Lecture 2: Experiment 1
EE380 (Control Systems)

Manavaalan Gunasekaran
PhD student
manvaal@iitk.ac.in

Arun Kant Singh
PhD student
arunkant@iitk.ac.in

Ramprasad Potluri
Associate Professor
potluri@iitk.ac.in

Department of Electrical Engineering
Indian Institute of Technology Kanpur

August 05, 2011

Contents

Outline of the experiment

Want to control the speed of a pmdc motor. Steps:

- Mathematical modeling
- Design using loop-shaping
- Simulation on PC
- Deployment on experimental setup

Tasks common to all 6 experiments

Simulation

- Perform PC-based simulation of CL system using GNU Octave.
- Perform PC-based simulation of digital control of a continuous-time system using GNU Octave.

Realization on hardware

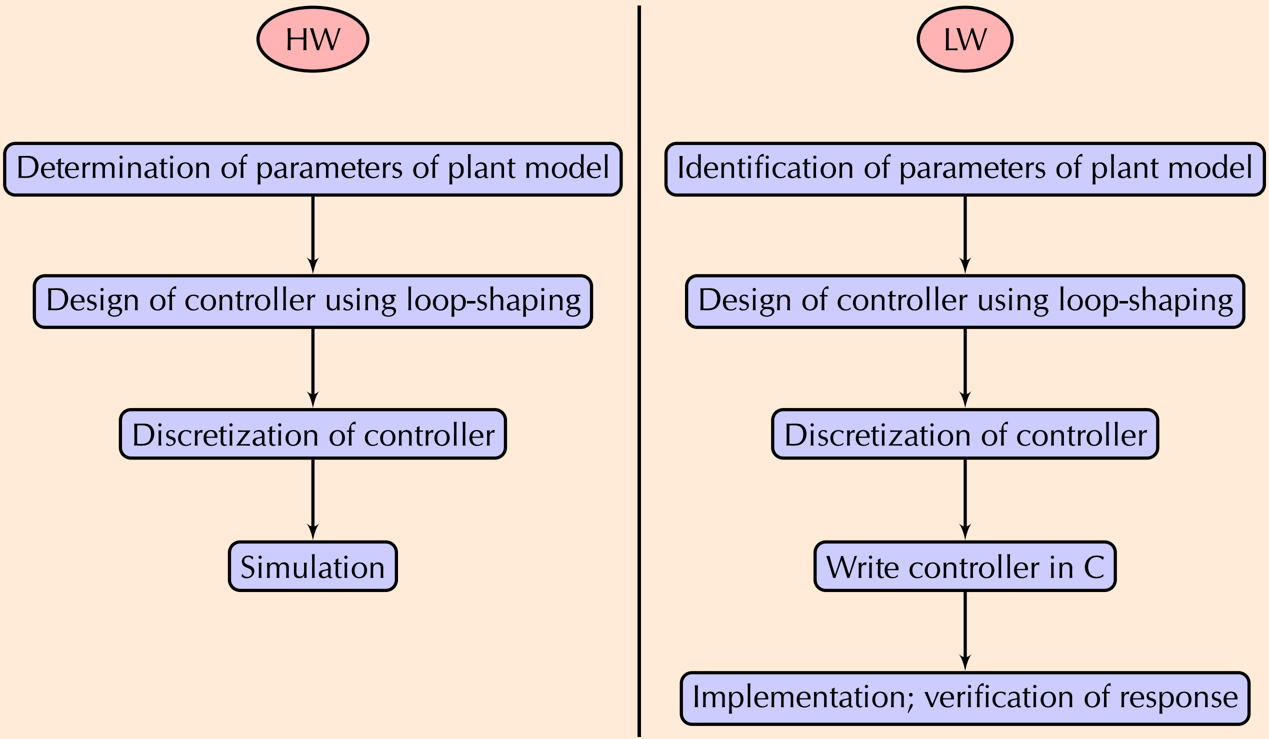
- Utilize the various components of an integrated development environment (IDE): editor, compiler, linker, debugger, and programmer to program a μC.
- Program controller using C language into μC.
- Monitoring: read data into PC from μC using UART modules.

Analysis

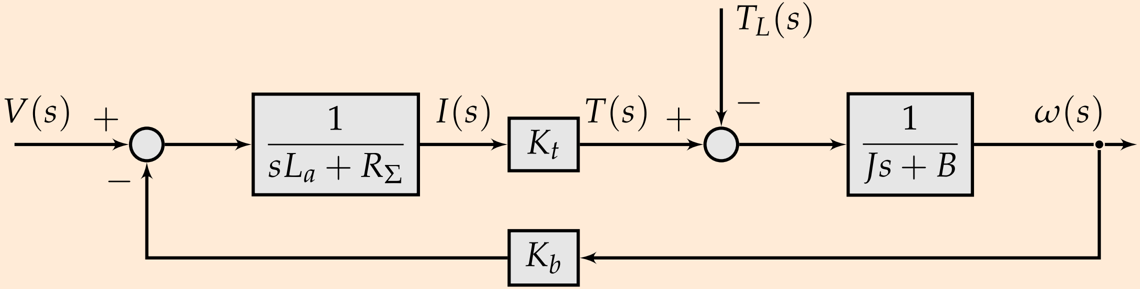
- Compare actual performance with predicted performance.

Homework (HW) vs. Lab work (LW)

HW meant to help practice most of LW in advance.



HW: Determine parameters of plant model



$$\frac{\omega(s)}{V(s)} = \frac{K_m}{\tau_m s + 1}$$

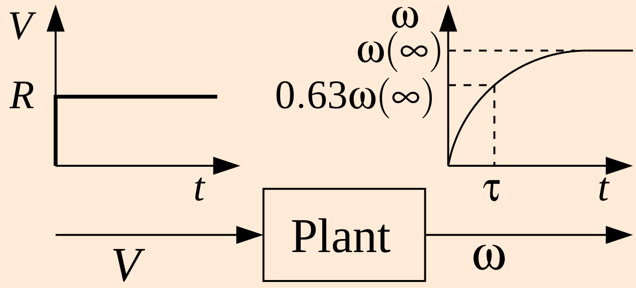
with

$$K_m = \frac{K_T}{R_\sum B + K_T K_b},$$

$$\tau_m = \frac{R_\sum J}{R_\sum B + K_T K_b}$$

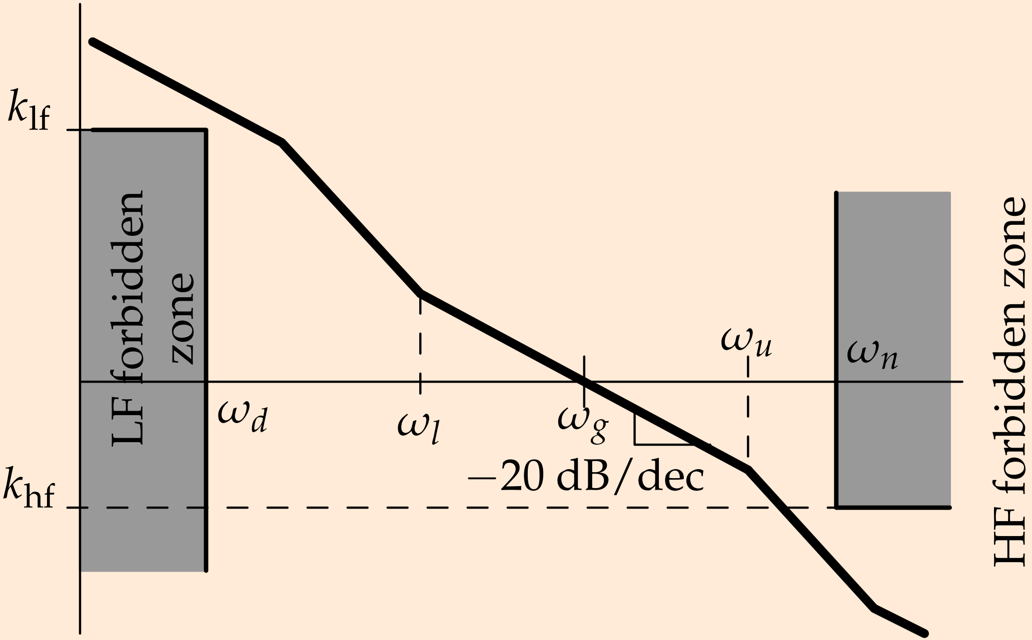
- K_T, K_b given in Table 1.1.
- J is rotor inertia in Table 1.1.
- B calculated in §1.4.1.
- R_Σ calculated in §1.4.2.

LW: Identification



- Plant = H-bridge + pmdc motor.
- Assume form $K/(\tau s + 1)$.
- $\omega(t) = RK(1 - e^{-t/\tau})$
- τ is time where speed is $0.6321\omega(\infty)$
- $RK = \omega(\infty)$.

Loop-shaping (1/5): Typical G_{des}

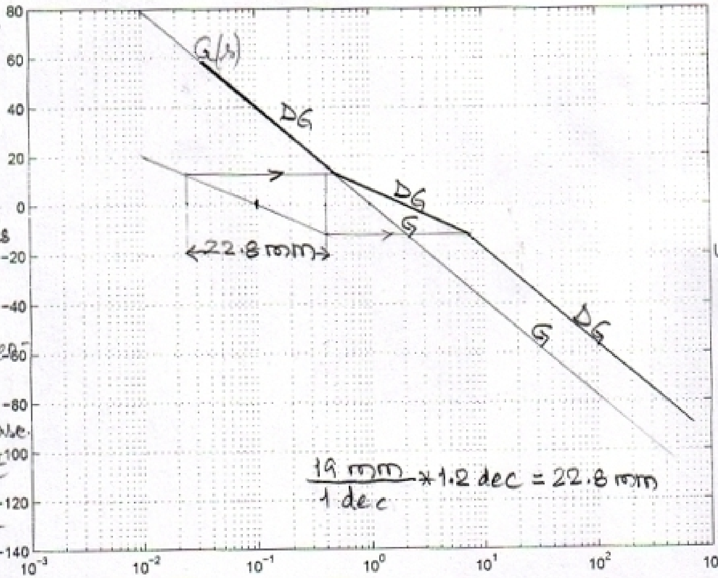


Loop-shaping (2/5): Example

2. The uncompensated unity feedback CL TF of a system is $M(s) = \frac{1}{s^2+1}$. Design a compensator (decide between lead and lag) that will provide the CL system's step response a damping coefficient of $\zeta \geq 0.55$.

step 1:
The OL TF is
$G(s) = \frac{1}{s^2}$

step 2:
The phase of $G(j\omega)$ is $-180°$ across all ω.
A lag compensator adds negative phase. So, it cannot create a (+ve) PM for this $G(s)$.
So, we choose a lead compensator.



step 3:
$\xi \geq 0.55 \Rightarrow PM \geq 55°$
$\Rightarrow$ let's decided to provide a $PM = 60°$. For this we will create DG such that it has a -20 dB/decade section that is 1.2 dec. wide and is centered about ω_g. Also, since there are no other spec-s, we can retain the present value of K_a.

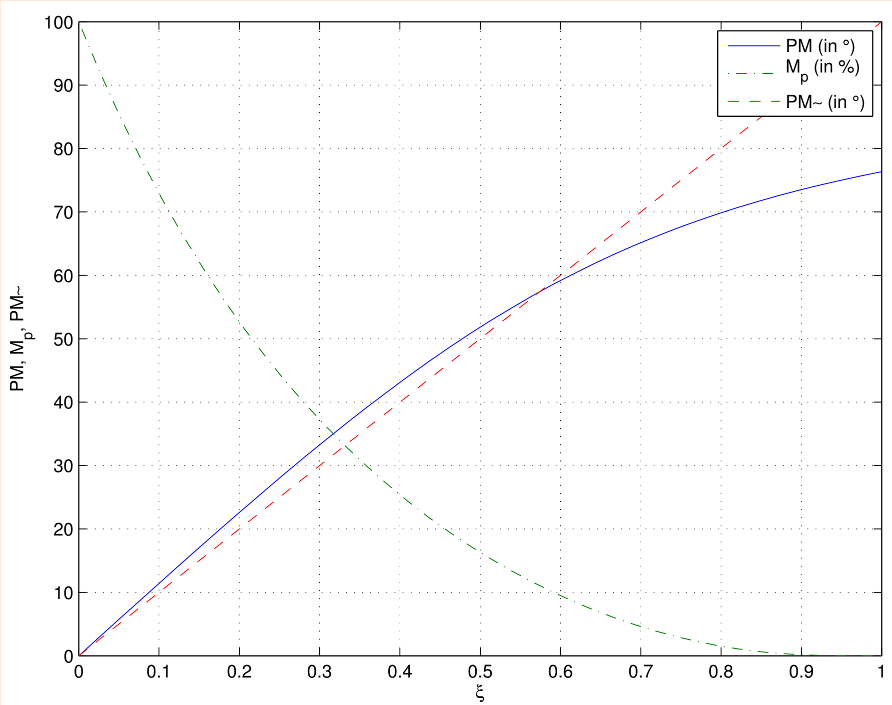
Step 4: Construct a 22.8 mm-wide -20 dB/dec section symmetrically positioned about 0.1 rad/s as shown. Use this section to complete the ABMP of DG as shown.

Instructor: Ramprasad Potluri, E-mail: potluri@iitk.ac.in. Office: WL217A, Lab: WL217B, Phones: (0512) 259-8837, 259-7735.
Assignment posted on March 20, 2009.

$$DG = G_{\text{des}}$$

Loop-shaping (3/5): $\zeta \longleftrightarrow M_p \longleftrightarrow \mathrm{PM}$

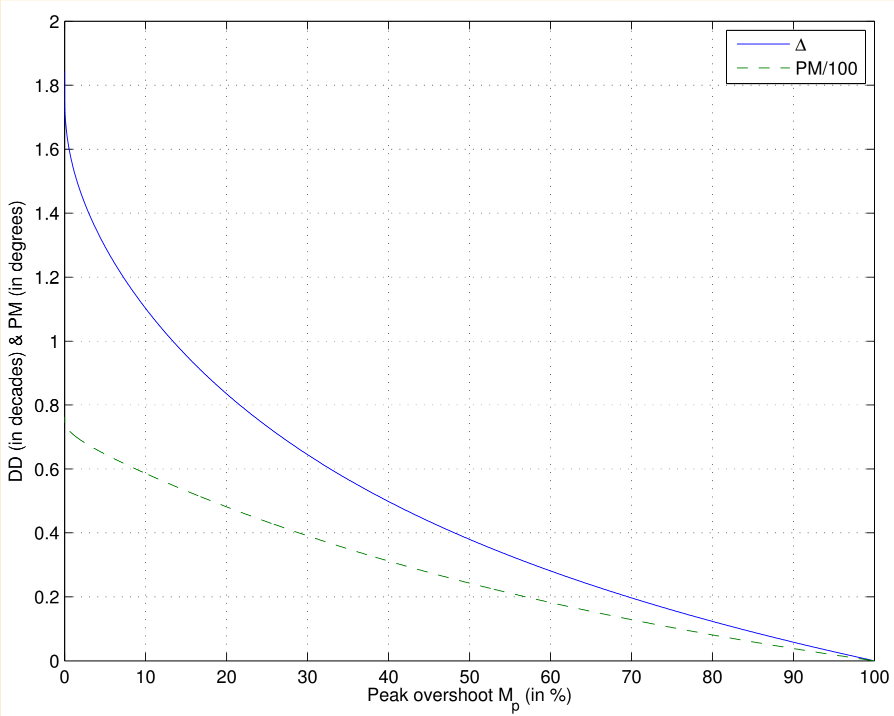
From Lecture 26 of EE250-2011



Back
Forward
Close

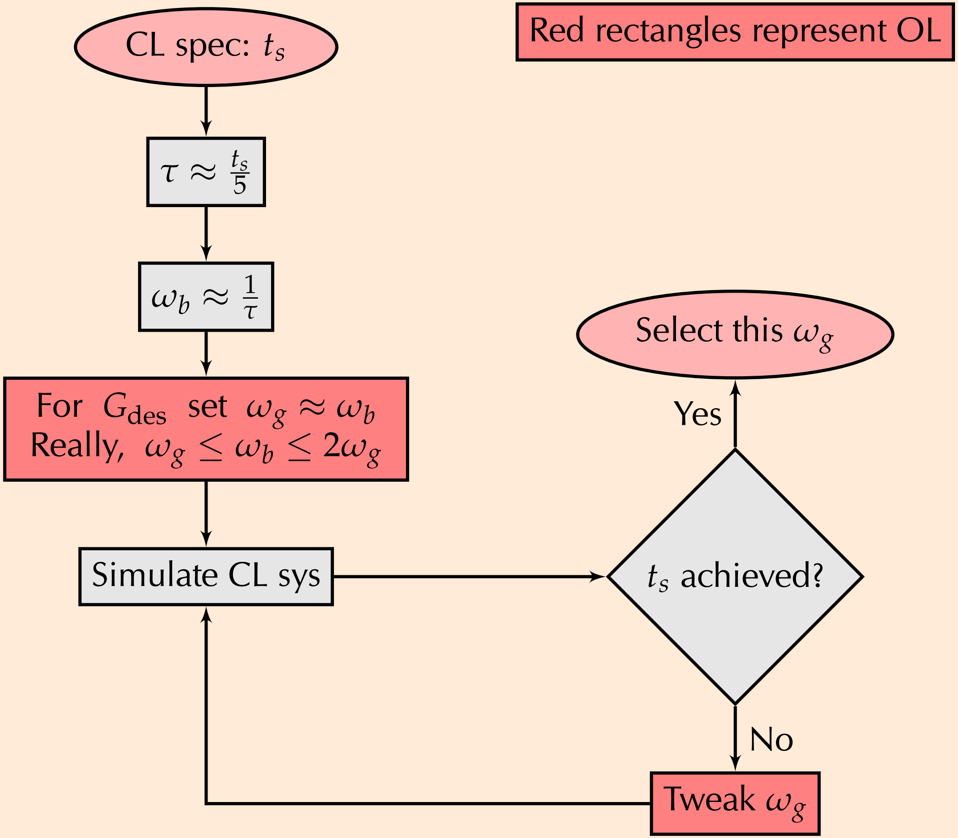
Loop-shaping (4/5): $M_p \longleftrightarrow$ DD $\longleftrightarrow$ PM

From Lecture 26 of EE250-2011



Back
Forward
Close

Loop-shaping (5/5): Determination of ω_g



Back
Forward
Close

Discretization

$$\frac{a_1 s + a_0}{s^2 + b_1 s + b_0}$$

Simulation diagram

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -b_0 x_1 - b_1 x_2 + u \\ y &= a_0 x_1 + a_1 x_2 \end{aligned}$$

Group terms

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Euler's approximation

$$\begin{aligned} x_{k+1} &= (A\Delta t + I)x_k + B\Delta t u_k, \\ y_k &= Cx_k + Du_k \end{aligned}$$

Simulate; LW: C code, Implement, Analyze

- Simulation: `easysim.m`
- Discretized controller
  $\longrightarrow$ C code:

- Implement: As in demo slides
- Analyze: Compare results

$$\begin{aligned} x_1(k+1) &= a_{11}x_1(k) + a_{12}x_2(k) + b_1u(k) \\ x_2(k+1) &= a_{21}x_1(k) + a_{22}x_2(k) + b_2u(k) \\ y(k) &= c_1x_1(k) + c_2x_2(k) + du(k) \end{aligned}$$

In `main-prog.c` before `main()` insert `float x1[2],x2[2];`
In `main()` insert `x1[0] = x2[0] = 0;`

```
x1[1] = a11 * x1[0] + a12 * x2[0] + b1 * u;
x2[1] = a21 * x1[0] + a22 * x2[0] + b2 * u;
    y = c1 * x1[0] + c2 * x2[0] + d * u;
x1[0] = x1[1];
x2[0] = x2[1];
```

Back
Forward
Close