Lecture 1: Overview of the laboratory
EE380 (Control Systems)

Ramprasad Potluri

Associate Professor

potluri@iitk.ac.in

Manavaalan Gunasekaran

PhD student

manvaal@iitk.ac.in

Department of Electrical Engineering

Indian Institute of Technology Kanpur

July 30, 2011

Contents

Aim of these control experiments

- Reinforce paper-based and PC-based design techniques.
- Help students acquire skills in converting design into practical system.

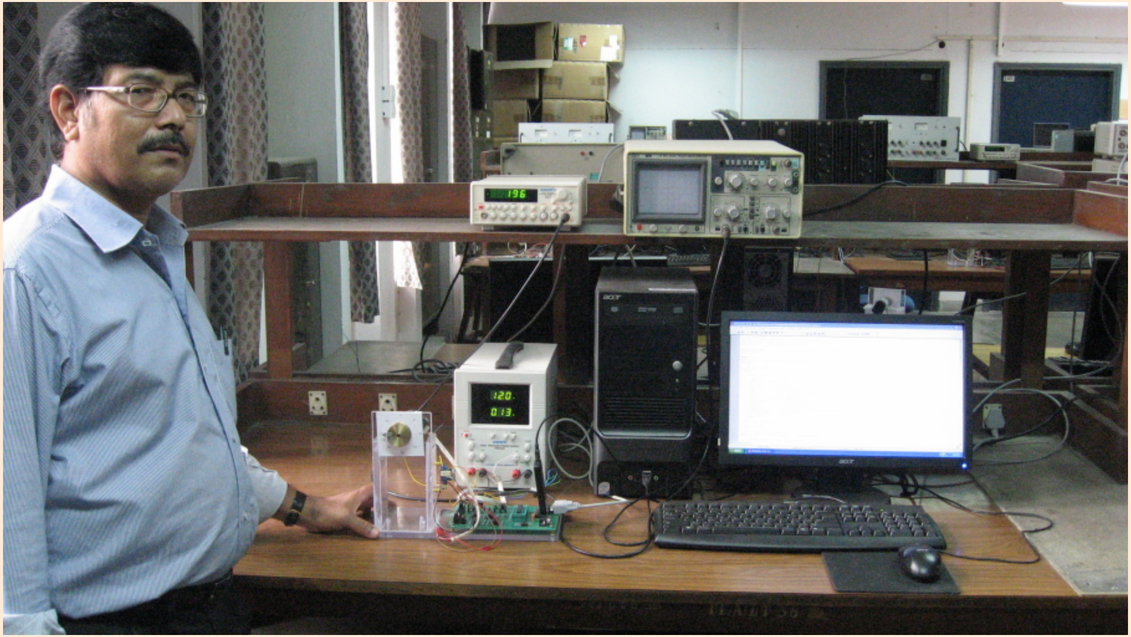
Help student develop confidence to say,
“I have practical experience with implementing control systems in addition to discussing them theoretically.”

Skills to be acquired in design $\longrightarrow$ practice

- Ability to identify the hardware & software that are needed in a basic control system.
- Ability to debug small errors that may show up during practical implementation.

These skills come only through at least a few weeks of work on problems, all of which may be related to one or two hardware setups that are not complex, and do not look complex.

The work bench



Equipment on each bench (top to bottom): Function generator, motor control setup, power supply, PC, programmer (PICkit 2).

List of experiments

1. PMDC motor modeling, identification, speed control
2. Speed of PMDC motor tracks reference sinusoid
3. Ziegler-Nichols tuning of speed controller of PMDC motor
4. Speed control using feedback of current
5. Current control
6. Disturbance observer

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.

Contents of Experiment 1

PMDC motor modeling, identification, speed control

- Develop mathematical model for PMDC motor using datasheets.
- Develop mathematical model for H-bridge – PMDC motor system using open-loop step response.
- Design negative feedback controller using Bode plot-based loop-shaping techniques.

Extra skills user develops

- Ability to read datasheets.
- System identification using step response.

Contents of Experiment 2

Speed of PMDC motor tracks reference sinusoid

- Identify the parameters of the mathematical model of H-bridge – PMDC motor system using least squares estimation (LSE).
- Design speed controller using loop-shaping to track sinusoids.

Extra skills user develops

- System identification using LSE.
- Understanding of the effects of nonlinearities.

Questions

- Where is problem of tracking sinusoids encountered in practice?
- Why test if a control system can track sinuosids?

Contents of Experiment 3

Ziegler-Nichols tuning of speed controller of PMDC motor

Extra skills user develops

- One tuning technique.
- Bringing multiple different techniques from classical control theory to bear on the problem of tuning: root locus, Routh-Hurwitz criterion, Nyquist criterion.

Contents of Experiment 4

Speed control using feedback of current

Extra skills user develops

- Estimation of speed using armature current.
- Know a way to keep loop working if encoder breaks down.

Contents of Experiment 5

Current control

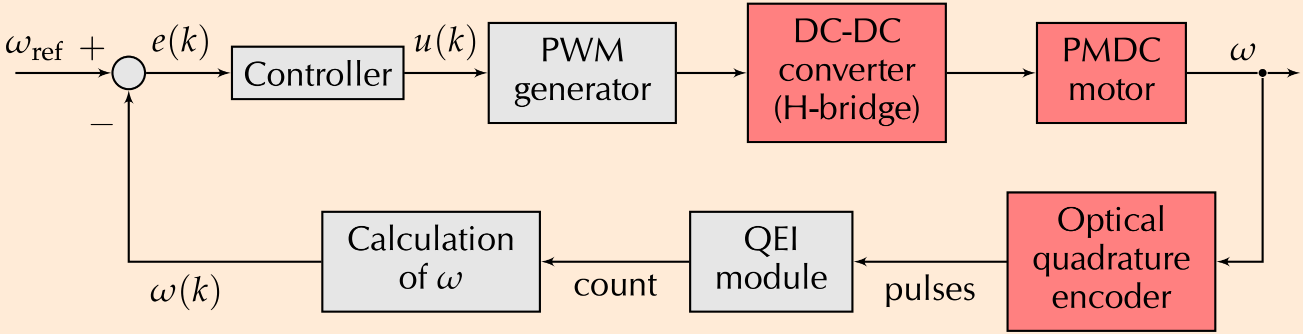
See slides on Experiment 5.

Contents of Experiment 6

Disturbance observer

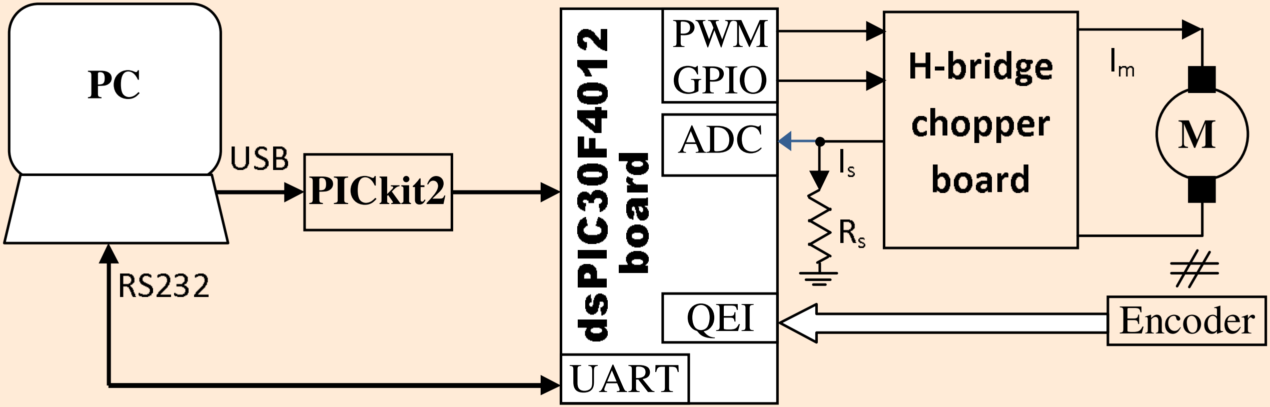
See slides on Experiment 6.

Block diagram of speed control scheme



- Grey part of block diagram is inside μC.
- Red part of block diagram is outside μC.

Block diagram of experimental setup



- Setup primarily meant to implement speed/position control.
- We may choose to use feedback of ω or i or both as necessary.
- Here, i is motor current.

  In Figure 1.6 of manual, which of I_{sens} and I_m does i represent?

Modules of μC required for motor control

- Timer: Timer interrupts mark sampling instants.
- QEI: Counts pulses from quadrature encoder.
- ADC: Reads analog inputs from outside μC.
- PWM: Produces variable duty ratio fixed frequency rectangular waveform.
- UART: Helps communicate with serial port of PC.