Ver. 1.0.8 (March 14th , 2023)

# Benchmark problem for magnetic-head positioning control in HDD

# User manual

Investigating R&D Committee on "Precision Servo Technology for High Value-Added Mechatronic System"
The Institute of Electrical Engineers of Japan

1. Introduction
2. Magnetic-head positioning control system in HDDs
3. Benchmark problem
4. Tips

1. Introduction
2. Magnetic-head positioning control system in HDDs
3. Benchmark problem
4. Tips

## Background

The data capacity of the hard disk drive (HDD) must increase to meet the demands for data storage. As a result, we must improve the positioning accuracy of the magnetic head in the HDD for a better future.

## Objective

To encourage researches about magnetic-head positioning control, we release a benchmark problem which works on the MATLAB. We can simulate the magnetic-head positioning control system used in the latest HDDs.

## Benchmark problem

- We can download the code from the MathWorks File Exchange https://jp.mathworks.com/matlabcentral/fileexchange/111515-magnetic-head-positioning-control-system-in-hdds
- Requires: MATLAB, Control System Toolbox
- Compatible with R2019b and later releases

In this benchmark problem, we build the simulator for the popular HDDs in 2022.

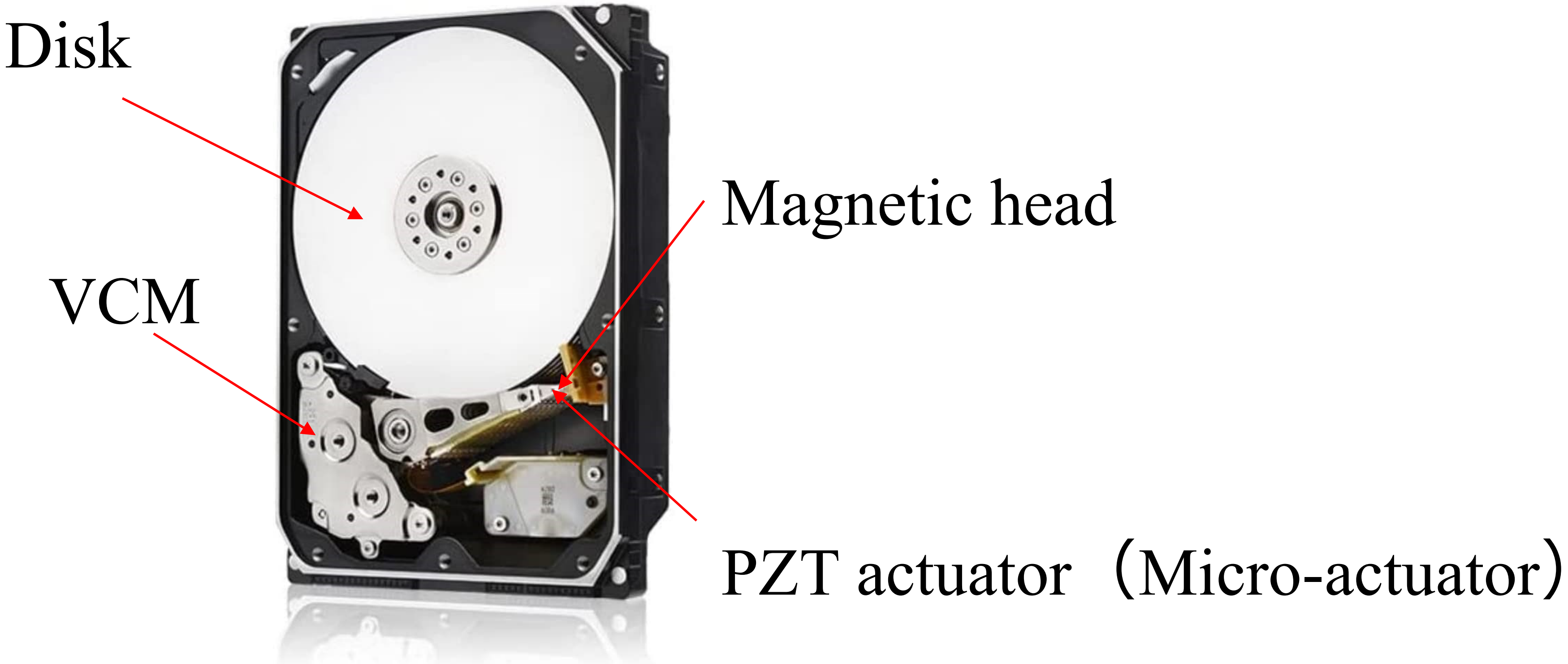
## Feature of the benchmark problem

- Platform: MATLAB (required Control System Toolbox)
- Simulated HDD: For cloud server (so-called “Nearline storage”)
  - Track pitch: 482 kTPI (52.7 nm)
  - 7200 rpm
  - Helium sealed
  - Dual-actuator system: Voice coil motor (VCM) ＋ PZT actuator
- Control design problem
  - Track-following control in storage boxes for the cloud server
  - Sampled-data control system with multi-rate filters
  - The controlled object: Dual-Input-Single-Output system

1. Introduction
2. Magnetic-head positioning control system in HDDs
3. Benchmark problem
4. Tips

# Magnetic-head positioning control system in the benchmark problem

- The control system has two actuators: VCM and PZT actuator.
- The control variable is the magnetic-head position.
- The PZT actuator has a stroke limitation (In this benchmark: 50 nm).
- We focus on the track-following control.



Disk

VCM

Magnetic head

PZT actuator（Micro-actuator）

# The mechanical characteristics of the actuators and the example controllers are given in the following paper.

T. Atsumi and S. Yabui, “Quadruple-Stage Actuator System for Magnetic-Head Positioning System in HDDs,”
The IEEE Transactions on Industrial Electronics, Vol. 67, No. 11, pp. 9184-9194, (2020-11)

9184 IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 67, NO. 11, NOVEMBER 2020

## Quadruple-Stage Actuator System for Magnetic-Head Positioning System in Hard Disk Drives

Takenori Atsumi, *Member, IEEE*, and Shota Yabui, *Member, IEEE*

***Abstract*—In this article, we present a magnetic-head positioning system in hard disk drives (HDDs) with a quadruple-stage actuator system. The quadruple-stage actuator system consists of a voice coil motor, a milliactuator with two piezoelectric (PZT) elements, a microactuator with two PZT elements, and a thermal actuator. The milliactuator is the first generation of a secondary actuator for a dual-stage actuator in the HDDs and moves a sway mode of a suspension in a head-stack assembly. The microactuator is the second generation of the secondary actuator for the dual-stage actuator in the HDDs and moves a yaw mode of the suspension. The thermal actuator is an actuator in a development phase for future HDDs. It consists of heaters embedded in the magnetic head and moves read/write elements a few nanometers in a horizontal direction with thermal expansion. The achievable disturbance-rejection performance of the proposed quadruple-stage-actuator system is higher than that of the conventional triple-stage actuator system because the proposed system can decrease a negative impact caused by the stroke limitation of the thermal actuator. As a result, the magnetic-head positioning system with the quadruple-stage-actuator system enables us to improve the positioning accuracy under the external vibrations by about 48% from that with the triple-stage-actuator system.**

***Index Terms*—Hard disks, microactuators, positioning control, servo systems.**

### I. Introduction

To INCREASE the data capacity of hard disk drives (HDDs), we have to improve the positioning accuracy of magnetic heads in the HDDs so that the size of bits for information stored on a disk decreases [1]–[3].

In the magnetic-head positioning system, one of the most critical issues is positioning errors caused by external vibrations of file servers used in data centers. Most of the positioning errors are below 2 kHz [4]. Thus, to compensate for the negative impacts of the external vibrations, we have to increase a servo bandwidth of the control system in the magnetic-head positioning system. However, the mechanical characteristics limit the servo bandwidth [5], [6]. Therefore, the magnetic-head positioning system should be fabricated with multiple actuators so that the servo bandwidth can be increased [7].

In current HDDs, a dual-stage-actuator system has been employed for the magnetic-head positioning systems. In the dual-stage-actuator system, the first actuator is a voice coil motor (VCM), and the second actuator is a piezoelectric (PZT) actuator [8]–[18]. Since 2016, we have had two kinds of PZT actuators in shipped HDDs. The first one is “milliactuator” which moves a sway mode of a suspension to control the magnetic-head position. The second one is “microactuator” which moves a yaw mode of the suspension. The frequency of the primary mechanical resonance of the microactuator is much higher than that of the milliactuator [19], [20].

For the magnetic-head positioning system, we have reported that a triple-stage actuator system was dramatically able to improve the positioning accuracy during a track-following control from the dual-stage actuator systems [21]–[23]. Since 2015, other research groups have also reported advantages of the triple-stage actuator system [24]–[26]. In the triple-stage actuator system, magnetic heads have a heater located in a horizontal direction of read/write elements in order to control the position of the read/write elements with thermal expansion induced by the heater with an electric current. However, in the triple-stage actuator system, a bottleneck of the servo bandwidth is not its mechanical resonances, but its stroke limitation.

In this article, to overcome the negative impact caused by the stroke limitation of the thermal actuator, we present a magnetic-head positioning system in an HDD with a quadruple-stage actuator system. The first stage is the VCM for moving a head-stack assembly (HSA), the second stage is the milliactuator for moving a sway mode of a suspension, the third stage is the microactuator for moving a yaw mode of a suspension, and the fourth stage is the thermal actuator. The proposed approach is verified by simulations based on measured controlled objects and a disturbance source.

### II. Quadruple-Stage Actuator System for HDD

A current shipped HDD is comprised of a cover, a base, a VCM, several PZT actuators, several magnetic heads, several

Manuscript received April 8, 2019; revised July 18, 2019; accepted November 5, 2019. Date of publication November 28, 2019; date of current version July 14, 2020. *(Corresponding author: Takenori Atsumi.)*

T. Atsumi is with the Department of Mechanical Engineering, Chiba Institute of Technology, Narashino 275-0016, Japan (e-mail: takenori.atsumi@p.chibakoudai.jp).

S. Yabui is with the Department of Mechanical Systems Engineering, School of Engineering, Nagoya University, Nagoya 464-8601, Japan (e-mail: yabui@nuem.nagoya-u.ac.jp).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2019.2955432

Frequency response of VCM

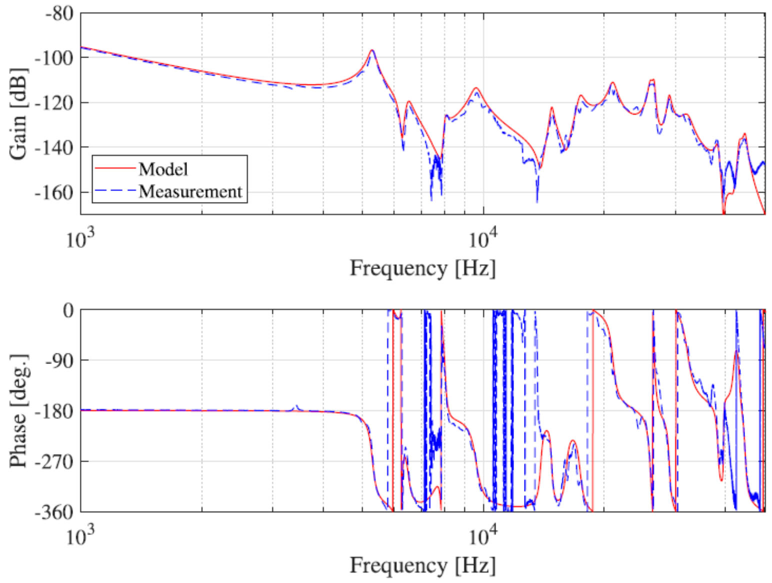


Fig. 5. Frequency response of $P_{\rm vcm}$.

Frequency response of PZT actuator

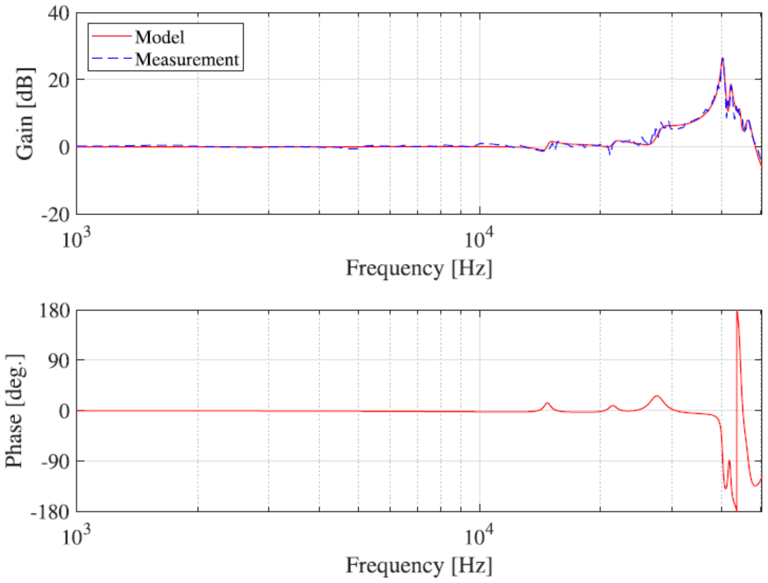


Fig. 7. Frequency response of $P_{\rm mic}$.

# Block diagram of magnetic-head positioning control system

$P_{cv}$ : VCM in continuous-time system, $P_{cp}$ : PZT actuator in continuous-time system

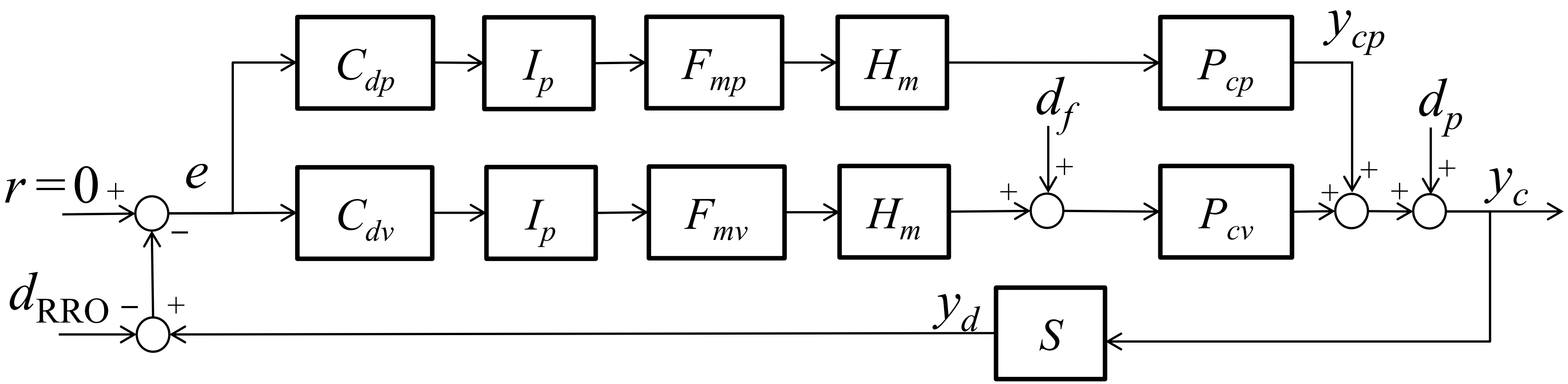
$C_{dv}$ : Feedback controller for VCM, $C_{dp}$ : Feedback controller for PZT actuator

$F_{mv}$ : Multi-rate filter for VCM, $F_{mp}$ : Multi-rate filter for PZT actuator

$I_p$ : Interpolator, $H_m$ : Multi-rate zero-order hold, $S$ : Sampler

$d_p$ : Fan-induced vibration, $d_f$: Rotational vibration (RV), $d_{\mathrm{RRO}}$ : Repeatable Run-Out (RRO)

$y_c$ : Head position in continuous time, $y_d$ : Head position in discrete time, $y_{cp}$ : Displacement of PZT actuator



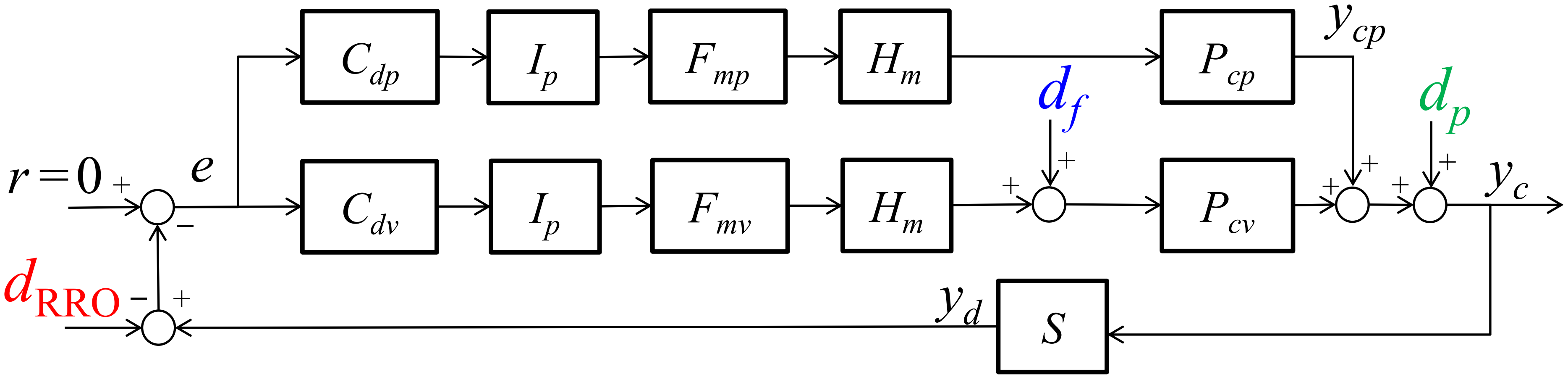
The control system is MISO sampled-data control system with multi-rate filters.

# Disturbances in the benchmark problem

- $d_{\mathrm{RRO}}$ ：RRO（Repeatable Run-Out)
  Oscillation of target tracks written on the disk. Note that it is regarded as noise.

- $d_f$：RV（Rotational vibration）
  External vibration caused by other HDDs in a storage box.

- $d_p$ ：Fan-induced vibration
  External vibration caused by cooling fans in a storage box.



1. Introduction

2. Magnetic-head positioning control system in HDDs

3. Benchmark problem

4. Tips

- Control system in the benchmark problem

- How to use the benchmark problem

- Example

- MATLAB code

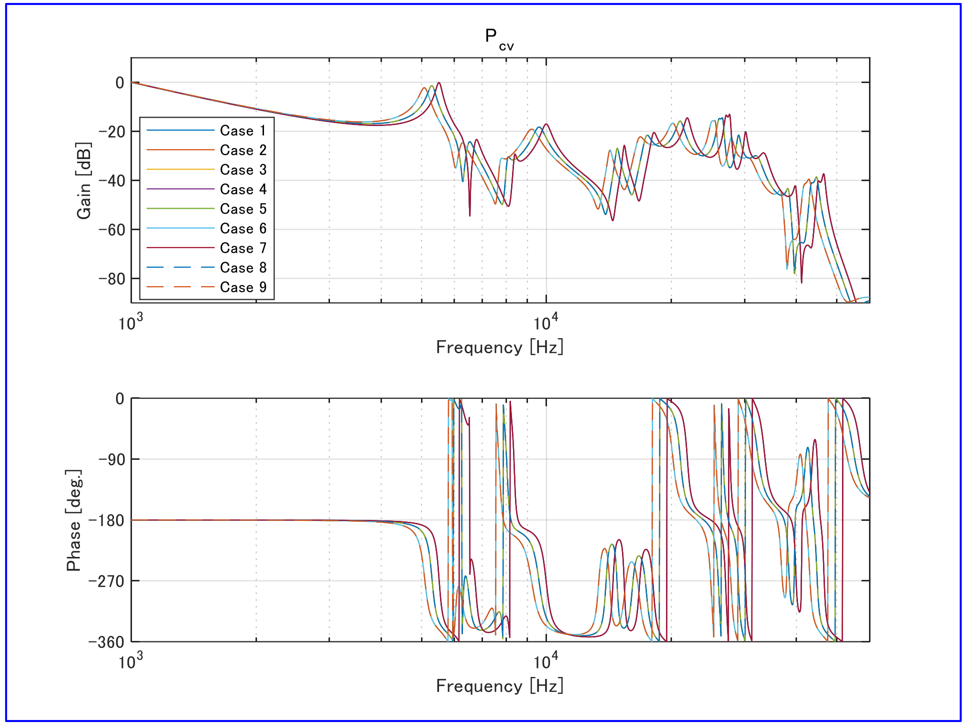
- Control system in the benchmark problem
- How to use the benchmark problem
- Example
- MATLAB code

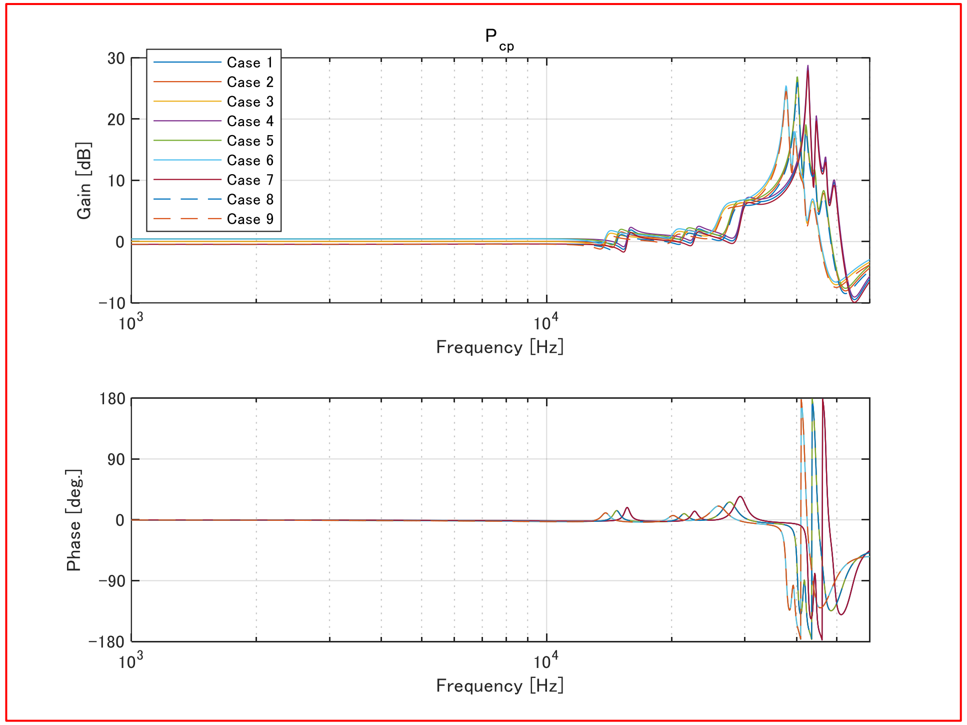
# Model of the controlled object

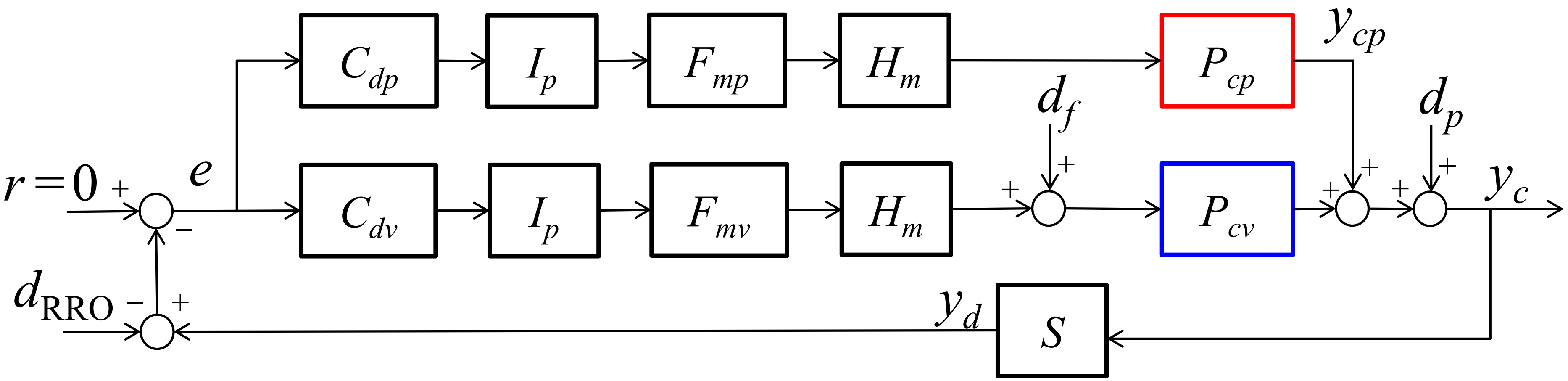
- Parameters of the nominal controlled object are shown in the following paper.
  T. Atsumi and S. Yabui, “Quadruple-Stage Actuator System for Magnetic-Head Positioning System in HDDs,”
  The IEEE Transactions on Industrial Electronics, Vol. 67, No. 11, pp. 9184-9194, (2020-11)
- Sampling time: 1/(7200/60)/420 $\approx 1.9841 \times 10^{-6}$ (7200rpm, 420 sector)
- Number of multi rate: 2
- Track pitch：482 kTPI (52.7 nm)
- Controlled object has variations
  - Temperature dependencies of mechanical resonant frequencies
    - LT(Low temp.)：VCM: +4 % from nominal model, PZT actuator: +6 % from nominal model
    - RT(Room temp.)：Same as nominal models
    - HT(High temp.)： VCM: $-4\%$ from nominal model, PZT actuator: $-6\%$ from nominal model
  - Temperature dependencies of damping ratios
    - LT(Low temp.)：VCM: -20 % from nominal model, PZT actuator: -20 % from nominal model
    - RT(Room temp.)：Same as nominal models
    - HT(High temp.)： VCM: $+20\%$ from nominal model, PZT actuator: $+20\%$ from nominal mode
  - PZT actuator’s gain
    - $\pm 5\%$
- The number of cases for controlled objects: 9

| Case No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Temp. | LT | RT | HT | LT | RT | HT | LT | RT | HT |
| PZT gain | Nominal | Nominal | Nominal | Nominal +5% | Nominal +5% | Nominal +5% | Nominal −5% | Nominal −5% | Nominal −5% |

# Frequency responses of the controlled object for all cases







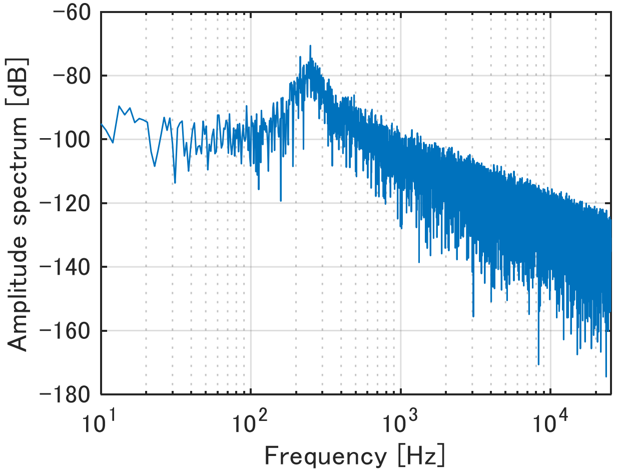
# Disturbances in the benchmark problem

- RV: Reproduced by the disturbance shown in the following paper
  T. Atsumi and S. Yabui, “Quadruple-Stage Actuator System for Magnetic-Head Positioning System in HDDs,” The IEEE Transactions on Industrial Electronics, Vol. 67, No. 11, pp. 9184-9194, (2020-11)

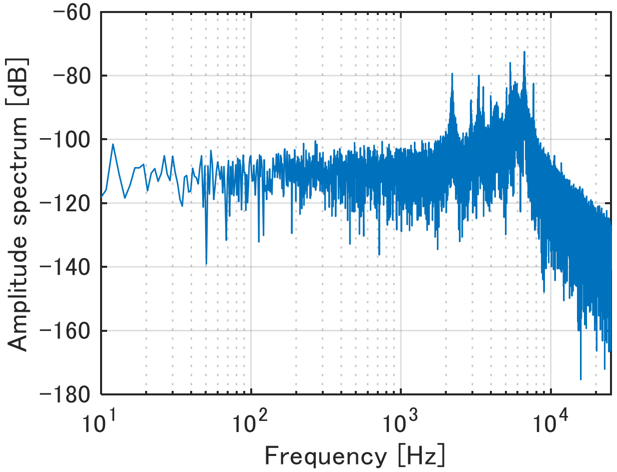
- Fan-induced vibration: : Reproduced by the disturbance shown in the following paper
  T. Eguchi, Y. Asai, K. Ichikawa, and M. Takada: ``Airborne and Structure-Borne Transmission of High Frequency Fan Vibration in a Storage Box", Proceedings of the ASME 2017 Conference on Information Storage and Processing Systems, Paper No. 5406 (2017)

- RRO: Time-domain signal which has same amplitude for all frequencies except 0 Hz.

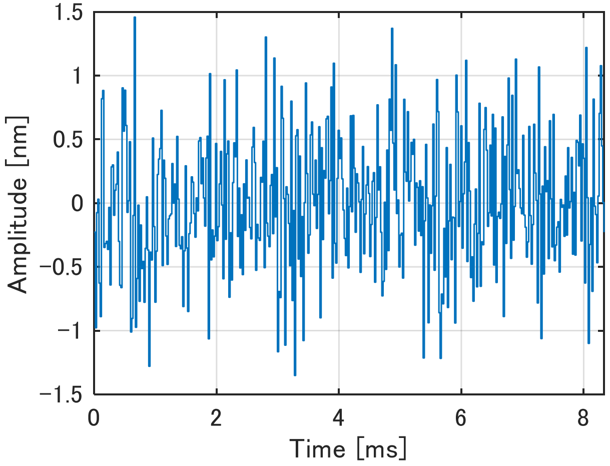
RV



Fan-induced vibe.



RRO



- Control system in the benchmark problem
- How to use the benchmark problem
- Example
- MATLAB code

# Setting

Step 1. Download the MATLAB code from MathWorks File Exchange (https://jp.mathworks.com/matlabcentral/fileexchange/111515-magnetic-head-positioning-control-system-in-hdds ) .

Step 2. Unzip the “HDDBenchmarkProblem.zip”.

Step 3. Start MATLAB.

Step 4. Set MATLAB Current Folder as the unzipped folder.

# Usage

Step 1. Define the dynamic system models (MATLAB command: “ss” or “tf”) of your designed controllers as follows.

- $C_{dv}$ : Sys_Cd_vcm (sampling time： 1/120/420 [s])
- $C_{dp}$ : Sys_Cd_pzt (sampling time： 1/120/420 [s])
- $F_{mv}$ : Sys_Fm_vcm (sampling time： 1/120/420/2 [s])
- $F_{mp}$ : Sys_Fm_pzt (sampling time： 1/120/420/2 [s])

Step 2. Save “Sys_Cd_vcm” and “Sys_Cd_pzt” as MAT file named “Data_Cd.mat”.

Step 3. Save “Sys_Fm_vcm” and “Sys_Fm_pzt” as MAT file named “Data_Fm.mat”.

Step 4. Execute “Simulation_trackfollow” which shows simulation results of the control system.

Step 5. Execute “Plot_control_system” which shows frequency responses of the control system.

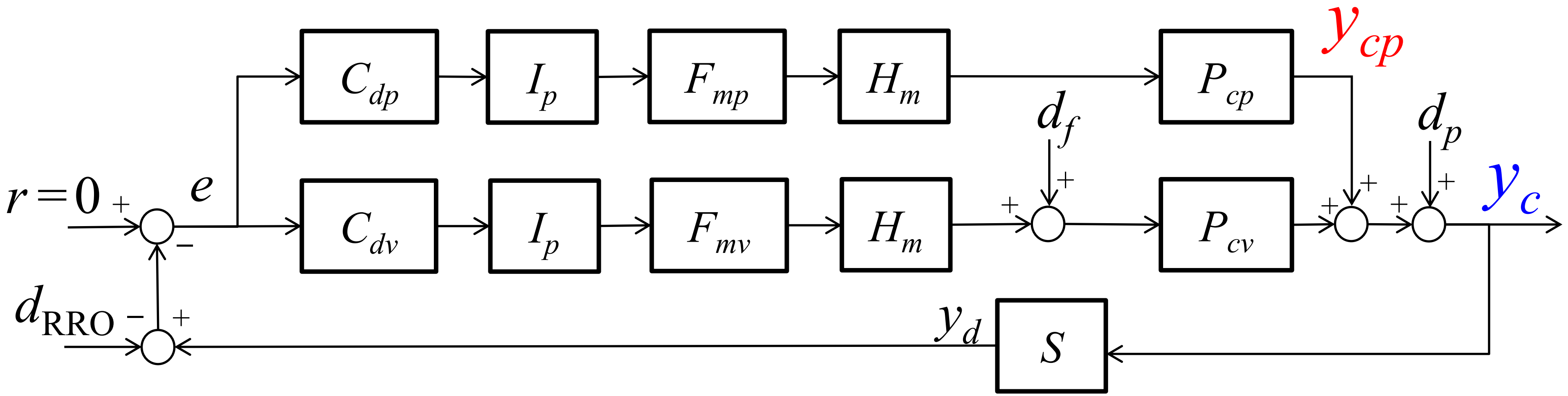
As an example, the unzipped folder includes “Data_Cd.mat” and “Data_Fm.mat” which include controllers in the following paper (Dual: VCM + Micro.).

T. Atsumi and S. Yabui, “Quadruple-Stage Actuator System for Magnetic-Head Positioning System in HDDs”, The IEEE Transactions on Industrial Electronics, Vol. 67, No. 11, pp. 9184-9194, (2020-11)

# Validation of control system

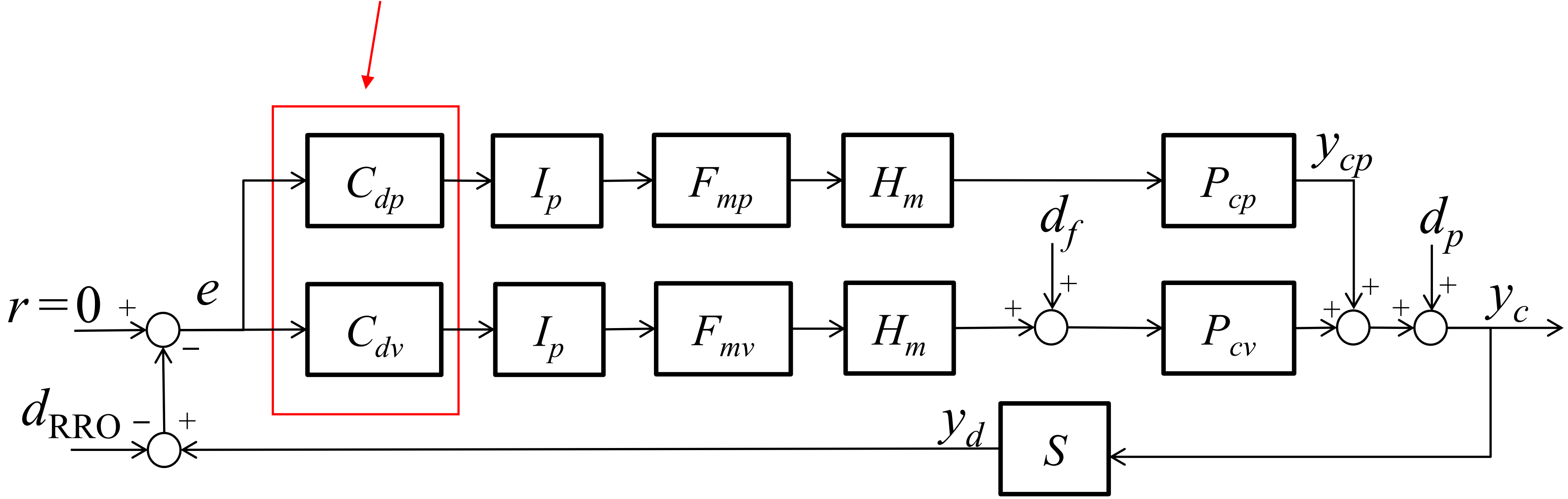
- We evaluate the worst case of 3$\sigma$ (three times of STD) value of $y_c$ in steady state response for one second.

- Maximum value of $|y_{cp}|$ must be smaller than 50 nm.



# Additional information

If you will employ controllers which are not LTI systems, you can rewrite from 109th to 116th lines in “Function_simulation.m”. The code is explained in P.38.



Rewrite the MATLAB code “Function_simulation.m”  by yourself.

- Control system in the benchmark problem

- How to use the benchmark problem

- Example

- MATLAB code

# Results of “Simulation_trackfollow” with the example controller

Figure 1: Amplitude spectrum of $d_f$

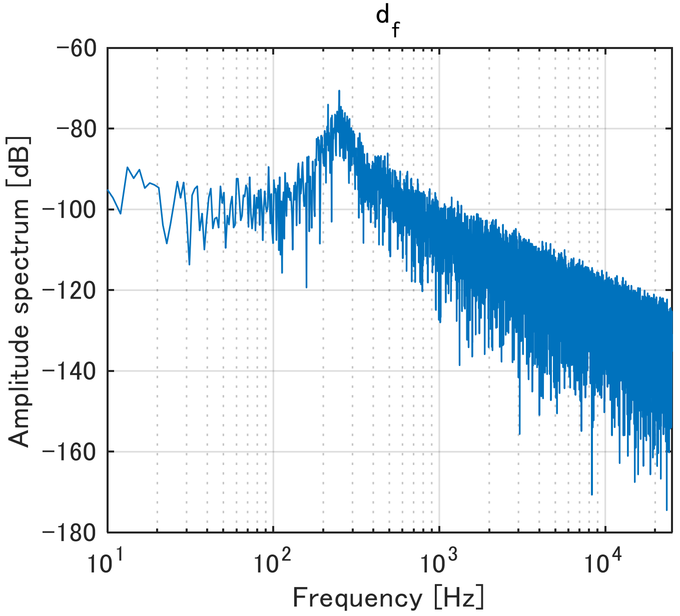


Figure 2: Amplitude spectrum of $d_p$

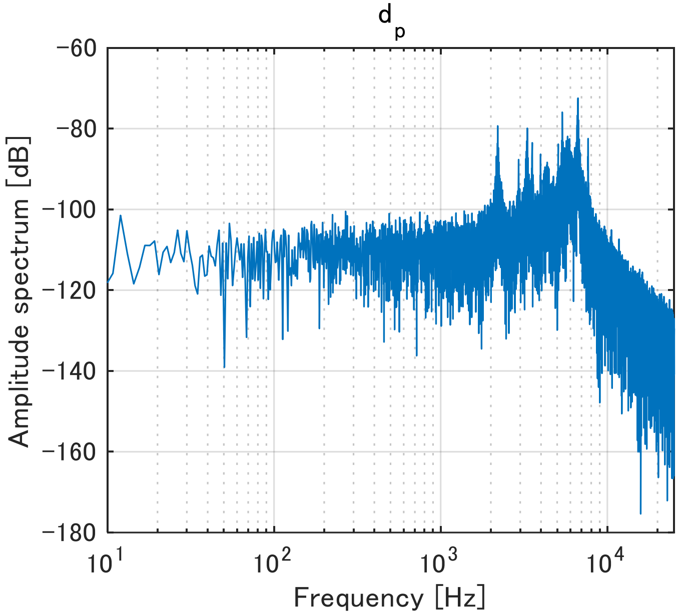
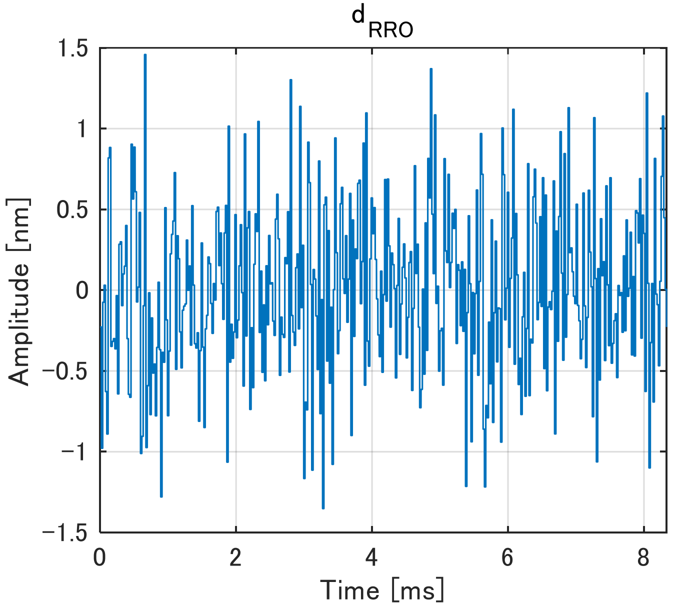
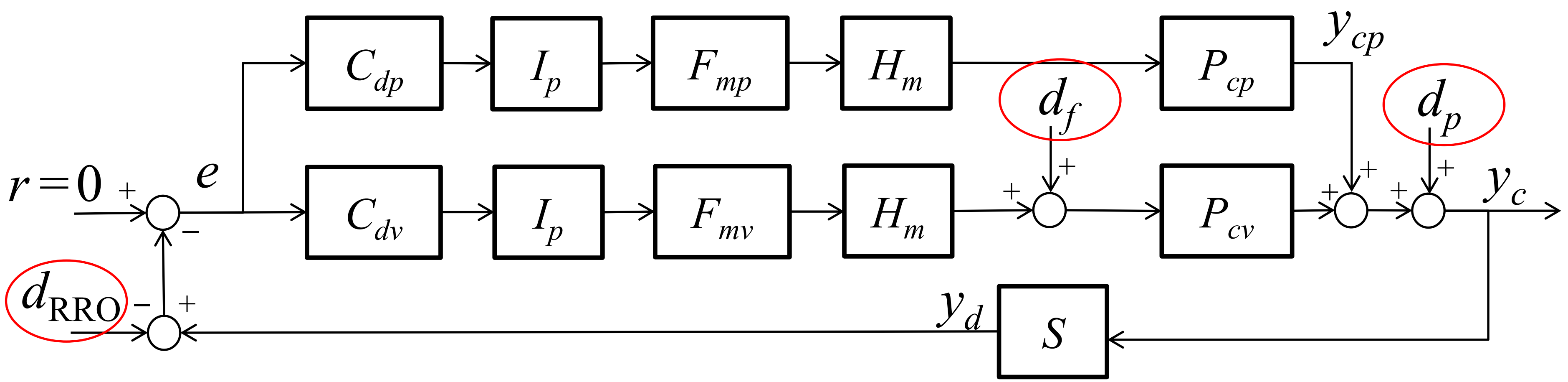


Figure 3: $d_{\mathrm{RRO}}$





# Results of “Simulation_trackfollow” with the example controller (Cont.)

Figure 4: $y_{cp}$

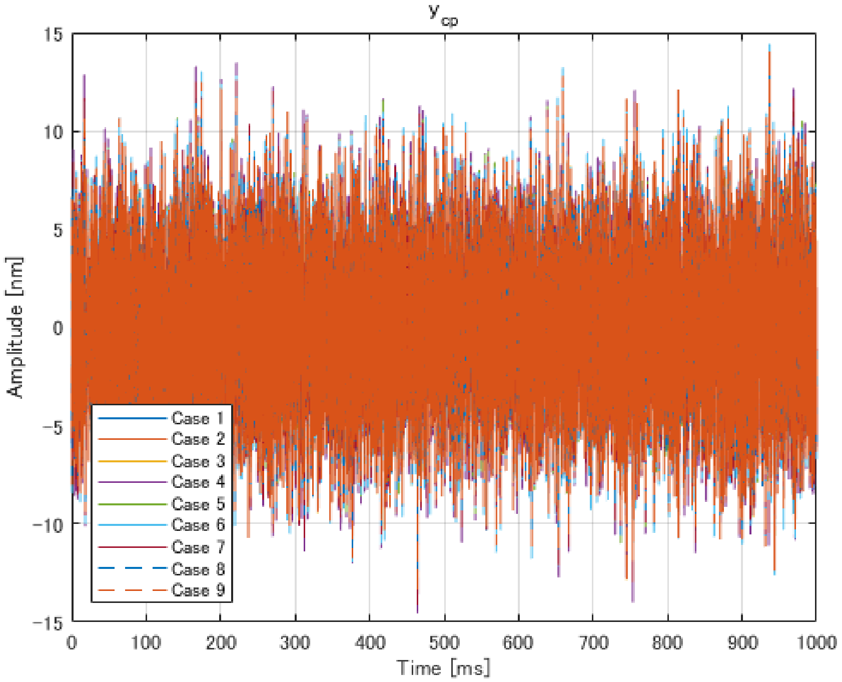
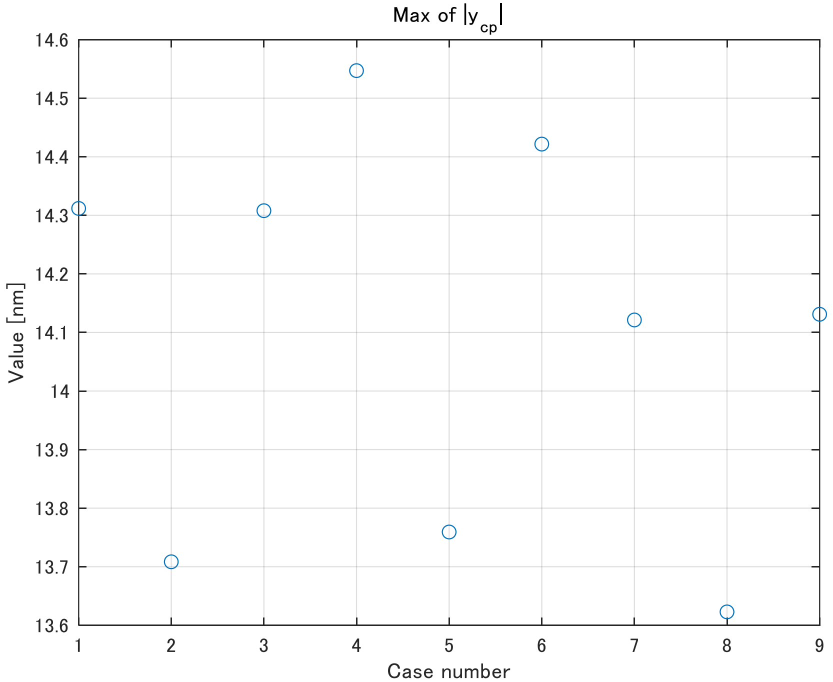
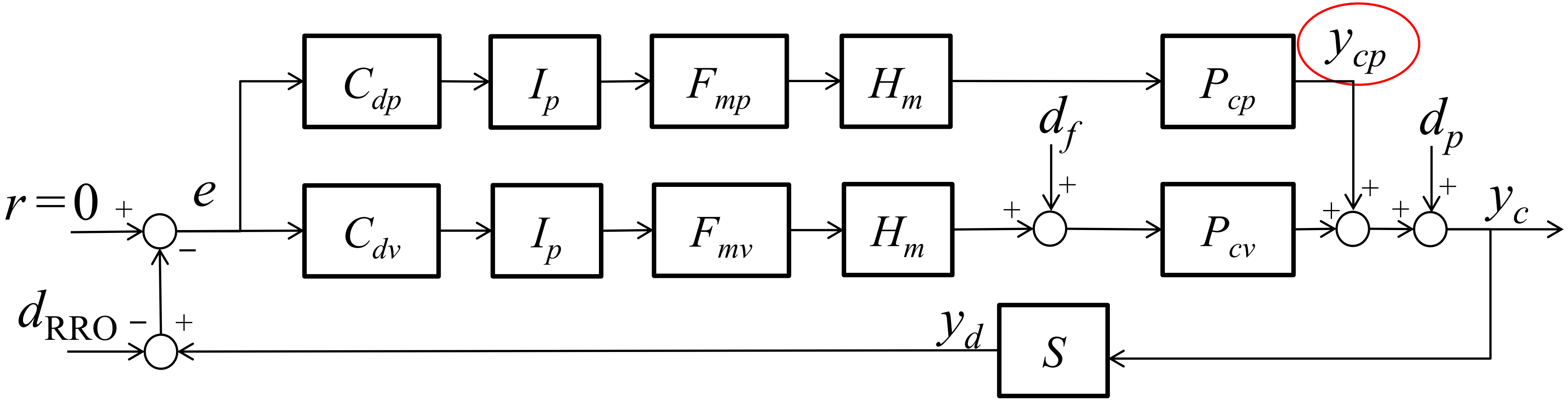


Figure 5: Max of $|y_{cp}|$





# Results of “Simulation_trackfollow” with the example controller (Cont.)

Figure 6: $y_c$

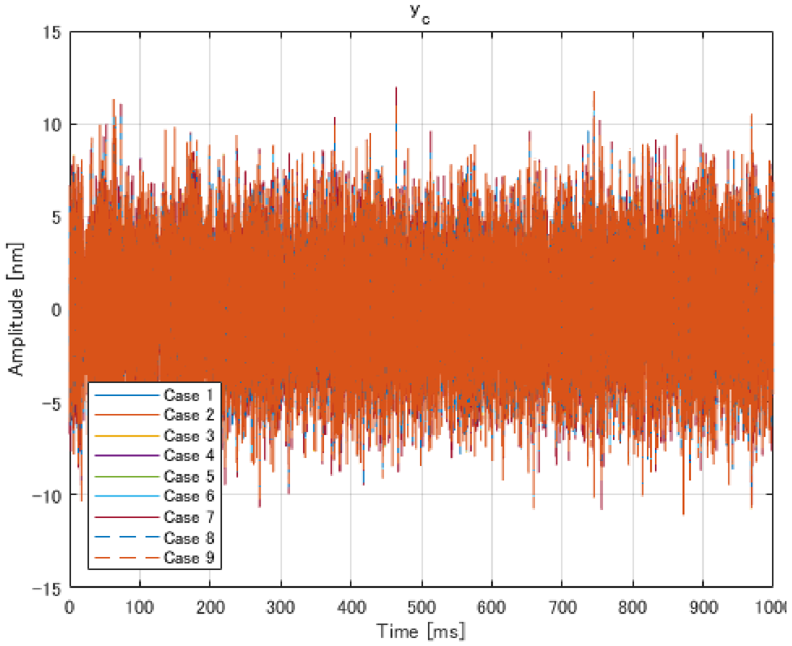


Figure 7: Amplitude spectrum of $y_c$

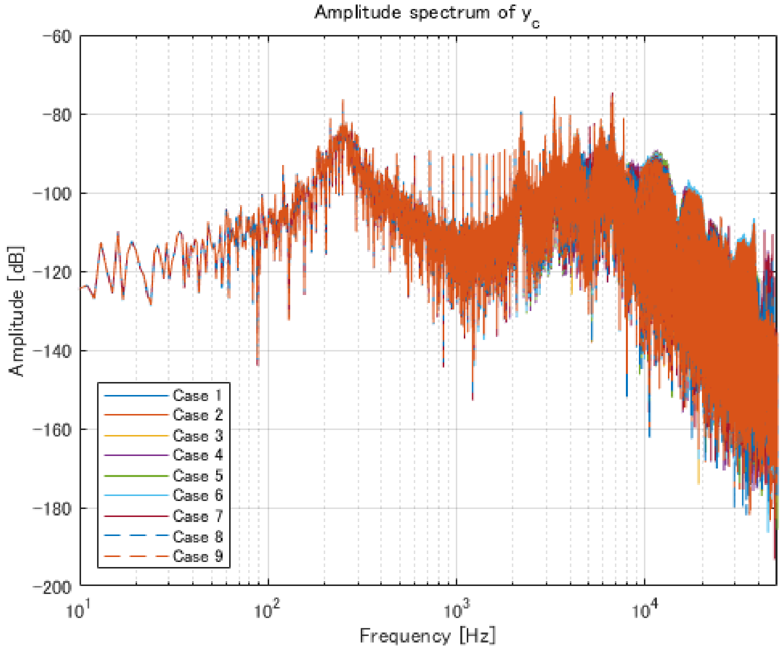
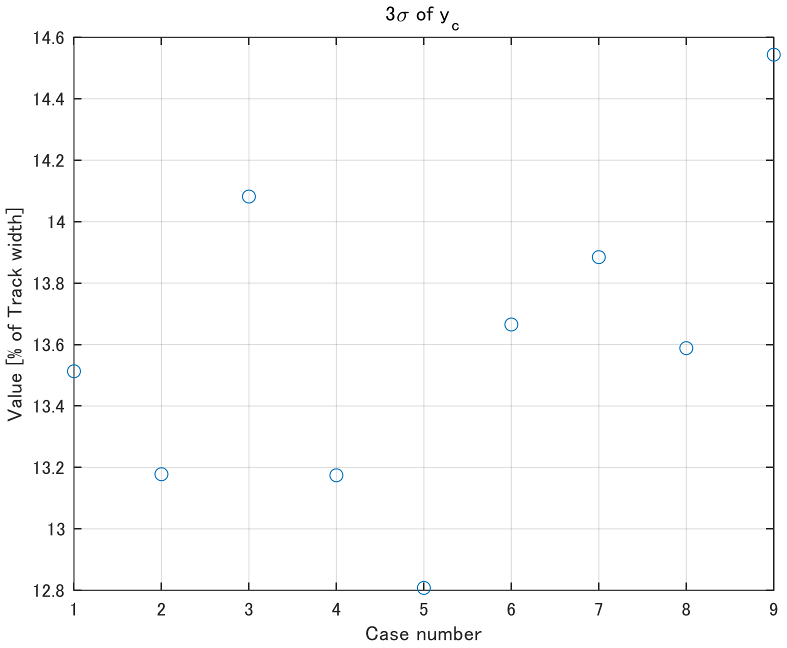
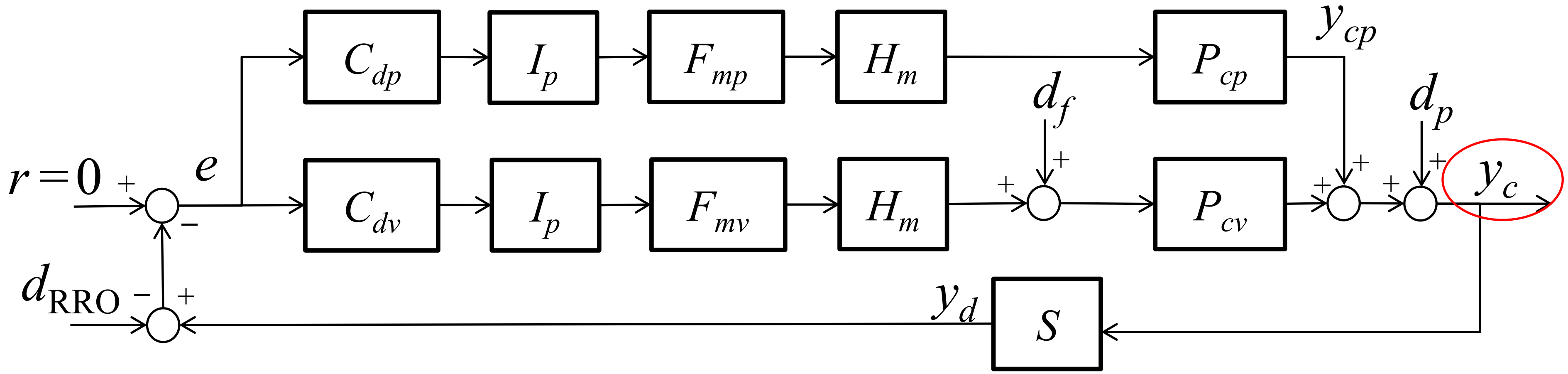


Figure 8: 3$\sigma$ values of $y_c$[% of Track width]





# Results of “Plot_control_system” with the example controller

Figure 9: $P_{cv}$

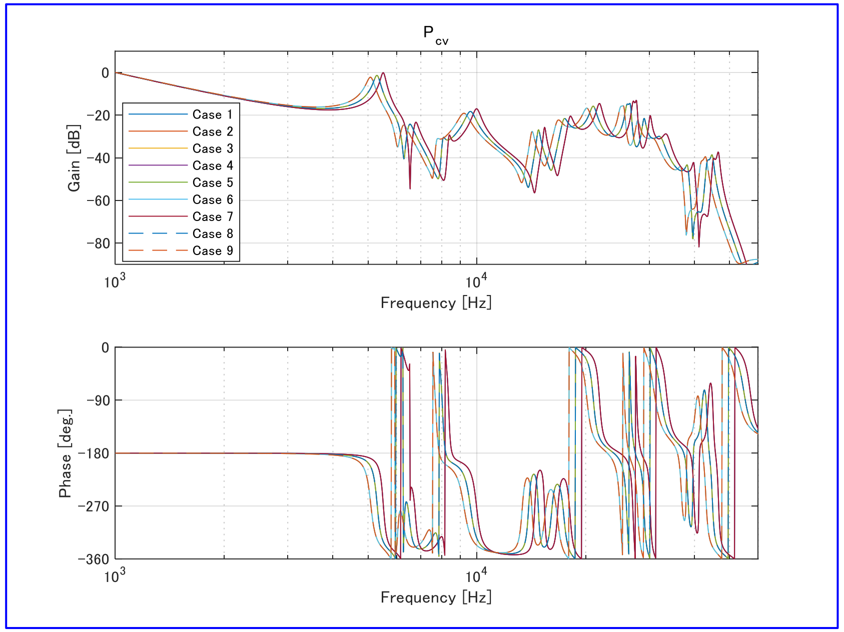
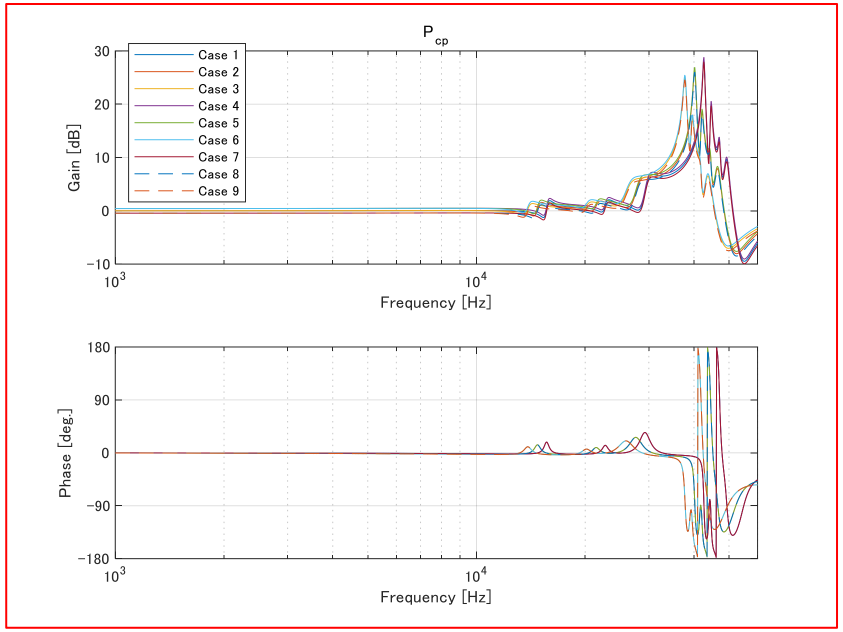
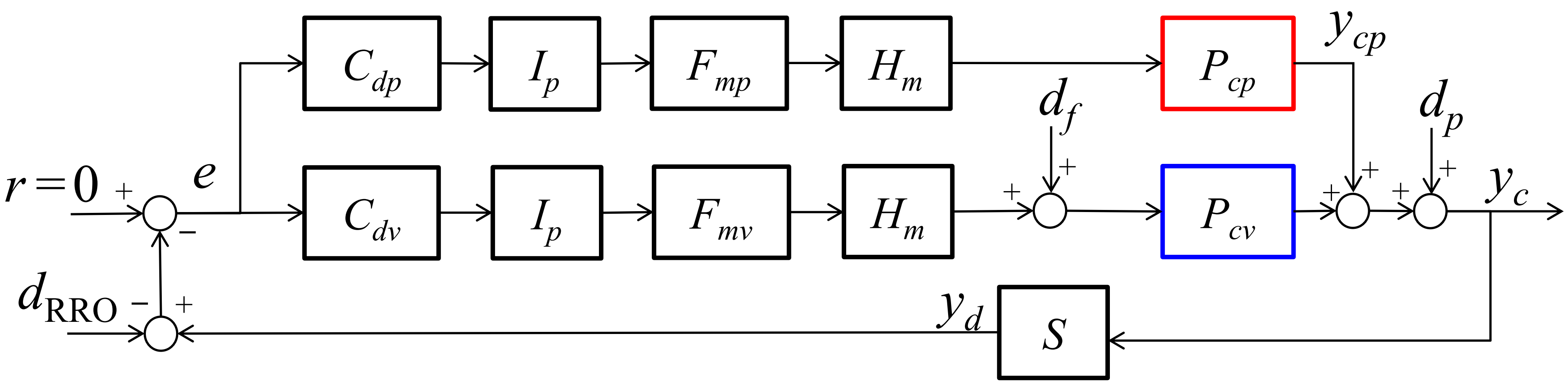


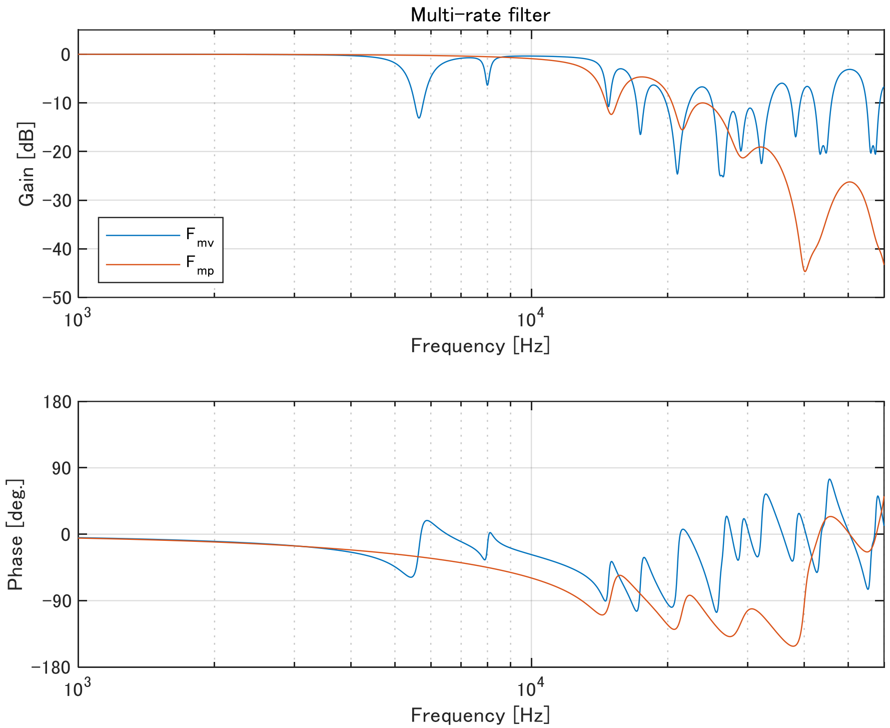
Figure 10: $P_{cp}$

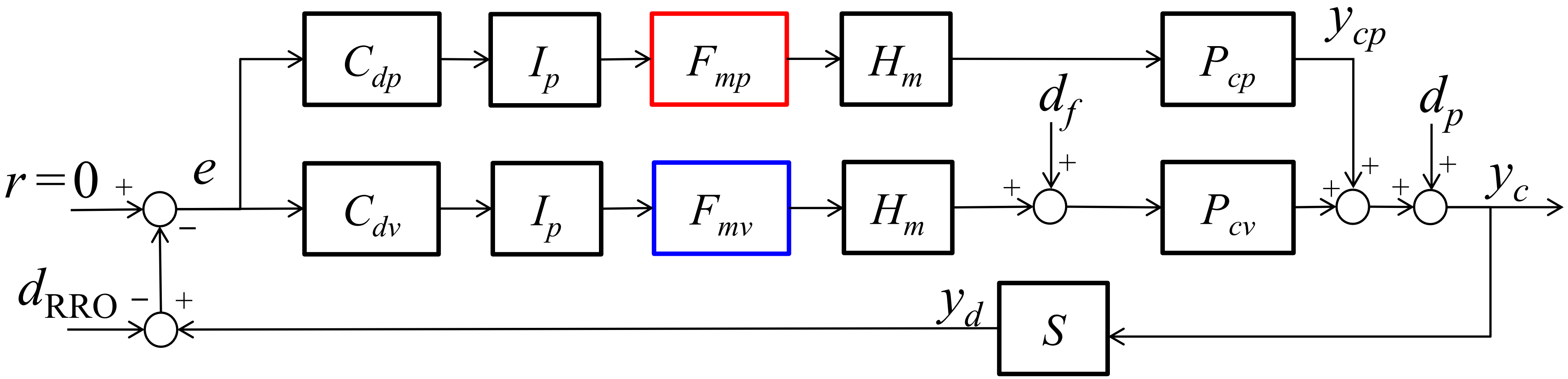




# Results of “Plot_control_system” with the example controller (Cont.)

Figure 11: Multi-rate filters





# Results of "Plot_control_system" with the example controller (Cont.)

Figure 12: Bode plot for $u_{dv} \longrightarrow y_d$

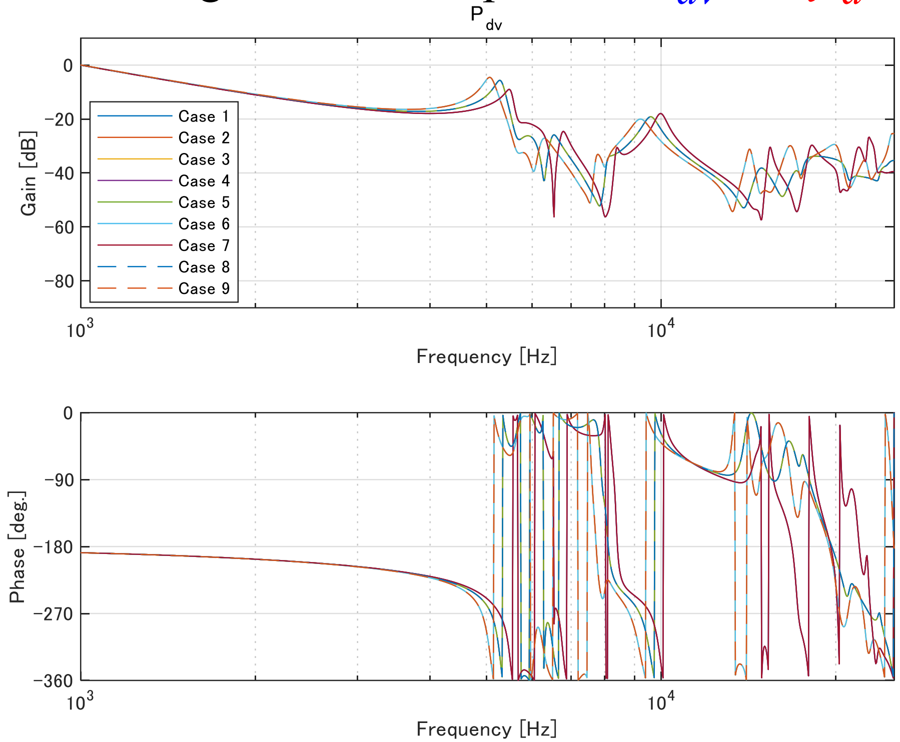
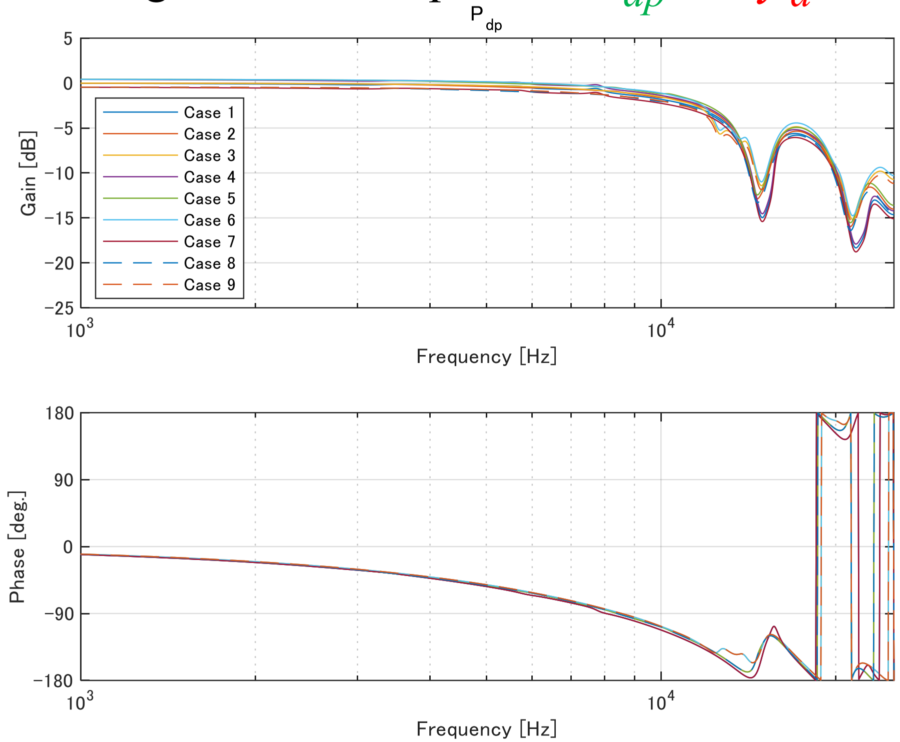
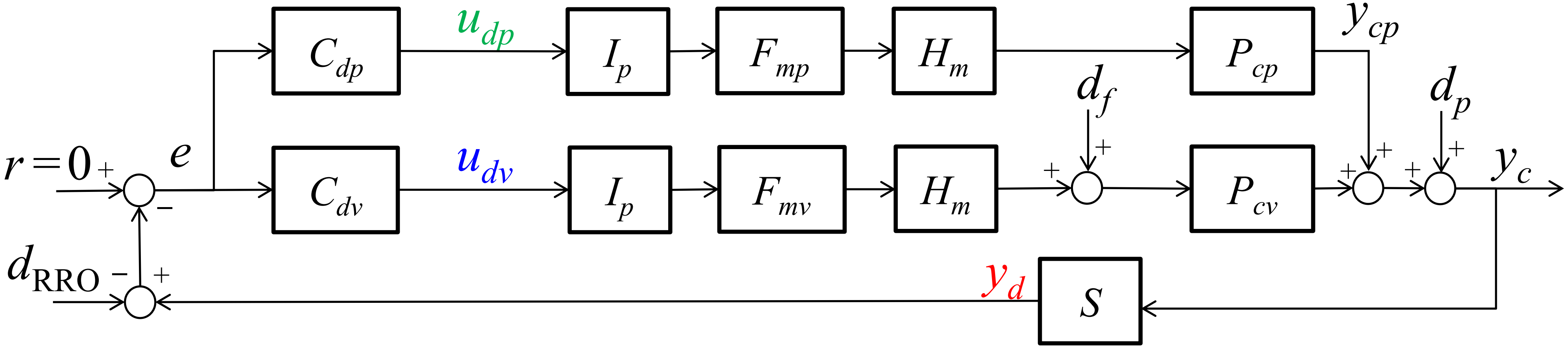


Figure 13: Bode plot for $u_{dp} \longrightarrow y_d$





# Results of “Plot_control_system” with the example controller (Cont.)

Figure 14 : Feedback controller for VCM

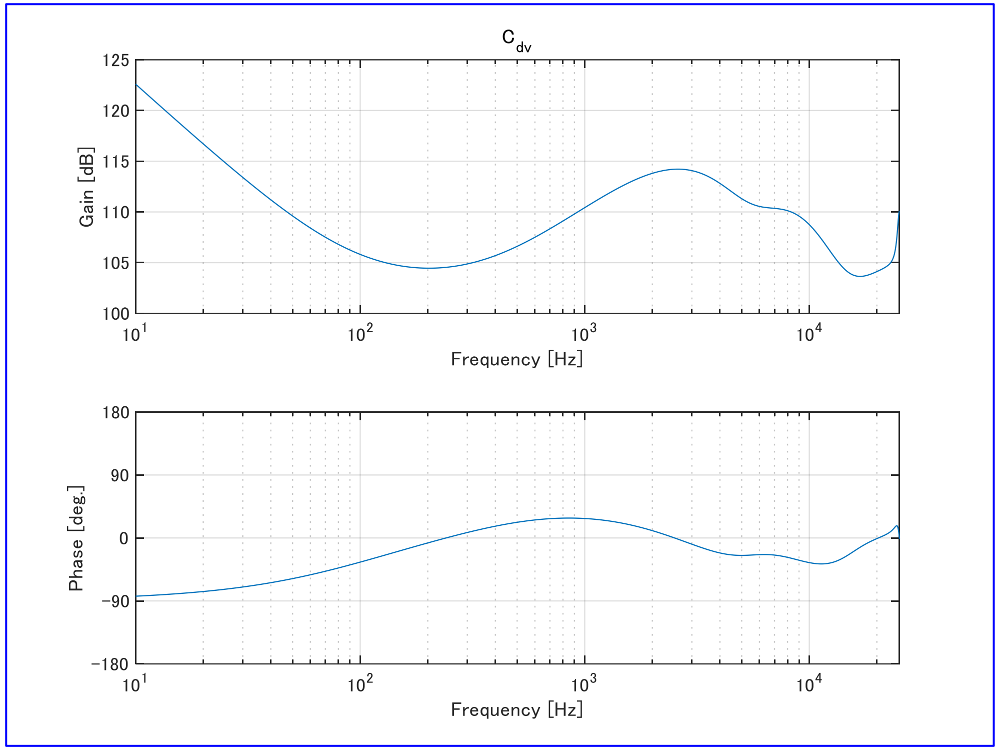
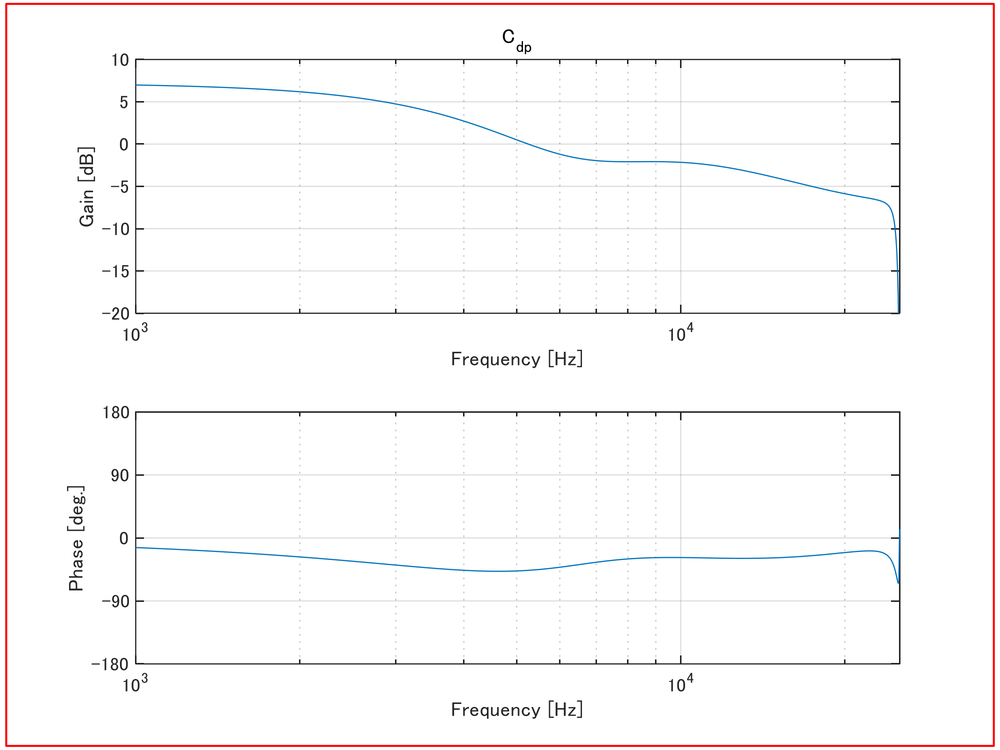
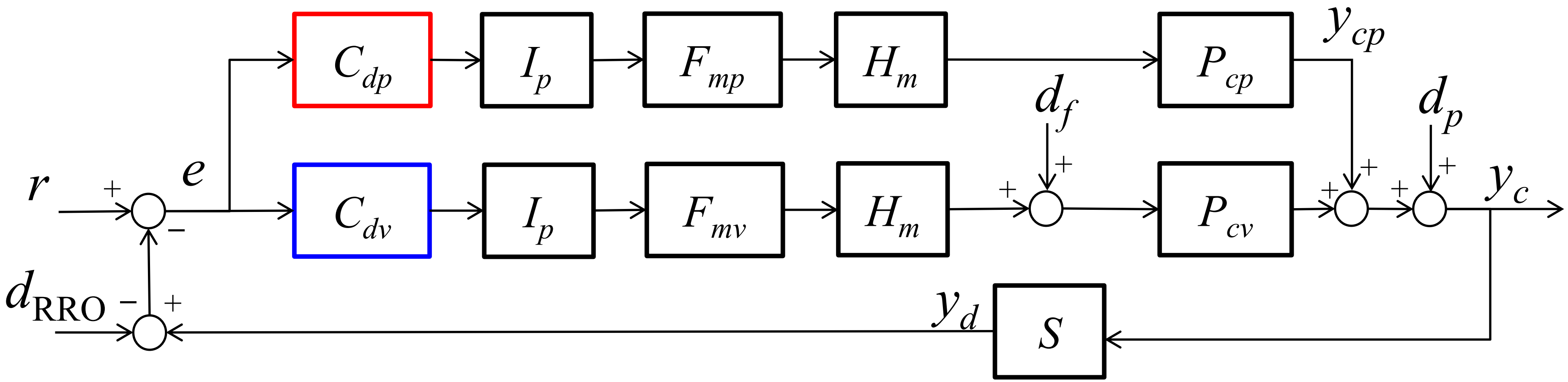


Figure 15 : Feedback controller for PZT actuator





# Results of “Plot_control_system” with the example controller (Cont.)

Figure 16: Bode plot for $e \longrightarrow y_d$

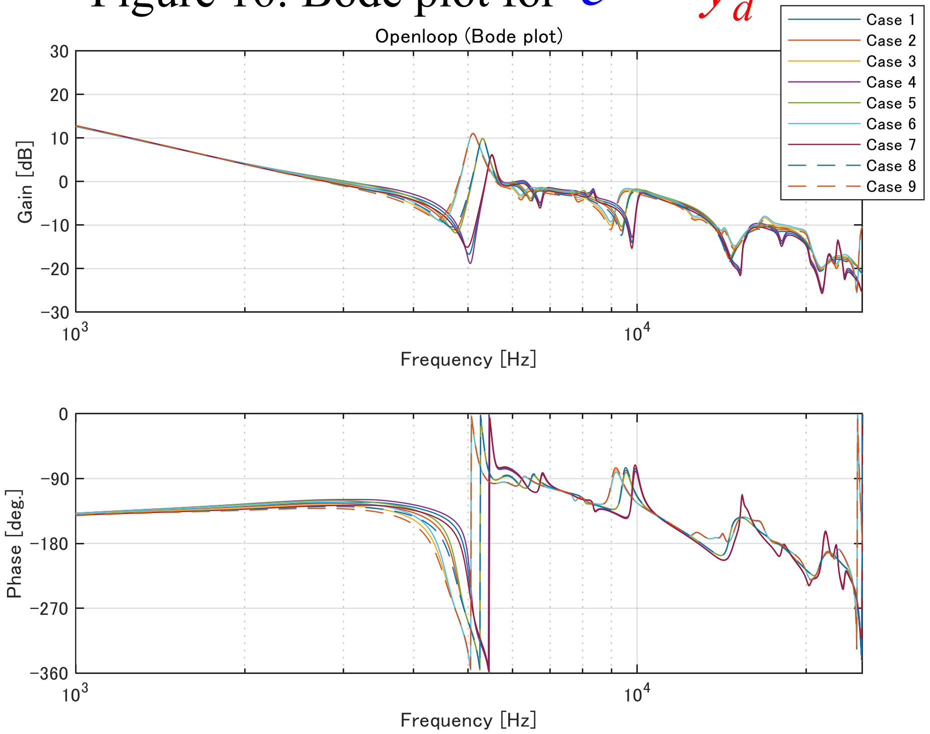
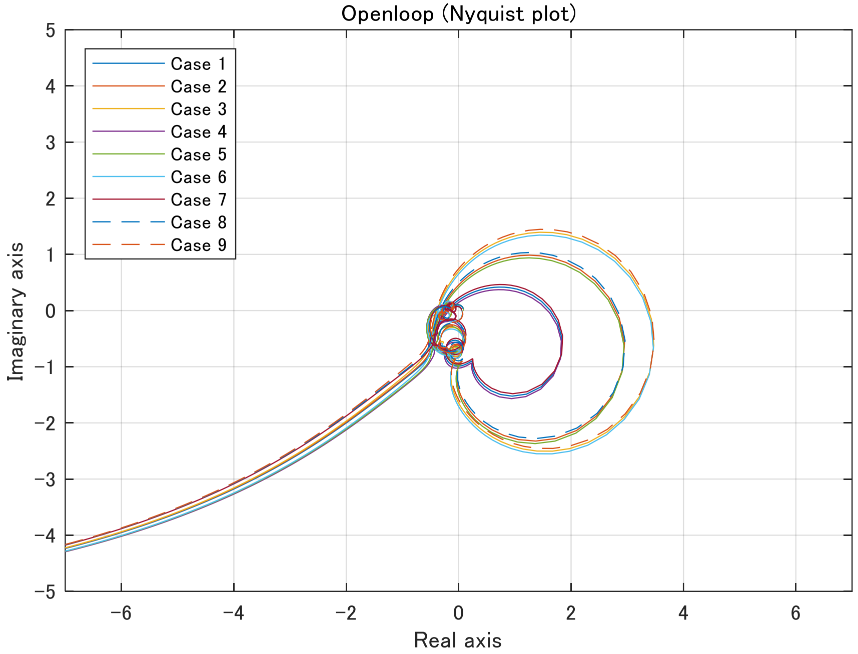
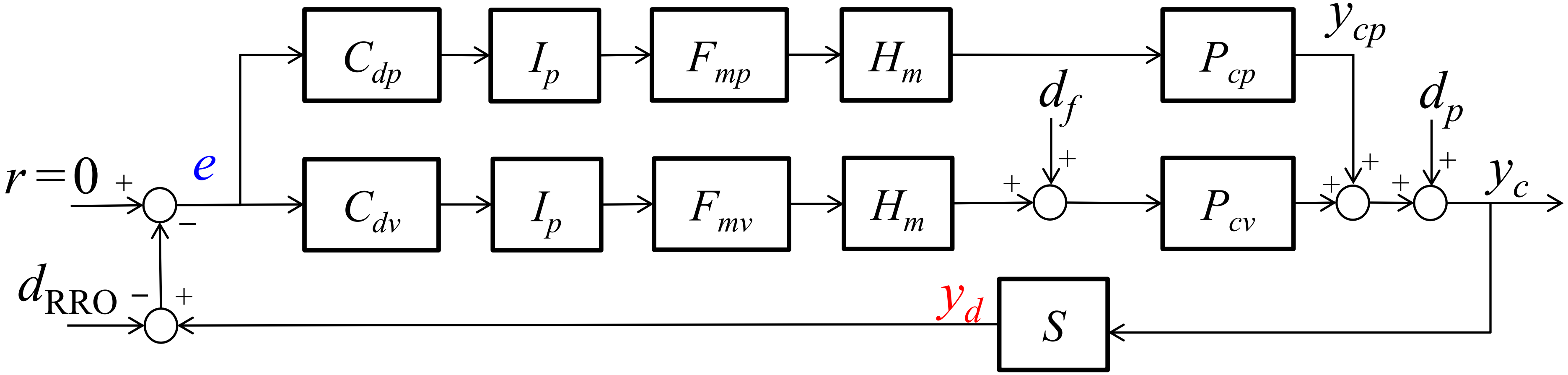


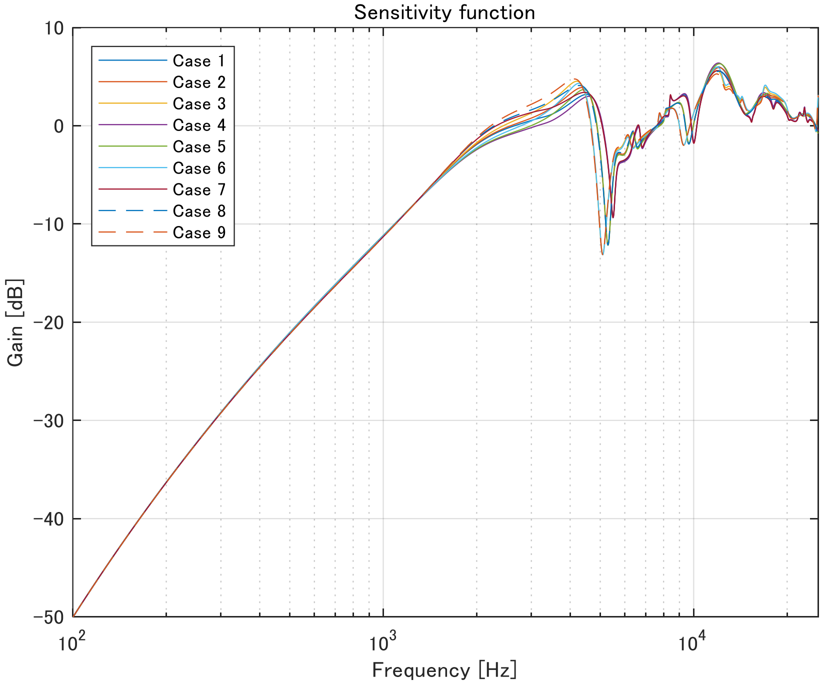
Figure 17: Nyquist plot for $e \longrightarrow y_d$

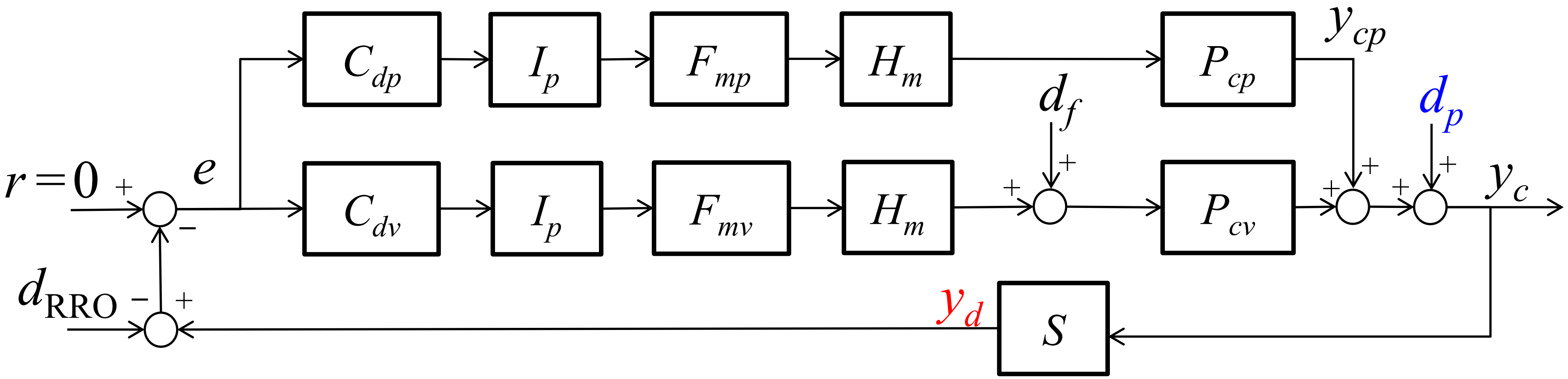




# Results of “Plot_control_system” with the example controller (Cont.)

Figure 18: Gain Bode plot for $d_p \to y_d$





- Control system in the benchmark problem
- How to use the benchmark problem
- Example
- MATLAB code

# Files in the benchmark problem

**Main-program code**

1. Simulation_trackfollow.m: This m-file executes the track-following simulation with the given controllers ($C_{dv}$, $C_{dp}$, $F_{mv}$, and $F_{mp}$) in “Data_Cd.mat” and “Data_Fm.mat”.
2. Plot_control_system.m: This m-file shows frequency responses of the control system with the given controllers ($C_{dv}$, $C_{dp}$, $F_{mv}$, and $F_{mp}$) in “Data_Cd.mat” and “Data_Fm.mat”.

**Sub-program code**

1. Plant.m: This m-file includes parameters of the control system and sets the dynamic system models of the controlled objects. This code is used “Simulation_trackfollow.m” and “Plot_control_system.m”.
2. Function_simulation.m: This m-file is a function file for the track-following simulation. The details are explained from P.33 to P.41.

**Data file**

1. Data_Cd.mat: This mat file includes feedback controllers ($C_{dv}$ : Sys_Cd_vcm, $C_{dp}$ : Sys_Cd_pzt).
2. Data_Fm.mat: This mat file includes multi-rate filters ($F_{dv}$ : Sys_FDd_vcm, $F_{dp}$ : Sys_Fd_pzt).
3. RROdata.txt: This text file includes the RRO data in time domain.

# Usage of “Function_simulation”

This function is used in “Simulation_trackfollow.m” (from 14th to 22nd lines).

Definition

```
function sim_result=Function_simulation(Sys_Pc_vcm,Sys_Pc_pzt,Sys_Cd_vcm,Sys_Fm_vcm,Sys_Cd_pzt,Sys_Fm_pzt,Ts,Mr_f)
```

OUTPUT

sim_result : Simulation results of track-following control with input system.

INPUT

Sys_Pc_vcm : Dynamic system model of $P_{cv}$ (VCM)
Sys_Pc_pzt : Dynamic system model of $P_{cp}$ (PZT actuator)
Sys_Cd_vcm : Dynamic system model of $C_{dv}$ (Feedback controller for VCM)
Sys_Fm_vcm : Dynamic system model of $F_{mv}$ (Multi-rate filter for VCM)
Sys_Cd_pzt : Dynamic system model of $C_{dp}$ (Feedback controller for PZT actuator)
Sys_Fm_pzt : Dynamic system model of $F_{mp}$ (Multi-rate filter for PZT actuator)
Ts : Sampling time of sampler
Mr_f : Multi-rate number

# Code description: Function_simulation.m

```
function sim_result=Function_simulation(Sys_Pc_vcm,Sys_Pc_pzt,Sys_Cd_vcm,Sys_Fm_vcm,Sys_Cd_pzt,Sys_Fm_pzt,Ts,Mr_f)
```

Define the function

```
s=tf('s');
```

Define Laplace operator

```
%% Simulation condition
Mr_p=20;                    % Multi-rate for continuous-time system
Tsc=Ts/Mr_p;                % Sampling time for continuous-time system
Tsim=1.1;                   % End of simulation time
```

```
%% Controlled object
Sys_Pcd_vcm=c2d(ssbal(ss(Sys_Pc_vcm)),Tsc);[A_Pcd_vcm,B_Pcd_vcm,C_Pcd_vcm,~]=ssdata(ssbal(Sys_Pcd_vcm));
Sys_Pcd_pzt=c2d(ssbal(ss(Sys_Pc_pzt)),Tsc);[A_Pcd_pzt,B_Pcd_pzt,C_Pcd_pzt,~]=ssdata(ssbal(Sys_Pcd_pzt));
```

Define the approximate continuous-time systems

```
%% Feedback Controller & Multi-rate filter
[A_Cd_vcm,B_Cd_vcm,C_Cd_vcm,D_Cd_vcm]=ssdata(ssbal(ss(Sys_Cd_vcm)));
[A_Fm_vcm,B_Fm_vcm,C_Fm_vcm,D_Fm_vcm]=ssdata(ssbal(ss(Sys_Fm_vcm)));
[A_Cd_pzt,B_Cd_pzt,C_Cd_pzt,D_Cd_pzt]=ssdata(ssbal(ss(Sys_Cd_pzt)));
[A_Fm_pzt,B_Fm_pzt,C_Fm_pzt,D_Fm_pzt]=ssdata(ssbal(ss(Sys_Fm_pzt)));
```

Define the state-space matrixes for feedback controllers and muti-rate filters

```
%% Disturbance signal
% FAN-Induced Disturbance
Sys_Dp1_c=0.6/(s^2+2*0.008*2200*2*pi*s+(2200*2*pi)^2);Sys_Dp1=c2d(ssbal(ss(Sys_Dp1_c)),Tsc);[A_Dp1,B_Dp1,C_Dp1,~]=ssdata(ssbal(Sys_Dp1));
Sys_Dp2_c=0.3/(s^2+2*0.005*2937*2*pi*s+(2937*2*pi)^2);Sys_Dp2=c2d(ssbal(ss(Sys_Dp2_c)),Tsc);[A_Dp2,B_Dp2,C_Dp2,~]=ssdata(ssbal(Sys_Dp2));
Sys_Dp3_c=1/(s^2+2*0.005*3300*2*pi*s+(3300*2*pi)^2);Sys_Dp3=c2d(ssbal(ss(Sys_Dp3_c)),Tsc);[A_Dp3,B_Dp3,C_Dp3,~]=ssdata(ssbal(Sys_Dp3));
Sys_Dp4_c=0.5/(s^2+2*0.005*3545*2*pi*s+(3545*2*pi)^2);Sys_Dp4=c2d(ssbal(ss(Sys_Dp4_c)),Tsc);[A_Dp4,B_Dp4,C_Dp4,~]=ssdata(ssbal(Sys_Dp4));
Sys_Dp5_c=0.3/(s^2+2*0.002*3980*2*pi*s+(3980*2*pi)^2);Sys_Dp5=c2d(ssbal(ss(Sys_Dp5_c)),Tsc);[A_Dp5,B_Dp5,C_Dp5,~]=ssdata(ssbal(Sys_Dp5));
Sys_Dp6_c=1/(s^2+2*0.01*4220*2*pi*s+(4220*2*pi)^2);Sys_Dp6=c2d(ssbal(ss(Sys_Dp6_c)),Tsc);[A_Dp6,B_Dp6,C_Dp6,~]=ssdata(ssbal(Sys_Dp6));
Sys_Dp7_c=1/(s^2+2*0.008*4380*2*pi*s+(4380*2*pi)^2);Sys_Dp7=c2d(ssbal(ss(Sys_Dp7_c)),Tsc);[A_Dp7,B_Dp7,C_Dp7,~]=ssdata(ssbal(Sys_Dp7));
```

Define fan-induced vibrations

# Code description: Function_simulation.m (Cont.)

```
Sys_Dp8_c=0.5/(s^2+2*0.002*5072*2*pi*s+(5072*2*pi)^2);Sys_Dp8=c2d(ssbal(ss(Sys_Dp8_c)),Tsc);[A_Dp8,B_Dp8,C_Dp8,~]=ssdata(ssbal(Sys_Dp8));
Sys_Dp9_c=2/(s^2+2*0.003*5370*2*pi*s+(5370*2*pi)^2);Sys_Dp9=c2d(ssbal(ss(Sys_Dp9_c)),Tsc);[A_Dp9,B_Dp9,C_Dp9,~]=ssdata(ssbal(Sys_Dp9));
Sys_Dp10_c=15/(s^2+2*0.04*5850*2*pi*s+(5850*2*pi)^2);Sys_Dp10=c2d(ssbal(ss(Sys_Dp10_c)),Tsc);[A_Dp10,B_Dp10,C_Dp10,~]=ssdata(ssbal(Sys_Dp10));
Sys_Dp11_c=10/(s^2+2*0.008*6660*2*pi*s+(6660*2*pi)^2);Sys_Dp11=c2d(ssbal(ss(Sys_Dp11_c)),Tsc);[A_Dp11,B_Dp11,C_Dp11,~]=ssdata(ssbal(Sys_Dp11));
Sys_Dp12_c=1.5/(s^2+2*0.003*7670*2*pi*s+(7670*2*pi)^2);Sys_Dp12=c2d(ssbal(ss(Sys_Dp12_c)),Tsc);[A_Dp12,B_Dp12,C_Dp12,~]=ssdata(ssbal(Sys_Dp12));
Sys_Dp13_c=2.5/(s^2+2*0.07*9200*2*pi*s+(9200*2*pi)^2);Sys_Dp13=c2d(ssbal(ss(Sys_Dp13_c)),Tsc);[A_Dp13,B_Dp13,C_Dp13,~]=ssdata(ssbal(Sys_Dp13));
```

Define fan-induced vibrations

```
% RRO
RRO_data=readmatrix("Data_RRO.txt")*0.5e-10;
num_sector=420;
```

Define RRO

```
% Rotational vibration
Sys_Df_c=3e-
10*(s+50*2*pi)/(s+3*2*pi)*(s^2+2*20*2000*2*pi*s+(2000*2*pi)^2)/(s^2+2*0.1*250*2*pi*s+(250*2*pi)^2);Sys_Df=c2d(ssbal(ss(Sys_Df_c)),Tsc);[A_Df,B_Df,C
_Df,~]=ssdata(ssbal(Sys_Df));
```

Define RV

```
% Random signal for disturbance
rng(1,'twister');u_random1 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(2,'twister');u_random2 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(3,'twister');u_random3 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(4,'twister');u_random4 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(5,'twister');u_random5 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(6,'twister');u_random6 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(7,'twister');u_random7 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(8,'twister');u_random8 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(9,'twister');u_random9 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(10,'twister');u_random10 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(11,'twister');u_random11 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(12,'twister');u_random12 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(13,'twister');u_random13 = (rand(round(Tsim/Tsc)+1,1)-0.5);
rng(14,'twister');u_random14 = (rand(round(Tsim/Tsc)+1,1)-0.5);
```

Define random signals for fan-induce vibe. and RV.

# Code description: Function_simulation.m (Cont.)

```
%% Simulation
% Initial value
x_Cd_vcm=zeros(length(A_Cd_vcm),1);x_Fm_vcm=zeros(length(A_Fm_vcm),1);
x_Cd_pzt=zeros(length(A_Cd_pzt),1);x_Fm_pzt=zeros(length(A_Fm_pzt),1);
dx_Df=zeros(length(A_Df),1);dy_Df=0;
ddf=0;
dx_Dp1=zeros(length(A_Dp1),1);dy_Dp1=0;
dx_Dp2=zeros(length(A_Dp2),1);dy_Dp2=0;
dx_Dp3=zeros(length(A_Dp3),1);dy_Dp3=0;
dx_Dp4=zeros(length(A_Dp4),1);dy_Dp4=0;
dx_Dp5=zeros(length(A_Dp5),1);dy_Dp5=0;
dx_Dp6=zeros(length(A_Dp6),1);dy_Dp6=0;
dx_Dp7=zeros(length(A_Dp7),1);dy_Dp7=0;
dx_Dp8=zeros(length(A_Dp8),1);dy_Dp8=0;
dx_Dp9=zeros(length(A_Dp9),1);dy_Dp9=0;
dx_Dp10=zeros(length(A_Dp10),1);dy_Dp10=0;
dx_Dp11=zeros(length(A_Dp11),1);dy_Dp11=0;
dx_Dp12=zeros(length(A_Dp12),1);dy_Dp12=0;
dx_Dp13=zeros(length(A_Dp13),1);dy_Dp13=0;
ddp=0;
dx_Pcd_vcm=zeros(length(A_Pcd_vcm),1);dy_Pcd_vcm=0;
dx_Pcd_pzt=zeros(length(A_Pcd_pzt),1);dy_Pcd_pzt=0;
dyc=0;
i_RRO=1;
```

Define initial values for simulation

```
% Number of inter-sampling
nn_s=Mr_p;
nn_m=Mr_p/Mr_f;
```

Define number of intersampling for multi-rate sampled-data control system.

# Code description: Function_simulation.m (Cont.)

```
% Simulation
for n=1:round(Tsim/Tsc)+1
```

Start of for-loop simulation

```
        t(n)=Tsc*(n-1);

        % Discrete-time system (Single-rate)
        if nn_s==Mr_p
```

Start of calculation for single-rate digital filter ($C_{dv}, C_{dp}$)

```
                nn_s=0;
                y(n)=dyc;
```

Set $y_d$

```
                % RRO
                dRRO(n)=RRO_data(i_RRO);
                if i_RRO == num_sector
                        i_RRO=1;
                else
                        i_RRO=i_RRO+1;
                end
```

Set $d_{\mathrm{RRO}}$

```
                e(n)= -y(n)+dRRO(n);
```

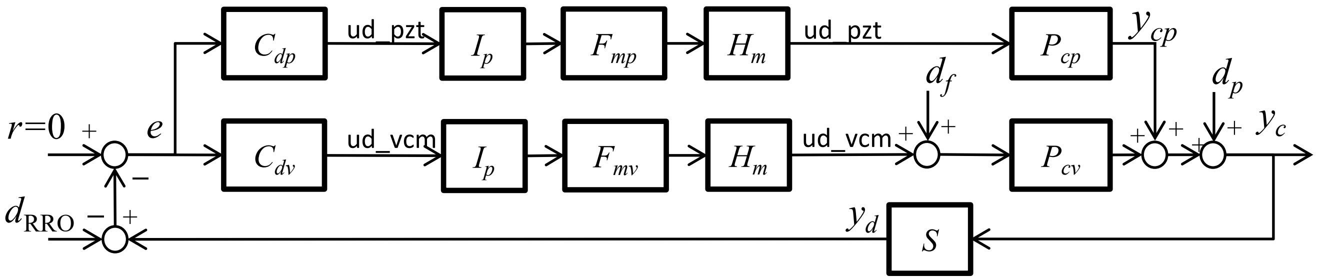
Set $e$

```
                % for VCM
                y_Cd_vcm=C_Cd_vcm*x_Cd_vcm+D_Cd_vcm*e(n);
                x_Cd_vcm=A_Cd_vcm*x_Cd_vcm+B_Cd_vcm*e(n);
                ud_vcm(n)=y_Cd_vcm;
```

Calculate $C_{dv}$

```
                % for PZT
                y_Cd_pzt=C_Cd_pzt*x_Cd_pzt+D_Cd_pzt*e(n);
                x_Cd_pzt=A_Cd_pzt*x_Cd_pzt+B_Cd_pzt*e(n);
                ud_pzt(n)=y_Cd_pzt;
```

Calculate $C_{dp}$



If you will employ controllers which are not LTI systems, you can rewrite this part by yourself.

# Code description: Function_simulation.m (Cont.)

```
else
        dRRO(n)=dRRO(n-1);
        e(n)=e(n-1);
        ud_vcm(n)=ud_vcm(n-1);
        ud_pzt(n)=ud_pzt(n-1);
```

Set $d_{\mathrm{RRO}}$, $e$, ud_vcm, and ud_pzt during inter sampling.

```
end
```

End of calculation for single-rate digital filter ($C_{dv}$, $C_{dp}$)

```
% Discrete-time system (Multi-rate)
if nn_m==Mr_p/Mr_f
        nn_m=0;
```

Start of calculation for multi-rate digital filter ($F_{mv}$, $F_{mp}$)

```
        % for VCM
        y_Fm_vcm=C_Fm_vcm*x_Fm_vcm+D_Fm_vcm*ud_vcm(n);
        x_Fm_vcm=A_Fm_vcm*x_Fm_vcm+B_Fm_vcm*ud_vcm(n);
        uc_vcm(n)=y_Fm_vcm;
```

Calculate $F_{mv}$

```
        % for PZT
        y_Fm_pzt=C_Fm_pzt*x_Fm_pzt+D_Fm_pzt*ud_pzt(n);
        x_Fm_pzt=A_Fm_pzt*x_Fm_pzt+B_Fm_pzt*ud_pzt(n);
        uc_pzt(n)=y_Fm_pzt;
```

Calculate $F_{mp}$

```
else
        uc_vcm(n)=uc_vcm(n-1);
        uc_pzt(n)=uc_pzt(n-1);
```

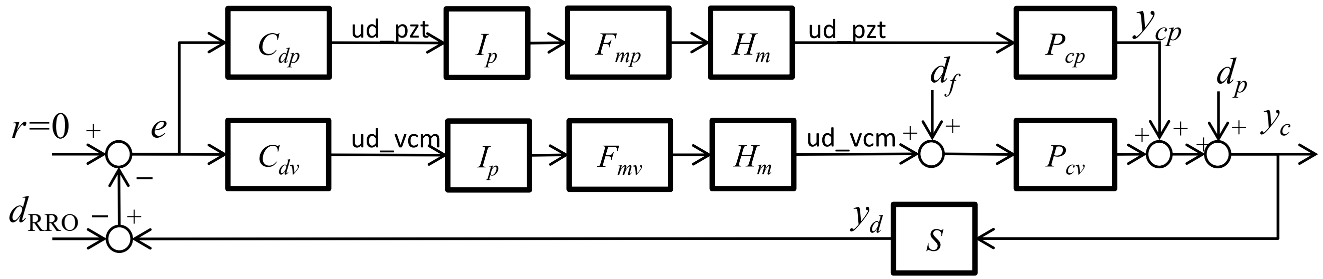
Set uc_vcm, and uc_pzt during inter sampling.

```
end
```

End of calculation for multi-rate digital filter ($F_{mv}$, $F_{mp}$)

```
% RV disturbance
x_Df=dx_Df;
y_Df=dy_Df;
dx_Df=A_Df*x_Df+B_Df*u_random1(n);
dy_Df=C_Df*dx_Df;
df(n)=ddf;
ddf = dy_Df;
```

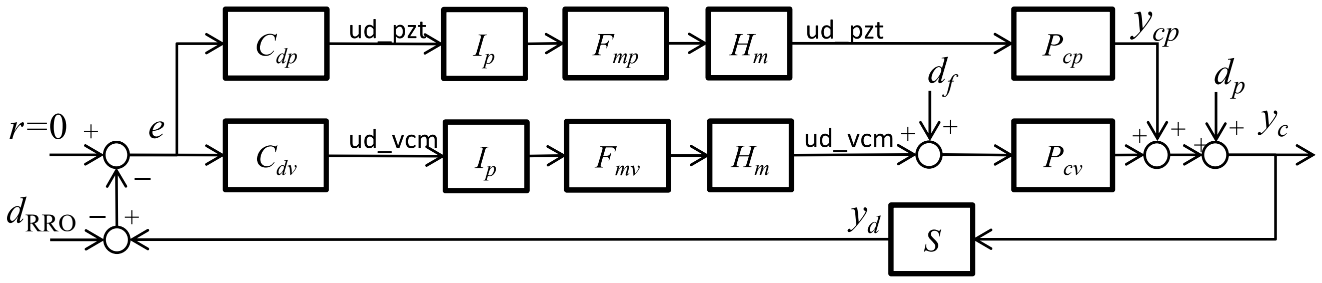
Calculate $d_f$ (RV)



# Code description: Function_simulation.m (Cont.)

```
    % Continuous-time system (VCM)
    x_Pcd_vcm=dx_Pcd_vcm;
    y_Pcd_vcm(n)=dy_Pcd_vcm;
    dx_Pcd_vcm=A_Pcd_vcm*x_Pcd_vcm+B_Pcd_vcm*(uc_vcm(n)+df(n));
    dy_Pcd_vcm=C_Pcd_vcm*dx_Pcd_vcm;
```

Calculate $P_{cv}$



```
    % Continuous-time system (PZT)
    x_Pcd_pzt=dx_Pcd_pzt;
    y_Pcd_pzt(n)=dy_Pcd_pzt;
    dx_Pcd_pzt=A_Pcd_pzt*x_Pcd_pzt+B_Pcd_pzt*uc_pzt(n);
    dy_Pcd_pzt=C_Pcd_pzt*dx_Pcd_pzt;
```

Calculate $P_{cp}$

```
    % FAN-induced disturbance
    x_Dp1=dx_Dp1;y_Dp1=dy_Dp1;dx_Dp1=A_Dp1*x_Dp1+B_Dp1*u_random2(n);dy_Dp1=C_Dp1*dx_Dp1;
    x_Dp2=dx_Dp2;y_Dp2=dy_Dp2;dx_Dp2=A_Dp2*x_Dp2+B_Dp2*u_random3(n);dy_Dp2=C_Dp2*dx_Dp2;
    x_Dp3=dx_Dp3;y_Dp3=dy_Dp3;dx_Dp3=A_Dp3*x_Dp3+B_Dp3*u_random4(n);dy_Dp3=C_Dp3*dx_Dp3;
    x_Dp4=dx_Dp4;y_Dp4=dy_Dp4;dx_Dp4=A_Dp4*x_Dp4+B_Dp4*u_random5(n);dy_Dp4=C_Dp4*dx_Dp4;
    x_Dp5=dx_Dp5;y_Dp5=dy_Dp5;dx_Dp5=A_Dp5*x_Dp5+B_Dp5*u_random6(n);dy_Dp5=C_Dp5*dx_Dp5;
    x_Dp6=dx_Dp6;y_Dp6=dy_Dp6;dx_Dp6=A_Dp6*x_Dp6+B_Dp6*u_random7(n);dy_Dp6=C_Dp6*dx_Dp6;
    x_Dp7=dx_Dp7;y_Dp7=dy_Dp7;dx_Dp7=A_Dp7*x_Dp7+B_Dp7*u_random8(n);dy_Dp7=C_Dp7*dx_Dp7;
    x_Dp8=dx_Dp8;y_Dp8=dy_Dp8;dx_Dp8=A_Dp8*x_Dp8+B_Dp8*u_random9(n);dy_Dp8=C_Dp8*dx_Dp8;
    x_Dp9=dx_Dp9;y_Dp9=dy_Dp9;dx_Dp9=A_Dp9*x_Dp9+B_Dp9*u_random10(n);dy_Dp9=C_Dp9*dx_Dp9;
    x_Dp10=dx_Dp10;y_Dp10=dy_Dp10;dx_Dp10=A_Dp10*x_Dp10+B_Dp10*u_random11(n);dy_Dp10=C_Dp10*dx_Dp10;
    x_Dp11=dx_Dp11;y_Dp11=dy_Dp11;dx_Dp11=A_Dp11*x_Dp11+B_Dp11*u_random12(n);dy_Dp11=C_Dp11*dx_Dp11;
    x_Dp12=dx_Dp12;y_Dp12=dy_Dp12;dx_Dp12=A_Dp12*x_Dp12+B_Dp12*u_random13(n);dy_Dp12=C_Dp12*dx_Dp12;
    x_Dp13=dx_Dp13;y_Dp13=dy_Dp13;dx_Dp13=A_Dp13*x_Dp13+B_Dp13*u_random14(n);dy_Dp13=C_Dp13*dx_Dp13;
    dp(n)=ddp;
    ddp = dy_Dp1+dy_Dp2+dy_Dp3+dy_Dp4+dy_Dp5+dy_Dp6+dy_Dp7+dy_Dp8+dy_Dp9+dy_Dp10+dy_Dp11+dy_Dp12+dy_Dp13;
```

Calculate $d_p$ (fan-induced vibe.)

```
    % Magnetic-head position
    yc(n)=dyc;dyc=dy_Pcd_vcm+dy_Pcd_pzt+ddp;
    nn_s=nn_s+1;nn_m=nn_m+1;
end
```

Set $y_c$

End of for-loop simulation

# Code description: Function_simulation.m (Cont.)

```
% Remove transient response
Nt=find(t>=0.1,1);
sim_result.time=t(Nt:Nt+Mr_p*num_sector*120-1)-t(Nt);
sim_result.uc_vcm=uc_vcm(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.uc_pzt=uc_pzt(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.yc=yc(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.yc_pzt=y_Pcd_pzt(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.yc=yc(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.dp=dp(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.df=df(Nt:Nt+Mr_p*num_sector*120-1);
sim_result.dRRO=dRRO(Nt:Nt+Mr_p*num_sector*120-1);
```

Remove transient responses from simulation results

```
% DFT
sim_result.freq=[0:length(sim_result.time)-1]/sim_result.time(end);
sim_result.Fr_yc=fft(sim_result.yc);
sim_result.Fr_yc_pzt=fft(sim_result.yc_pzt);
sim_result.Fr_dp=fft(sim_result.dp);
sim_result.Fr_df=fft(sim_result.df);
```

Calculate DFT of simulation results in steady state responses

1. Introduction

2. Magnetic-head positioning control system in HDDs

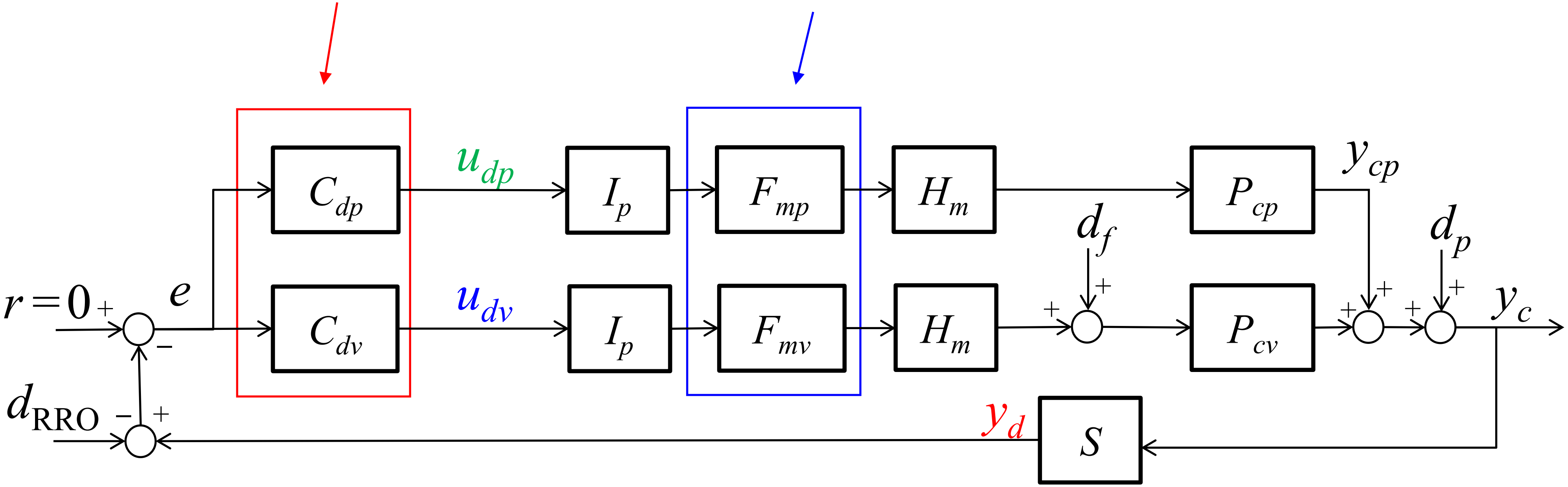
3. Benchmark problem

4. Tips

If you will design single-rate controllers only, you can employ the multi-rate filters “$F_{mv}$” and “$F_{mp}$” in downloaded “Data_Fm.mat”. In this case, the controlled objects are the transfer functions from $u_{dv}$ to $y_d$ (VCM) and from $u_{dp}$ to $y_d$ (PZT actuator). These frequency responses are shown in P.28.

Design these filters only.

Employ the multi-rate filters in the example.



If you have little experience about the magnetic-head positioning control system, you can try the old version of the HDD benchmark problem which is SISO control system.

- Website: [http://www2.iee.or.jp/~dmec/HDD_bench_e.html](http://www2.iee.or.jp/~dmec/HDD_bench_e.html)

- Academic papers which show the design results above HDD benchmark problem

  1. Q. Zheng and M. Tomizuka, "A Disturbance Observer Approach to Detecting and Rejecting Narrow-Band Disturbances in Hard Disk Drives," in Proceedings of The 10th IEEE International Workshop on Advanced Motion Control, 2008, pp. 254-259.

  2. T. Atsumi and W. C. Messner, "Modified Bode Plots for Robust Performance in SISO Systems with Structured and Unstructured Uncertainties," The IEEE Transactions on Control Systems Technology, vol. 20, no. 2, pp. 356-368, (2012)

If you have little experience about a multi-rate or sampled-data control system, following papers may indicate good tips for you.

1. T. Hara and M. Tomizuka, “Performance Enhancement of Multi-rate Controller for Hard Disk Drives,” The IEEE Transactions on Magnetics, vol. 35, no. 2, pp. 898-903, (1999)

2. M. Hirata, M. Takiguchi and K. Nonami, “Track-following control of hard disk drives using multi-rate sampled-data $H_\infty$ control,” in Proceeding of The 42nd IEEE International Conference on Decision and Control 2003, pp. 3414-3419.

3. T. Hara, “Openloop Gain Criteria above the Nyquist Frequency for HDD Head Positioning Control,” in Proceedings of The Asia-Pacific Magnetic Recording Conference, 2006, pp. 1-2

4. J. Ishikawa, “A Study on Multirate Sampled-Data Control for Hard Disk Drives,” in Proceedings of The 10th IEEE International Workshop on Advanced Motion Control, 2008, pp. 266-271.

5. T. Atsumi, A. Okuyama, and S. Nakagawa, “Vibration Control Above the Nyquist Frequency in Hard Disk Drives,” The IEEE Transactions on Industrial Electronics, vol. 55, no. 10, pp. 3751-3757, (2008)

6. T. Atsumi and W. C. Messner, “Mixed Sensitivity Problem in Sampled-Data Positioning Control Systems”, Journal of System Design and Dynamics, vol. 5, no. 6, pp. 1294-1309, (2011)

If you have little experience about the DISO control system design, the decouple controller design may help you. It is a popular technique in the HDD servo field to approximate a DISO control design problem by two SISO control design problem.

- Academic papers about the decouple controller design method.

1. K. Mori, T. Munemoto, H. Otsuki, Y. Yamaguchi, and K. Akagi, “A DUAL-STAGE MAGNETIC DISK DRIVE ACTUATOR USING A PIEZOELECTRIC DEVICE FOR A HIGH TRACK DENSITY,” The IEEE Transactions on Magnetics, vol. 27, no. 6, pp. 5298-5300, (1991)

2. S. Yabui, T. Atsumi, and T. Inoue, “Coupling Controller Design for MISO System of Head Positioning Control Systems in HDDs,” The IEEE Transactions on Magnetics, vol. 56, no. 5, Art no. 8100909, (2020)

# END

Benchmark problem for magnetic-head positioning control in HDD
User manual

Investigating R&D Committee on “Precision Servo Technology for High Value-Added Mechatronic System”
The Institute of Electrical Engineers of Japan