Introduction to Modern Controls

Modeling of Dynamic Systems

Why modeling?

Modeling of physical systems:

- a vital component of modern engineering

Why modeling?

Modeling of physical systems:

- a vital component of modern engineering
- often consists of complex coupled differential equations

Why modeling?

Modeling of physical systems:

- a vital component of modern engineering
- often consists of complex coupled differential equations
- only when we have good understanding of a system can we optimally control it:

Why modeling?

Modeling of physical systems:

- a vital component of modern engineering
- often consists of complex coupled differential equations
- only when we have good understanding of a system can we optimally control it:
  - can simulate and predict actual system response, and

Why modeling?

Modeling of physical systems:

- a vital component of modern engineering
- often consists of complex coupled differential equations
- only when we have good understanding of a system can we optimally control it:
  - can simulate and predict actual system response, and
  - design model-based controllers

Two general approaches of modeling

- based on physics:

Two general approaches of modeling

- based on physics:
  - using fundamental engineering principles such as Newton's laws, energy conservation, etc

Two general approaches of modeling

- based on physics:
  - using fundamental engineering principles such as Newton's laws, energy conservation, etc
- based on measurement data:

Two general approaches of modeling

- based on physics:
  - using fundamental engineering principles such as Newton's laws, energy conservation, etc
- based on measurement data:
  - using input-output response of the system

Two general approaches of modeling

- based on physics:
  - using fundamental engineering principles such as Newton's laws, energy conservation, etc
- based on measurement data:
  - using input-output response of the system
  - a field itself known as system identification

The Impact of Control Technology - 2nd Edition

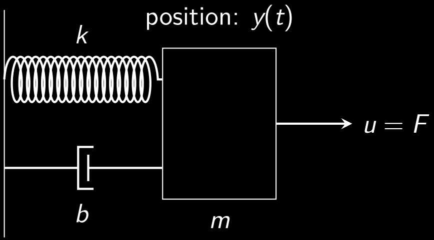
Part 1: Success Stories for Control

Download Compilation of Success Stories (application/pdf)	Download
Auto-tuners for PID Controllers (application/pdf)	Download
Control in Mobile Phones (application/pdf)	Download
Nonlinear Multivariable Flight Control (application/pdf)	Download
Advanced Control of Pharmaceutical Crystallization (application/pdf)	Download
Autopilot for Small Unmanned Aerial Vehicles (application/pdf)	Download
Controlling Energy Capture from Wind (application/pdf)	Download
Mobile-Robot-Enabled Smart Warehouses (application/pdf)	Download
Trip Optimizer for Railroads (application/pdf)	Download
Control in Stroke Rehabilitation (application/pdf)	Download
Advanced Control for the Cement Industry (application/pdf)	Download
Dynamics and Control for Deep-Sea Marine Risers (application/pdf)	Download
Active Safety Control for Automobiles (application/pdf)	Download
Advanced Energy Solutions for Power Plants (application/pdf)	Download
Optimal Ship-Unloading Solutions (application/pdf)	Download
Automotive Engine-Based Traction Control (application/pdf)	Download
Automated Manual Transmissions (application/pdf)	Download
Ethylene Plantwide Control and Optimization (application/pdf)	Download
Digital Printing Control: Print Shop in a Box (application/pdf)	Download
Verification of Control System Software (application/pdf)	Download
Coordinated Ramp Metering for Freeways (application/pdf)	Download
Advanced Zinc Coating Control in Galvanizing Lines (application/pdf)	Download
Digital Fly-by-Wire Technology (application/pdf)	Download
H-infinity Control for European Telecommunication Satellites (application/pdf)	Download
Road Grade Estimation for Advanced Driver Assistance Systems (application/pdf)	Download
Controller Performance Monitoring (application/pdf)	Download
Control for Formula One! (application/pdf)	Download
Advanced Control Design for Automotive Powertrains (application/pdf)	Download
Control of NASA's Space Launch System (application/pdf)	Download
Robust Adaptive Control for the Joint Direct Attack Munition (application/pdf)	Download
Improved Audio Reproduction with Control Theory (application/pdf)	Download

IEEE Control Systems Society™

- often the tools are combined in practice

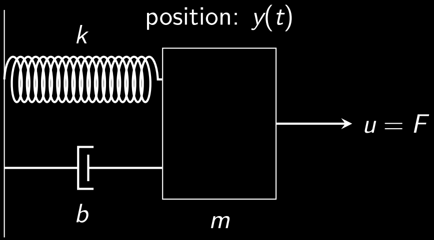
Example: Mass spring damper



- Newton's second law gives

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t), \; y(0) = y_0, \; \dot{y}(0) = \dot{y}_0$$

Example: Mass spring damper

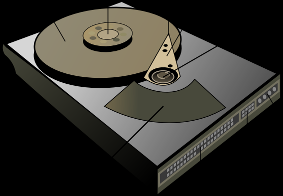


- Newton's second law gives

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t), \;\; y(0) = y_0, \;\; \dot{y}(0) = \dot{y}_0$$

- modeled as a second-order ODE with input $u(t)$ and output $y(t)$

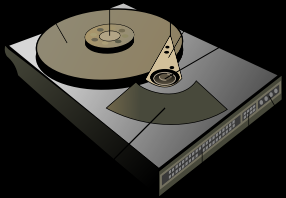
Example: HDD



- Newton's second law for rotation

$$\underbrace{\sum_i \tau_i}_{\text{net torque}} = \underbrace{J}_{\text{moment of inertia}} \underbrace{\alpha}_{\text{angular acceleration}}$$

Example: HDD



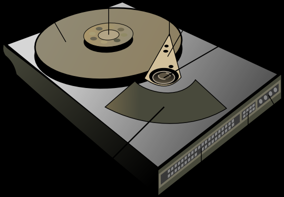
- Newton's second law for rotation

$$\underbrace{\sum_i \tau_i}_{\text{net torque}} = \underbrace{J}_{\text{moment of inertia}} \underbrace{\alpha}_{\text{angular acceleration}}$$

- letting θ :=output and τ :=input yields

$$\ddot{\theta} = \alpha = \frac{1}{J}\tau$$

Example: HDD

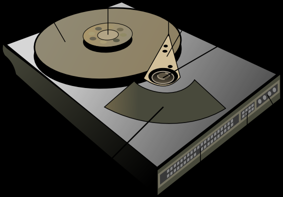


$$\ddot{\theta} = \alpha = \frac{1}{J}\tau \Leftrightarrow \Theta(s) = \frac{1}{Js^2}\mathrm{T}(s)$$

- with damping:

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = \kappa\tau \Leftrightarrow \Theta(s) = \frac{\kappa}{s^2 + 2\zeta\omega_n s + \omega_n^2}\mathrm{T}(s)$$

Example: HDD



$$\ddot{\theta} = \alpha = \frac{1}{J}\tau \Leftrightarrow \Theta(s) = \frac{1}{Js^2}\mathrm{T}(s)$$

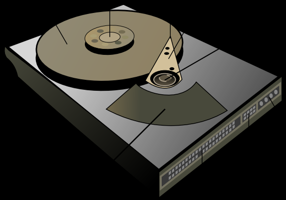
- with damping:

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = \kappa\tau \Leftrightarrow \Theta(s) = \frac{\kappa}{s^2 + 2\zeta\omega_n s + \omega_n^2}\mathrm{T}(s)$$

- with multiple modes:

$$\ddot{\theta}_i + 2\zeta_i\omega_i\dot{\theta}_i + \omega_i^2\theta_i = \kappa_i\tau \Leftrightarrow \Theta_i(s) = \frac{\kappa_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}\mathrm{T}(s)$$

Example: HDD

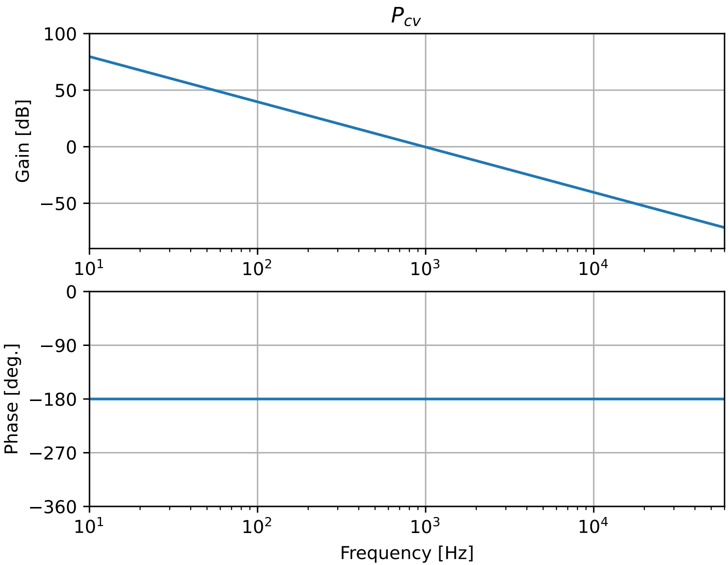


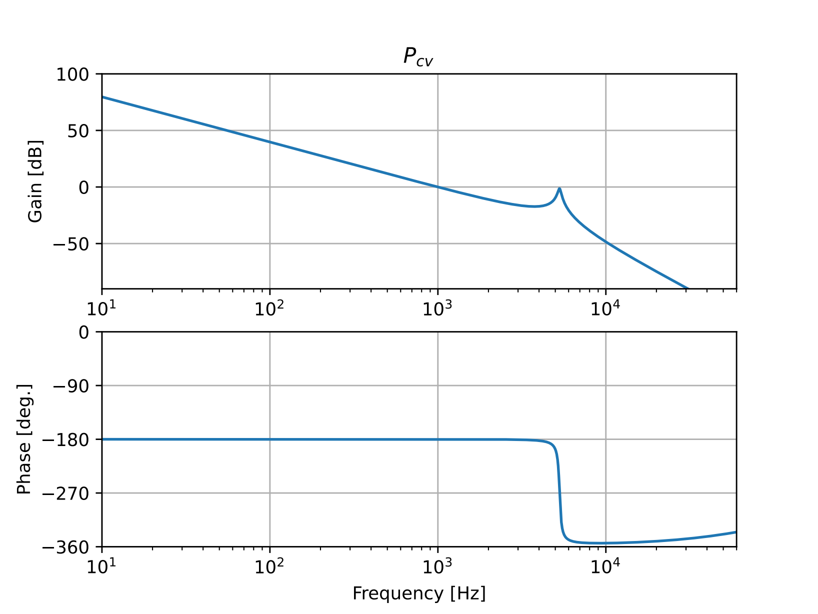
$$\ddot{\theta}_i + 2\zeta_i\omega_i\dot{\theta}_i + \omega_i^2\theta_i = \kappa_i\tau \Leftrightarrow \Theta_i(s) = \frac{\kappa_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}\mathrm{T}(s)$$

- final model:

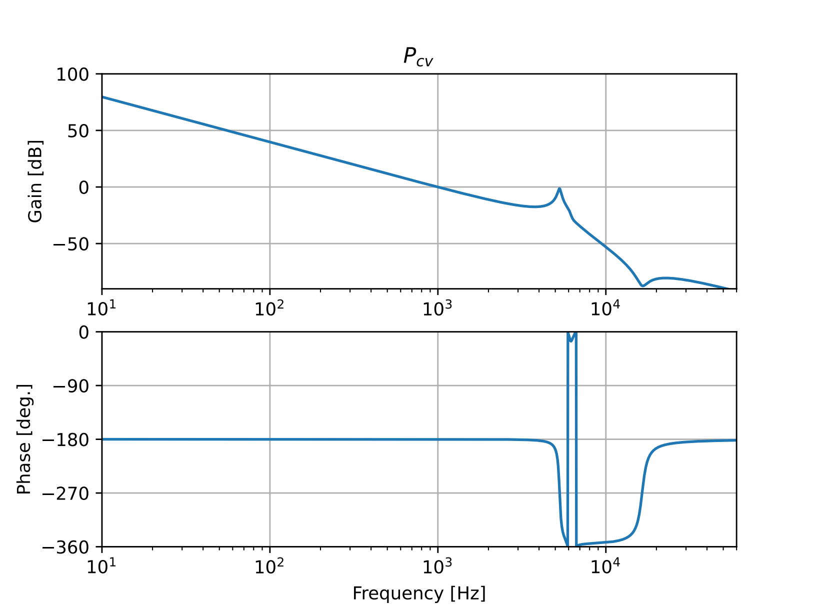
$$\Theta(s) = \sum_{i=1}^{n} \frac{\kappa_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}\mathrm{T}(s)$$

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
import control as ct
num_sector = 420  # Number of sector
num_rpm = 7200  # Number of RPM
Kp_vcm = 3.7976e+07  # VCM gain
omega_vcm = np.array([0, 5300, 6100, 6500, 8050, 9600, 14800, 17400,
                     21000, 26000, 26600, 29000, 32200, 38300, 43300,
                     ↪ 44800]) * 2 * np.pi
kappa_vcm = np.array([1, -1.0, +0.1, -0.1, 0.04, -0.7, -
                     0.2, -1.0, +3.0, -3.2, 2.1, -1.5, +2.0, -0.2,
                     ↪ +0.3, -0.5])
zeta_vcm = np.array([0, 0.02, 0.04, 0.02, 0.01, 0.03, 0.01,
                    0.02, 0.02, 0.012, 0.007, 0.01, 0.03, 0.01, 0.01,
                    ↪ 0.01])
Sys_Pc_vcm_c1 = ct.TransferFunction([], [1])  # Create an empty
↪ transfer function
for i in range(len(omega_vcm)):
    Sys_Pc_vcm_c1 = Sys_Pc_vcm_c1 + ct.TransferFunction(np.array(
        [0, 0, kappa_vcm[i]]) * Kp_vcm, np.array([1, 2 * zeta_vcm[i] *
        ↪ omega_vcm[i], (omega_vcm[i]) ** 2]))
```

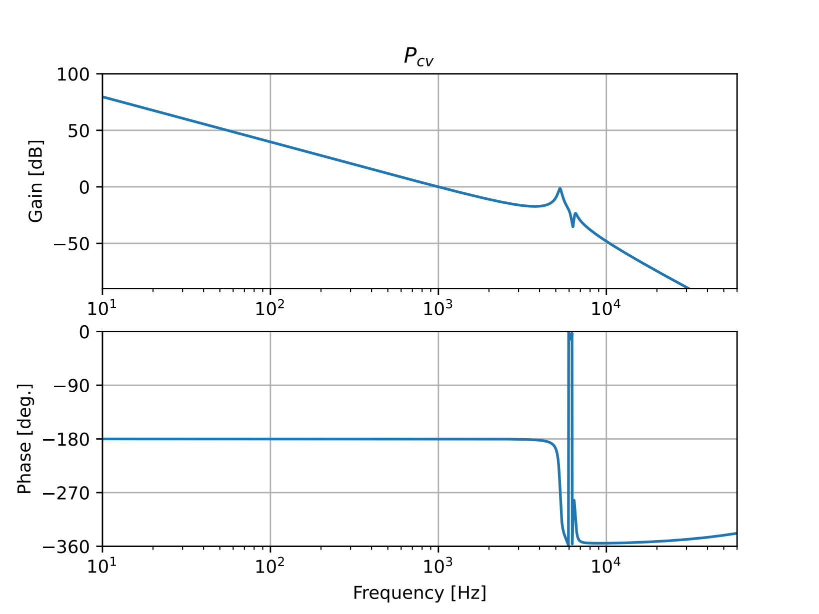

P_{cv}
Gain [dB]
100
50
0
−50
Phase [deg.]
0
−90
−180
−270
−360
10^1
10^2
10^3
10^4
Frequency [Hz]

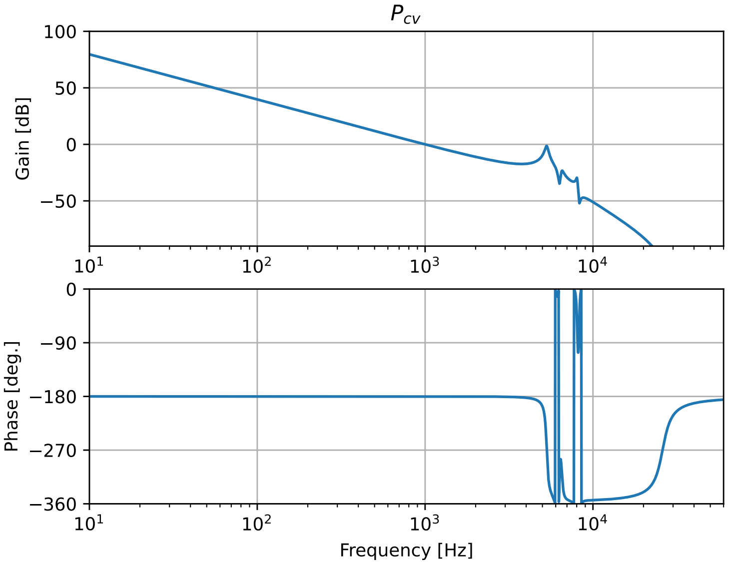


P_{cv}
Gain [dB]
100
50
0
−50
10^1
10^2
10^3
10^4
Phase [deg.]
0
−90
−180
−270
−360
10^1
10^2
10^3
10^4
Frequency [Hz]

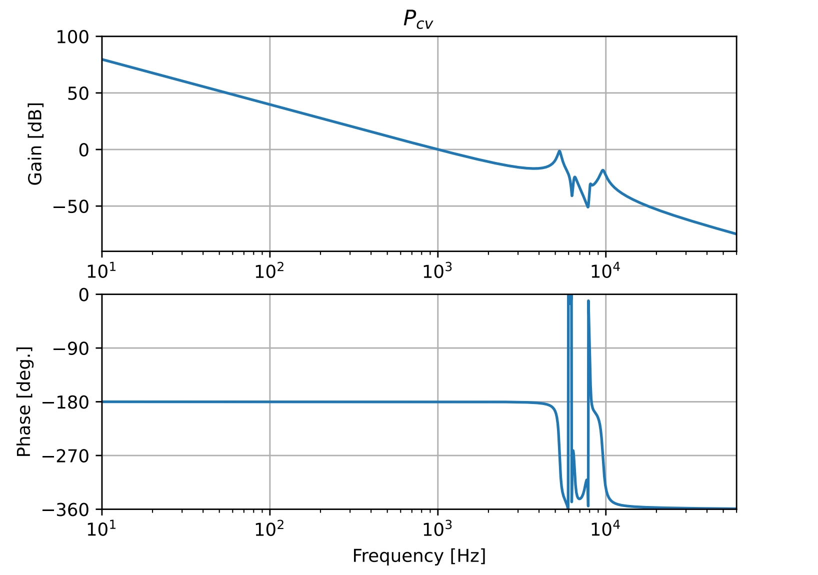


P_{cv}
100
50
0
−50
Gain [dB]
10^1
10^2
10^3
10^4
0
−90
−180
−270
−360
Phase [deg.]
10^1
10^2
10^3
10^4
Frequency [Hz]

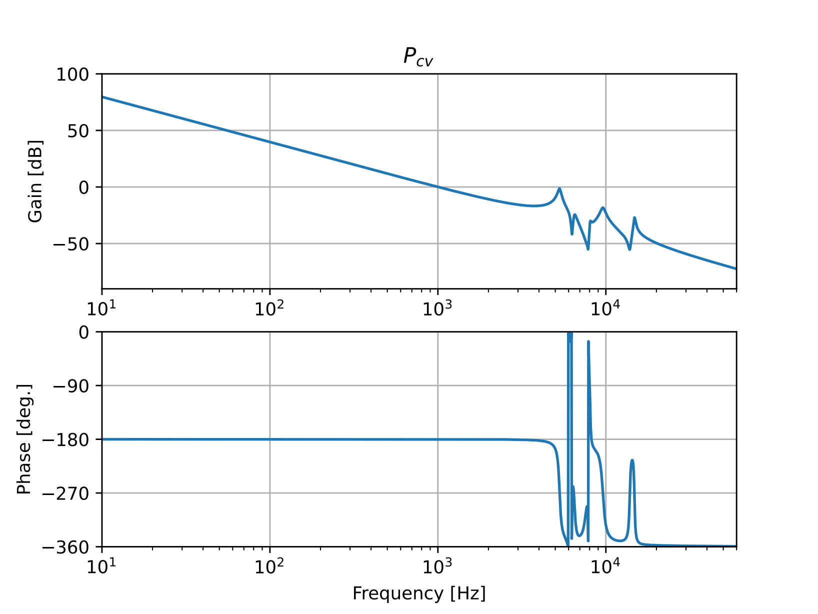

P_{cv}
100
50
0
−50
Gain [dB]
10^1
10^2
10^3
10^4
0
−90
−180
−270
−360
Phase [deg.]
10^1
10^2
10^3
10^4
Frequency [Hz]

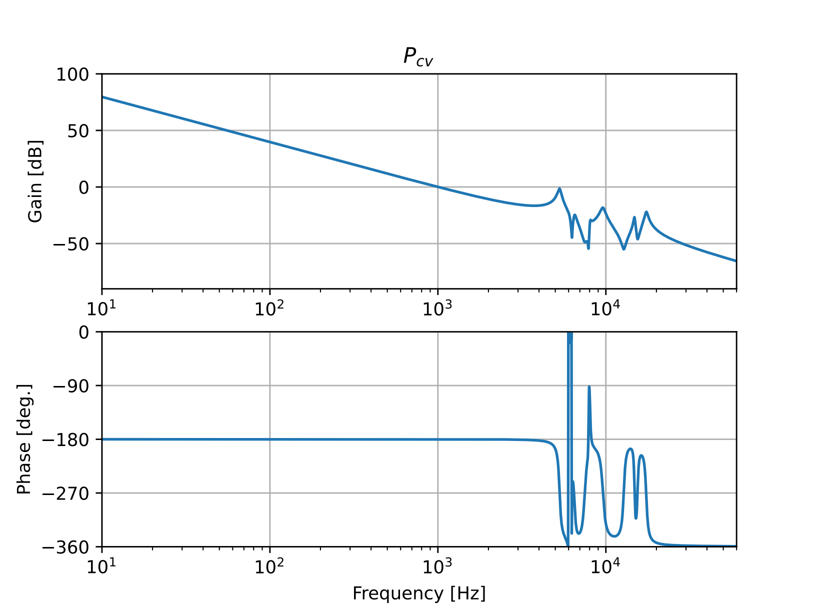


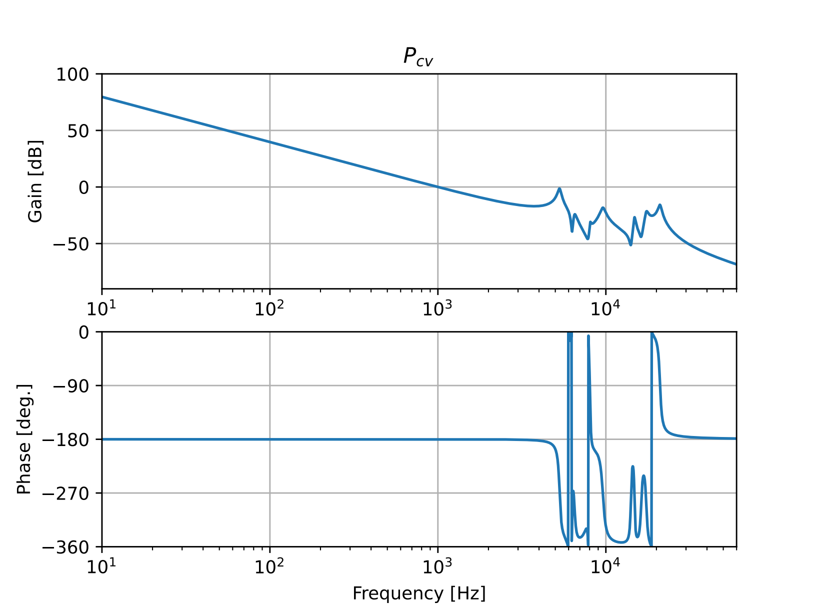
P_{cv}
100
50
0
−50
Gain [dB]
10^1
10^2
10^3
10^4
0
−90
−180
−270
−360
Phase [deg.]
Frequency [Hz]

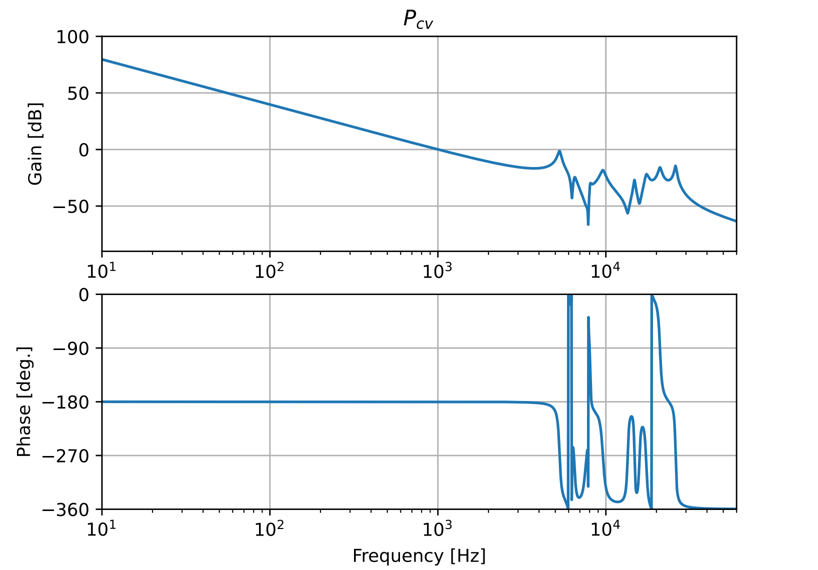


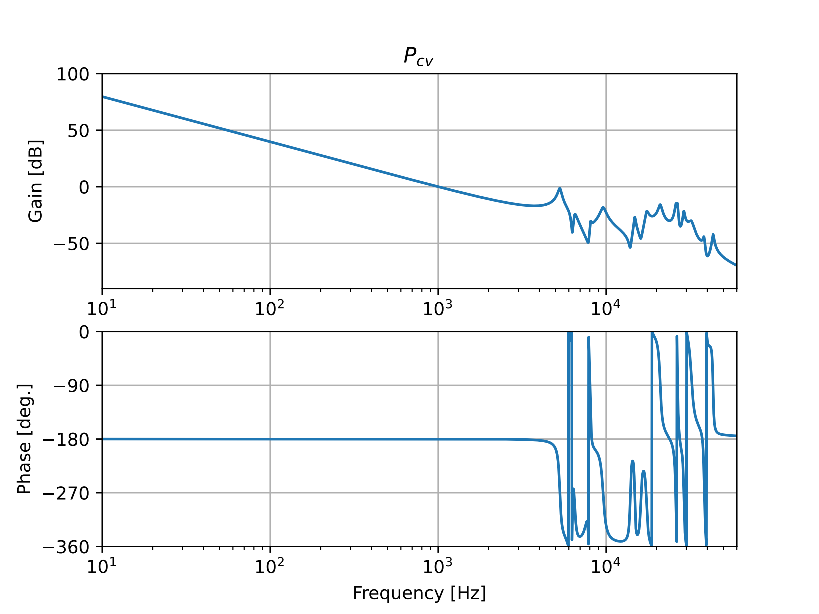
P_{cv}
100
50
0
−50
Gain [dB]
10^1
10^2
10^3
10^4
0
−90
−180
−270
−360
Phase [deg.]
10^1
10^2
10^3
10^4
Frequency [Hz]


P_{cv}
Gain [dB]
100
50
0
−50
10^1
10^2
10^3
10^4
Phase [deg.]
0
−90
−180
−270
−360
Frequency [Hz]


P_{cv}
Gain [dB]
100
50
0
−50
Phase [deg.]
0
−90
−180
−270
−360
10^1
10^2
10^3
10^4
Frequency [Hz]

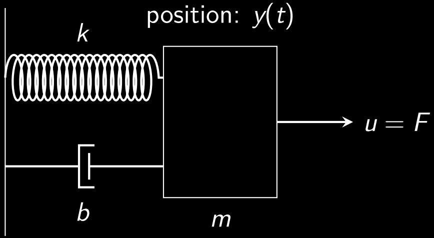

P_{cv}
100
50
0
−50
Gain [dB]
10^1
10^2
10^3
10^4
0
−90
−180
−270
−360
Phase [deg.]
10^1
10^2
10^3
10^4
Frequency [Hz]


P_{cv}
Gain [dB]
100
50
0
−50
10^1
10^2
10^3
10^4
Phase [deg.]
0
−90
−180
−270
−360
Frequency [Hz]

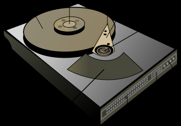

P_{cv}
Gain [dB]
100
50
0
−50
10^1
10^2
10^3
10^4
Phase [deg.]
0
−90
−180
−270
−360
10^1
10^2
10^3
10^4
Frequency [Hz]

Models of continuous-time systems

- modeled as differential equations:



$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t)\,, \;\; y(0) = y_0, \;\; \dot{y}(0) = \dot{y}_0$$



$$\ddot{\theta}_i + 2\zeta_i\omega_i\dot{\theta}_i + \omega_i^2\theta_i = \kappa_i\tau \Leftrightarrow \Theta_i(s) = \frac{\kappa_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}\mathrm{T}(s)$$

Models of continuous-time systems

General continuous-time systems:

$$\frac{d^n y(t)}{dt^n} + a_{n-1}\frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + b_{m-1}\frac{d^{m-1} u(t)}{dt^{m-1}} + \cdots + b_0 u(t)$$

with the initial conditions $y(0) = y_0, \ \ldots, y^{(n)}(0) = y_0^{(n)}$.

Models of discrete-time systems

General discrete-time systems

- inputs and outputs defined at discrete time instances $k = 1, 2, \ldots$

Models of discrete-time systems

General discrete-time systems

- inputs and outputs defined at discrete time instances $k = 1, 2, \ldots$
- described by ordinary difference equations in the form of

$$y(k)+a_{n-1}y(k-1)+\cdots+a_0y(k-n) = b_mu(k+m-n)+\cdots+b_0u(k-n)$$

Models of discrete-time systems

General discrete-time systems

- inputs and outputs defined at discrete time instances $k = 1, 2, \ldots$
- described by ordinary difference equations in the form of

$$y(k) + a_{n-1}y(k-1) + \cdots + a_0 y(k-n) = b_m u(k+m-n) + \cdots + b_0 u(k-n)$$

Example: bank statements

- $x(k+1) = (1+\rho)x(k) + u(k), x(0) = x_0$

Models of discrete-time systems

General discrete-time systems

- inputs and outputs defined at discrete time instances $k = 1, 2, \dots$
- described by ordinary difference equations in the form of

$$y(k) + a_{n-1}y(k-1) + \cdots + a_0 y(k-n) = b_m u(k+m-n) + \cdots + b_0 u(k-n)$$

Example: bank statements

- $x(k+1) = (1+\rho)x(k) + u(k), x(0) = x_0$
- k – month counter; ρ – interest rate; $x(k)$ – wealth at the beginning of month k; $u(k)$ – money saved at the end of month k; x_0 – initial wealth in account

Model properties: static v.s. dynamic, causal v.s. acausal

$$u \longrightarrow \boxed{\mathcal{M}} \longrightarrow y$$

Model $\mathcal{M}$ is said to be

- *memoryless* or *static* if $y(t)$ depends only on $u(t)$

Model properties: static v.s. dynamic, causal v.s. acausal

$$u \longrightarrow \boxed{\mathcal{M}} \longrightarrow y$$

Model $\mathcal{M}$ is said to be

- *memoryless* or *static* if $y(t)$ depends only on $u(t)$
- *dynamic* (has memory) if y at time t depends on input values at other times

Model properties: static v.s. dynamic, causal v.s. acausal

$$u \longrightarrow \boxed{\mathcal{M}} \longrightarrow y$$

Model $\mathcal{M}$ is said to be

- *memoryless* or *static* if $y(t)$ depends only on $u(t)$
- *dynamic* (has memory) if y at time t depends on input values at other times
- e.g.: $y(t) = \mathcal{M}(u(t)) = \gamma u(t)$, $y(t) = \int_0^t u(\tau) d\tau$, $y(k) = \sum_{i=0}^k u(i)$

Model properties: static v.s. dynamic, causal v.s. acausal

$$u \longrightarrow \boxed{\mathcal{M}} \longrightarrow y$$

Model $\mathcal{M}$ is said to be

- *memoryless* or *static* if $y(t)$ depends only on $u(t)$
- *dynamic* (has memory) if y at time t depends on input values at other times
- e.g.: $y(t) = \mathcal{M}(u(t)) = \gamma u(t)$, $y(t) = \int_0^t u(\tau) d\tau$, $y(k) = \sum_{i=0}^k u(i)$
- *causal* if $y(t)$ depends on $u(\tau)$ for $\tau \leq t$

Model properties: static v.s. dynamic, causal v.s. acausal

$$u \longrightarrow \boxed{\mathcal{M}} \longrightarrow y$$

Model $\mathcal{M}$ is said to be

- *memoryless* or *static* if $y(t)$ depends only on $u(t)$
- *dynamic* (has memory) if y at time t depends on input values at other times
- e.g.: $y(t) = \mathcal{M}(u(t)) = \gamma u(t)$, $y(t) = \int_0^t u(\tau) d\tau$, $y(k) = \sum_{i=0}^k u(i)$
- *causal* if $y(t)$ depends on $u(\tau)$ for $\tau \leq t$
- *strictly causal* if $y(t)$ depends on $u(\tau)$ for $\tau < t$, e.g.: $y(t) = u(t - 10)$

Linearity and time-invariance

The system $\mathcal{M}$ is called

- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2

Linearity and time-invariance

The system $\mathcal{M}$ is called

- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2

- *time-invariant* if its properties do not change with respect to time

Linearity and time-invariance

The system $\mathcal{M}$ is called

- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2
- *time-invariant* if its properties do not change with respect to time
- e.g., $\dot{y}(t) = Ay(t) + Bu(t)$ is linear and time-invariant

Linearity and time-invariance

The system $\mathcal{M}$ is called

- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2
- *time-invariant* if its properties do not change with respect to time
- e.g., $\dot{y}(t) = Ay(t) + Bu(t)$ is linear and time-invariant
- $\dot{y}(t) = 2y(t) - \sin(y(t))u(t)$ is nonlinear, yet time-invariant

Linearity and time-invariance

The system $\mathcal{M}$ is called

- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2
- *time-invariant* if its properties do not change with respect to time
- e.g., $\dot{y}(t) = Ay(t) + Bu(t)$ is linear and time-invariant
- $\dot{y}(t) = 2y(t) - \sin(y(t))u(t)$ is nonlinear, yet time-invariant
- $\dot{y}(t) = 2y(t) - t\sin(y(t))u(t)$ is time-varying

Linearity and time-invariance

The system $\mathcal{M}$ is called

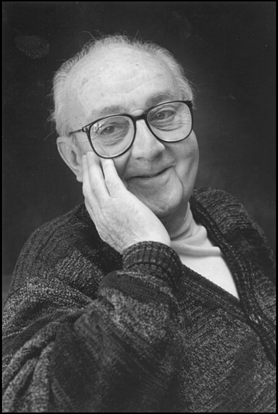
- *linear* if satisfying the *superposition* property:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t))$$

  for any input signals $u_1(t)$ and $u_2(t)$, and any real numbers α_1 and α_2

- *time-invariant* if its properties do not change with respect to time
- e.g., $\dot{y}(t) = Ay(t) + Bu(t)$ is linear and time-invariant
- $\dot{y}(t) = 2y(t) - \sin(y(t))u(t)$ is nonlinear, yet time-invariant
- $\dot{y}(t) = 2y(t) - t\sin(y(t))u(t)$ is time-varying
- assuming the same initial conditions, if we shift $u(t)$ by a constant time interval, i.e., consider $\mathcal{M}(u(t+\tau_0))$, then $\mathcal{M}$ is time-invariant if the output $\mathcal{M}(u(t+\tau_0)) = y(t+\tau_0)$

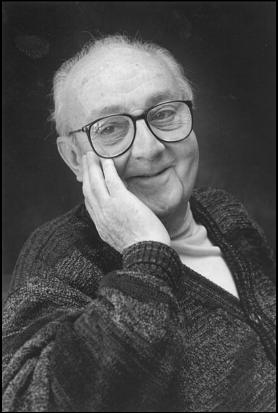
George Box



"All Models are Wrong, but Some are Useful"

- statistical models always fall short of the complexities of reality but can still be useful nonetheless

George Box



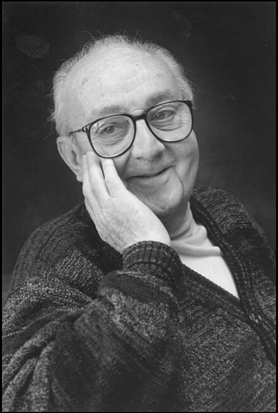
"All Models are Wrong, but Some are Useful"

- statistical models always fall short of the complexities of reality but can still be useful nonetheless
- a dynamic system may simply be too complex (consider the neural system of human brains)

George Box

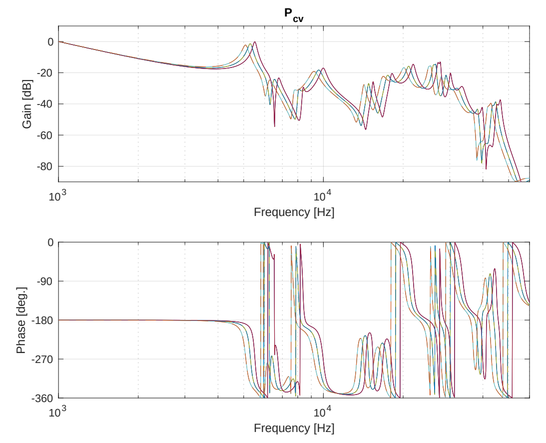
“All Models are Wrong, but Some are Useful”

- statistical models always fall short of the complexities of reality but can still be useful nonetheless
- a dynamic system may simply be too complex (consider the neural system of human brains)
- or there are inevitable hardware uncertainties such as the fatigue of gears or bearings in a car

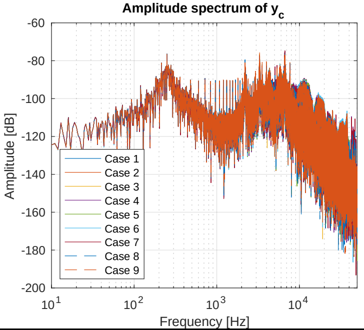


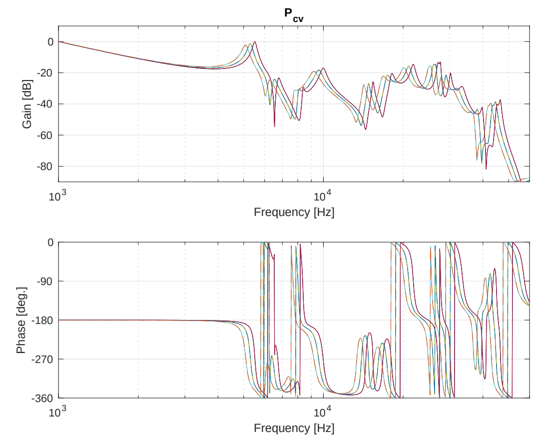
Example (HDDs under perturbation)

- temperature influence
- manufacturing variations
- but, control works!

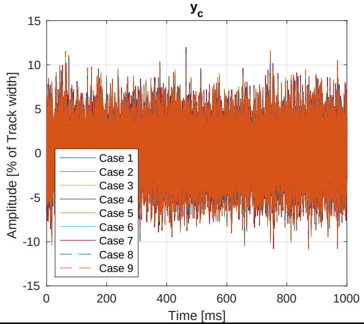


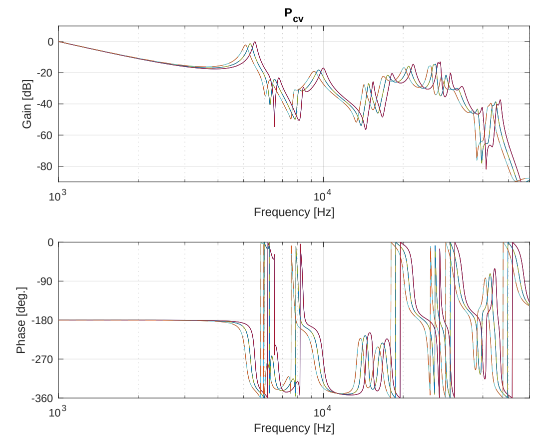
Example (HDDs under perturbation)



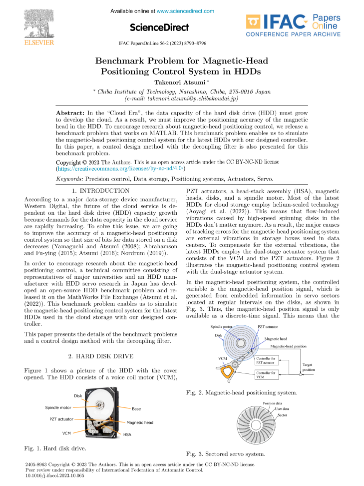


Example (HDDs under perturbation)





Example (HDDs under perturbation)



Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 8790–8796

Benchmark Problem for Magnetic-Head Positioning Control System in HDDs

Takenori Atsumi *

* *Chiba Institute of Technology, Narashino, Chiba, 275-0016 Japan (e-mail: takenori.atsumi@p.chibakoudai.jp)*

Abstract: In the “Cloud Era”, the data capacity of the hard disk drive (HDD) must grow to develop the cloud. As a result, we must improve the positioning accuracy of the magnetic head in the HDD. To encourage research about magnetic-head positioning control, we release a benchmark problem that works on MATLAB. This benchmark problem enables us to simulate the magnetic-head positioning control system for the latest HDDs with our designed controller. In this paper, a control design method with the decoupling filter is also presented for this benchmark problem.

Keywords: Precision control, Data storage, Positioning systems, Actuators, Servo.

1. INTRODUCTION

According to a major data-storage device manufacturer, Western Digital, the future of the cloud service is dependent on the hard disk drive (HDD) capacity growth because demands for the data capacity in the cloud service are rapidly increasing. To solve this issue, we are going to improve the accuracy of a magnetic-head positioning control system so that size of bits for data stored on a disk decreases (Yamaguchi and Atsumi (2008); Abrahamson and Fu-ying (2015); Atsumi (2016); Nordrum (2019)).

In order to encourage research about the magnetic-head positioning control, a technical committee consisting of representatives of major universities and an HDD manufacturer with HDD servo research in Japan has developed an open-source HDD benchmark problem and released it on the MathWorks File Exchange (Atsumi et al. (2022)). This benchmark problem enables us to simulate the magnetic-head positioning control system for the latest HDDs used in the cloud storage with our designed controller.

This paper presents the details of the benchmark problems and a control design method with the decoupling filter.

2. HARD DISK DRIVE

Figure 1 shows a picture of the HDD with the cover opened. The HDD consists of a voice coil motor (VCM),

Fig. 1. Hard disk drive.

PZT actuators, a head-stack assembly (HSA), magnetic heads, disks, and a spindle motor. Most of the latest HDDs for cloud storage employ helium-sealed technology (Aoyagi et al. (2022)). This means that flow-induced vibrations caused by high-speed spinning disks in the HDDs don’t matter anymore. As a result, the major causes of tracking errors for the magnetic-head positioning system are external vibrations in storage boxes used in data centers. To compensate for the external vibrations, the latest HDDs employ the dual-stage actuator system that consists of the VCM and the PZT actuators. Figure 2 illustrates the magnetic-head positioning control system with the dual-stage actuator system.

In the magnetic-head positioning system, the controlled variable is the magnetic-head position signal, which is generated from embedded information in servo sectors located at regular intervals on the disks, as shown in Fig. 3. Thus, the magnetic-head position signal is only available as a discrete-time signal. This means that the

Fig. 2. Magnetic-head positioning system.

Fig. 3. Sectored servo system.

10.1016/j.ifacol.2023.10.065