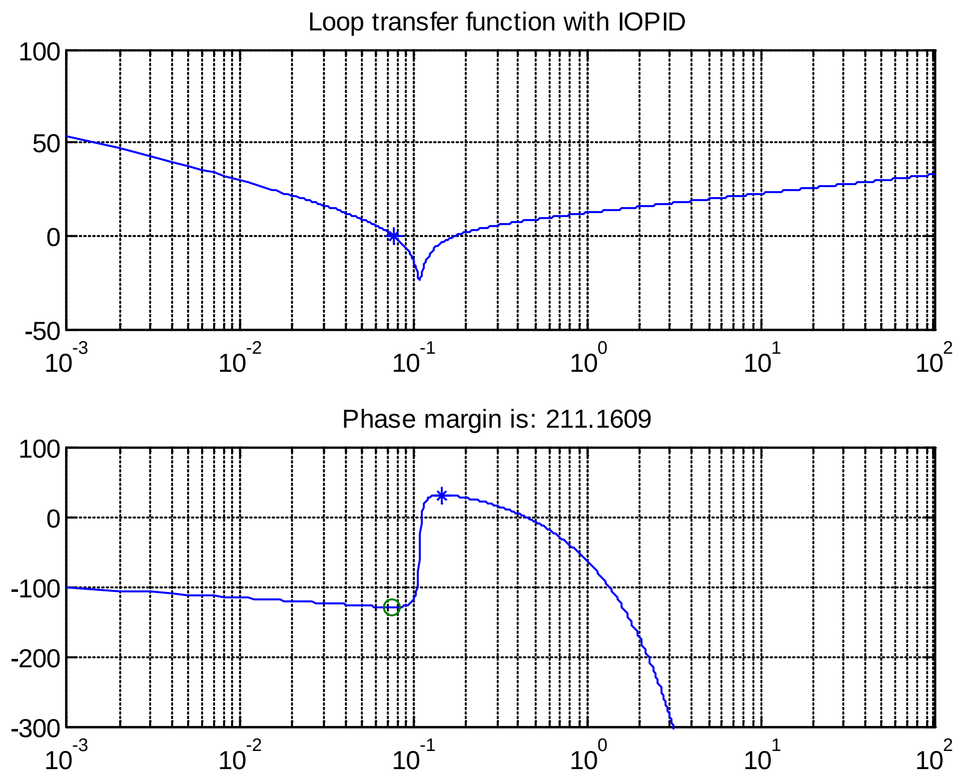




The tuning **fails** by using DOPI.
This confirms that for some plant, this tuning rule has no solution

x_dopi = 0.0075 -0.0180 1.1039

Some Results

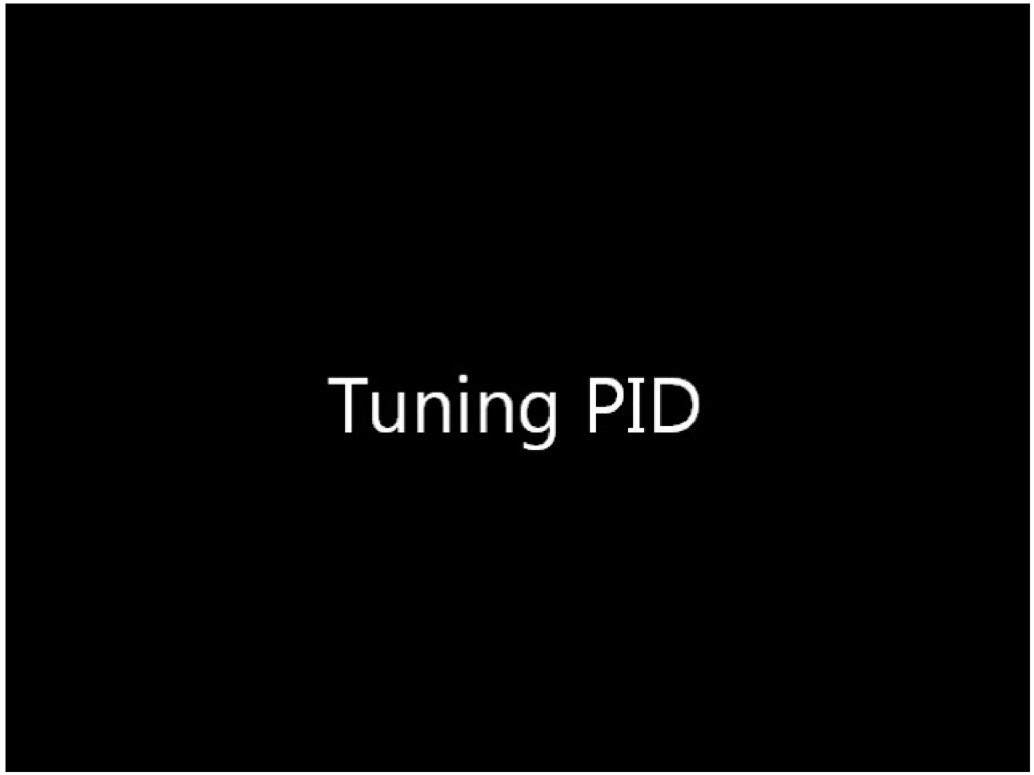


Bode plot of loop transfer function with tuned **IOPID**

x_pid = 0.0048   0.0102   0.8440

error = abs(wc-wd(end)) + abs(Pm-50)

Some Results



Bode plot of loop transfer function with tuned **IOPID**

x_pid = 0.0048   0.0102   0.8440

error = abs(wc-wd(end)) + abs(Pm-50)

Future Work

- Mathematically show for what type of plants the 3 tuning equations are not applicable
- Implementing the 5 controllers on varieties of plant models.

Acknowledgements

- Dr. Yangquan Chen
- Hadi Malek
- Dr. Luo Ying
- Calvin Coopmans

Q&A

FOPI

FO[PI]

DOPI

[PI]