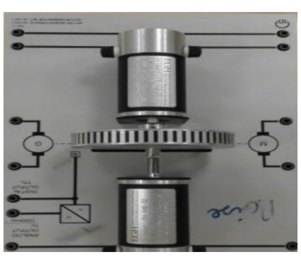
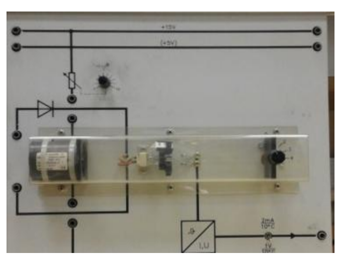
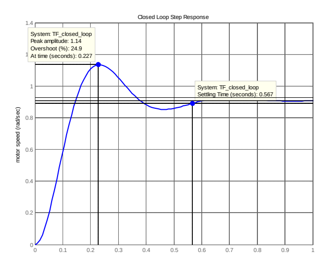
King Saud University

College of Engineering, Electrical Engineering Department

Labwork Manual
EE 356 Control and Instrumentation Laboratory
(كهر 356 معمل التحكم و القياسات)







Student Name : ……………………………………

Student ID : ……………………………………

Academic Year : ……………………………………

[Updated: September 2015]

EE356 Control and Instrumentation Laboratory

Course Description:

Experiments to support control theory using physical processes (e.g. water level, temperature control, light intensity control, etc); Control system simulation using Matlab; Modeling of physical (experimental) equipment; Static performance; Transient analysis; Measuring devices; Two-position control; Proportional control; PID control; Introduction to Electrical instrumentation and Measurements.

Textbook: "Modern Control Systems", Dorf and R. Bishop, Addison-Wesley, 1998.

Co-requisite: *EE 351Automatic Control* [*EE KSU new plan, 2008*]

Grading Policy:

- Participation (attendances, activity, … etc) = 10 %
- Reports = 20 %
- Midterm = 30 %
- Final Exam = 40 %

General Instructions:

- ✓ Laboratory work is only permitted during scheduled periods and **under supervision** from the instructor
- ✓ Each student is required to bring: **Manual**, pencil, blank paper, etc
- ✓ Each student must do all laboratory sessions with his group (or section). He should NOT join other sections without **written permission** from the instructor.
- ✓ **Eating and drinking** are NOT permitted in the laboratory. So, don't bring bottled water into the laboratory.
- ✓ Students must follow carefully the procedure given in the manual sheet. They should not carry out experiments or make innovations without the approval of the instructor. They should ask the instructor if they do not fully understand the instructions.
- ✓ If there is any problem with the equipment, the student should inform the instructor. The student **should not remove/replace the equipment** set-up by himself.
- ✓ After finish the experiment, the student should **switch off** the PC and **un-plug** all cables (or connections).

[This manual is updated by Sutrisno Ibrahim in September 2015, contact: suibrahim@ksu.edu.sa]

Report Outline

Student name: Student ID: Group: Signature:	....	Course Code: Experiment #: Title:	....

Objectives:
-
-
-
-

Results:
-
-
-
-

Comments:
-
-
-
-

Conclusion:
-
-
-
-

Table1: ....

Figure1: ....

Table2: ....

Figure2: ....

Contents

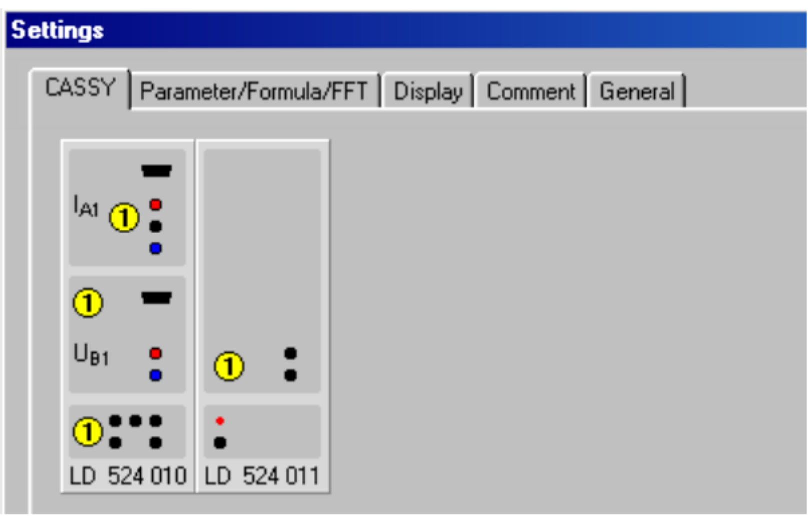
Experiment # CASSY INTRODUCTION

Introduction

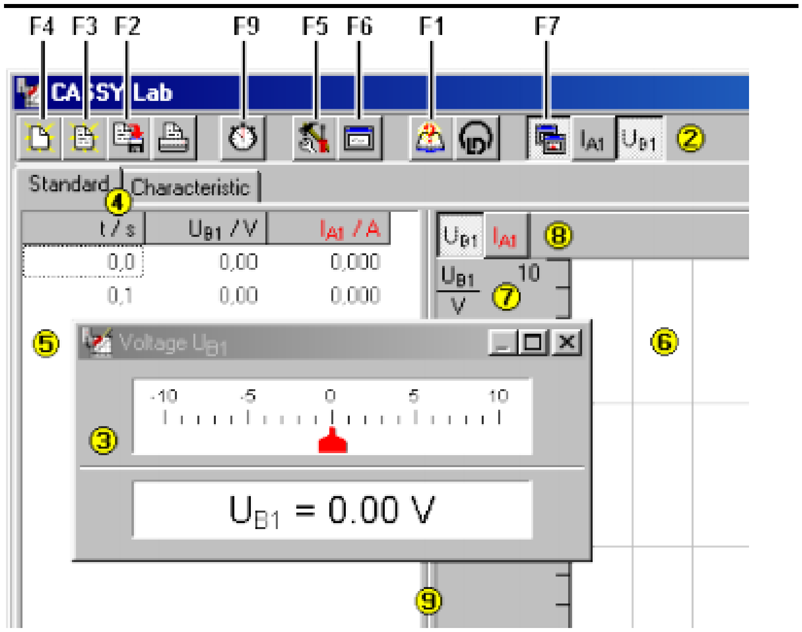
This experiment is intended to provide overview of CASSY lab software. Make sure that CASSY software is already properly installed and the CASSY module is ON and connected with the PC. CASSY® is a registered trademark of LD Didactic GmbH.

First measurements

When the software detects one or more CASSY devices, the CASSY tab of the setup dialog (**F5**) displays the current configuration (including any attached sensor boxes). To conduct a measurement, just click on the corresponding input or output ①.



An active input or output (channel) is then marked in colour and placed among the speed buttons ② at the top right of the main window (here I_{A1} and U_{B1}). These buttons are the fastest way to display or close a display instrument ③ for that channel (left mouse button) or to change a setting (right mouse button). In addition, the channel initially appears automatically in the table ⑤ and in the diagram ⑥.



You can access the basic functions quickly using the speed buttons ② in the top bar. The most important speed buttons are also mapped to function keys.

Below this bar, you can toggle between table ⑤ and diagram ⑥ display by clicking on one of the display tabs ④ when different display modes have been defined (here **Standard** and **Characteristic**). The table and diagram can be enlarged or reduced with respect to each other by moving the boundary ⑨ with the mouse.

At many points, **both** mouse buttons (left and right) can be used to execute different functions:

Control element	**Left mouse button**	**Right mouse button**
① CASSY setup	Activate and modify a channel	Activate and modify a channel.
② Channel button	Open and close a display instrument, drag and drop in ② and ⑤ through ⑧	Set up a channel
③ Display instrument	Move boundary between analog and digital display, drag and drop values in ⑤	Set up a channel
④ Name of display	Toggle to another defined display	
⑤ Table	Edit measured values, drag and drop values within the table or the channels to ②	Set display attributes of table, e.g. text size, delete rows and measurement series
⑥ Diagram	Mark evaluation ranges	Settings and evaluations in diagram
⑦ Scale	Move scale	Set minimum, maximum and conversion of scale
⑧ Axis symbols	Toggle y-scale, drag and drop into ②	Set up a channel
⑨ Boundary	Move boundary between channel and diagram	

The hot-key assignments of the speed buttons ② can often enable you to use the software more efficiently.

 **F4**

Clears either the current measurement while retaining the settings or, when no measurement is displayed, the current settings.

Pressing this key twice clears the current measurement with all settings.

 **F3**

Opens a measurement series with its settings and its evaluations.

It is also possible to append a measurement series to an existing series (without having to load its settings and evaluations as well). This is possible when the same measurement quantities are used for all series. Alternatively, a further measurement series can be measured and appended subsequently.

The software is also equipped with an ASCII import filter (file type ***.txt**).

 **F2**

Saves the current measurement series with its settings and its evaluations.

You can also save just the settings (without measurement data) to make it easier to repeat an experiment at a later date.

The software is additionally equipped with an ASCII export filter (file type ***.txt**). However, you can also open the CASSY Lab files (file type ***.lab**) using any text editor.

Prints out the currently active table or diagram.

 **F9**

Starts and stops a new measurement.

Alternatively, you can stop measurements by setting a measuring time.

 **F5**

Changes the current settings (e.g. CASSY, Parameter/Formula/FFT, Display, Comment, Serial Interface). This function must be activated **twice** for the measuring parameters.

 **F6**

Toggles large display of the status-line information on and off.

 **F1**

Opens this help file.

Displays the current version number of the software and enables entry of the activation code.

 **F7**

Closes all open display instruments or reopens them.

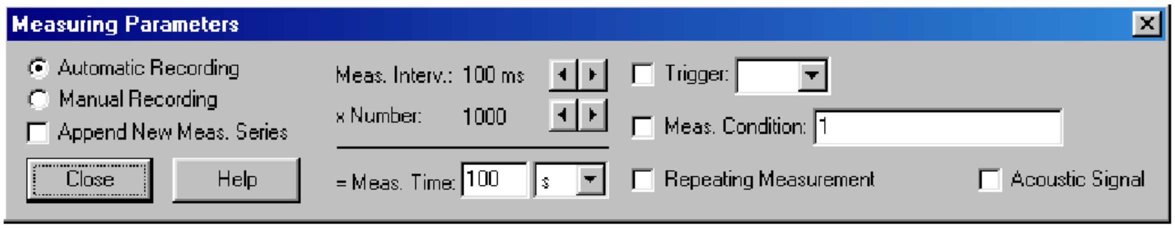
Measurement

 **F9**

Starts and stops a new measurement. You can use the right mouse button to open the table display menu in the table and the evaluation menu in the diagram.

 **F5**

Allows you to change the settings and (when activated **twice**) the measuring parameters which control the actual measurement.



The defaults that appear in this dialog depend on the currently connected sensor boxes. This simplifies matching to a particular measuring task, as the typical sensor box configuration is already finished. For measurements with the **MCA box** this window looks different.

Automatic recording

The software determines the exact time for the recording of a measured value. After a measurement is started (e.g. with **F9**), the software first waits for any **trigger** that may have been set, and then records one measured-value row each time the predefined time interval elapses. The **interval**, the **number** of measuring points per measurement as well as the total **measuring time** can be matched to the individual requirements before starting the experiment. You can select continuous display by setting a **repeating measurement**.

At time intervals above 100 ms, the software evaluates a **measuring condition** in addition to the trigger, and can also emit an **acoustic signal** when a measured value is recorded. The measuring condition is a formula. A formula result not equal to 0 means ON="Measured-value recording possible", while a formula result equal to 0 means OFF="Measured-value recording inhibited". The measurement then runs for as long as the measurement is started **and** the formula returns the result ON. For example, if you want to run a measurement on 21 April 1999 between 1:00 p.m. and 2:00 p.m. (13:00 and 14:00 hours), you can use the formula: date = 21.4.1999 and time >= 13:00 and time <= 14:00.

For some measurement quantities (e.g. rate, frequency, transit time, obscuration time, path when using the GM box or the timer box), the software does not evaluate the specified time interval. In this case the measurement is controlled by the gate time or the measurement pulses themselves.

Manual recording

The user determines the exact time for the recording of a measured value. At each start (e.g. with **F9**), the software records precisely **one** measured-value row, i.e. the current display values of the instruments are entered in the table and the diagram. Thus, manual recording must be executed repeatedly in order to capture a complete measurement series.

Append new measurement series

The Append function enables sequential recording of multiple measurement series. All measurement series are displayed in the table and the diagram together. A different color is used to display each additional measurement series.

Alternatively, the individual measurement series can first be recorded one after another and saved individually. When loading multiple comparable measurement series (with identical quantities), measured series can also be appended "retroactively".

Graphical Evaluations

You can access a wide variety of powerful graphical evaluation functions in the diagram by clicking the right mouse button.

- Change Axis Assignment
- Display Coordinates
- Select Line Width
- Select Value Display
- Select Rulers
- Show Grid
- Zoom
- Zoom Off
- Set Marker
- Text
- Vertical Line
- Horizontal Line
- Measure Difference
- Draw Mean
- Fit Function
- Calculate Integral
- Calculate Poisson Distribution
- Calculate Gaussian Distribution
- Calculate Peak Center
- Calculate Form Factor
- Calculate Ripple
- Fit Gaussian curves
- Find Equivalence Point
- Find Systole and Diastole
- Delete Last Evaluation
- Delete All Evaluations
- Delete Range (only Measured Values)
- Copy Diagram/Window

Change Axis Assignment

Activates the Display tab. This lets you change the assignment of the x-axis and up to 8 y-axes. Mathematical conversion of axes is also possible.

Alternatively, you can move the axis assignments back and forth between the channel buttons and the diagram using drag and drop.

Display Coordinates

When you activate this function, the status line shows the current coordinates of the mouse pointer, as long as it is over a diagram. The coordinate display remains active until you deactivate this menu point by selecting it again, or one of the evaluations Set Marker, Draw Mean Value, Fit Function, Calculate Integral or one of the Other Evaluations shows a result in the status line.

You can also insert the current coordinates in the diagram. However, make sure that you access the evaluation function Text via the keyboard with Alt+T without changing the coordinates of the mouse pointer, as otherwise the wrong coordinates would be adopted.

You can save the current setting as the default for subsequent program starts in General.

Shortcut

Keyboard: Alt + C

Select Line Width

You can modify the line width for display of the diagrams and the evaluations which you carry out in them. You can choose between thin, medium and thick lines.

You can save the current setting as the default for subsequent program starts in General.

Experiment # M1

Static and Transient Performance

1. Objectives

In this experiment, the students will learn the basic operation of a DC motor speed controlled system and will also learn the static and transient behavior of the dc motor. Following are the objectives of the experiment:

- Testing the ***linearity*** of the DC motor model.
- Determination of the ***Static Characteristics*** of the DC motor.
- Evaluation of the ***step response*** of the DC motor.
- Determination of the linear model (***transfer function***) of the DC motor.

2. Theory and Background

2.1 General

A common ***actuator*** in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in the Figure M1.1.

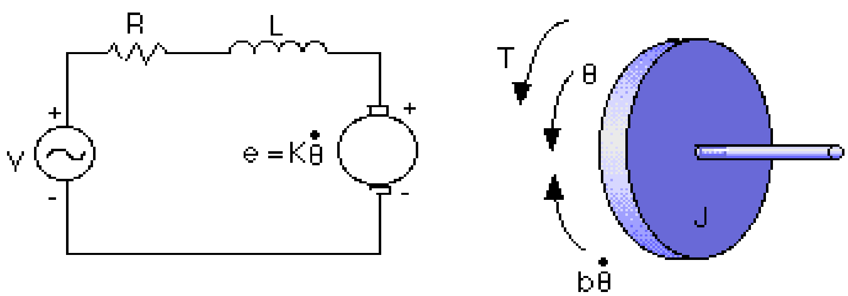


Figure M1.1: Motor Model

2.2 Mathematical model of DC motor

The motor torque, **T**, is related to the armature current, **i**, by the torque constant K_t. The back emf, **e**, is related to the rotational velocity $\dot{\theta}$ by the back emf constant K_b. Thus:

$$T = K_t \, i$$

$$e = K_b \, \dot{\theta}$$

From the Figure M1.1 we can write the following equations based on Newton's law combined with Kirchhoff's law (the rotor and shaft are assumed to be rigid):

$$\mathbf{J\ddot{\theta} + b\dot{\theta} = K_t\, i}$$
$$\mathbf{L\frac{di}{dt} + Ri = V - K_b\,\dot{\theta}}$$
$$\boldsymbol{\omega} = \dot{\boldsymbol{\theta}} = \frac{\mathbf{d}\boldsymbol{\theta}}{\mathbf{dt}}$$

Where

ω = Angular speed of the rotating shaft.
J = moment of inertia of the rotor.
b = damping ratio of the mechanical system.
R = electric resistance of motor winding.
L = electric inductance of motor winding.
V = DC Source Voltage (input).
θ = position of shaft (output).

2.3 Transfer Function

Using Laplace Transform, the above model equations can be expressed in terms of **s**.

$$\mathbf{(Js + b)\,\omega(s) = K_t\; I(s)}$$
$$\mathbf{(Ls + R)\,I(s) = V(s) - K_b\;\omega(s)}$$

By eliminating **I(s)** we can get the following open-loop transfer function, where the rotational ***speed*** is the output and the ***voltage*** is the input.

$$\mathbf{G(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{(Js + b)(Ls + R) + K_t K_b}}$$

$$\text{As}\quad \tau_m = \frac{\mathbf{J}}{\mathbf{b}} >> \tau_e = \frac{\mathbf{L}}{\mathbf{R}}$$

$$\tau_m = \text{mechenical time constant}$$
$$\tau_e = \text{electrical time constant}$$

So we can obtain the ***first order linear Transfer Function*** of the DC motor from the above equation**:**

$$\mathbf{G(s) = \frac{K}{1 + \tau s}}$$

where $\mathbf{K} = \dfrac{\boldsymbol{\Delta\omega(\infty)}}{\boldsymbol{\Delta V(\infty)}} = \dfrac{\text{Steady State change in output}}{\text{Steady State change in input}}$

$\omega(\infty) = y(\infty)$ is the final (steady state) value of the output
$V(\infty) = u(\infty)$ is the input to the system

τ is the time constant of the system and is described by Figure M1.2

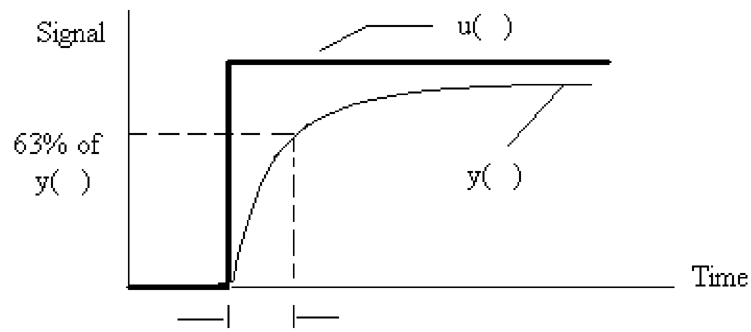


Figure M1.2 Transient response for a Step Input.

3. Experimental Setup

3.1 Components and Appliances

1	Stabilized power supply units	726	86
1	Reference variable generator	734	02
1	Power amplifier	734	13
1	Motor-Generator Set	734	11
1	Load Switch	734	39
1	CASSY-interface with Computer		

3.2 Block Diagram

Figure M1.3 shows the block diagram of the DC Motor Speed System, while Figure M1.4 represents the transfer function of the system.

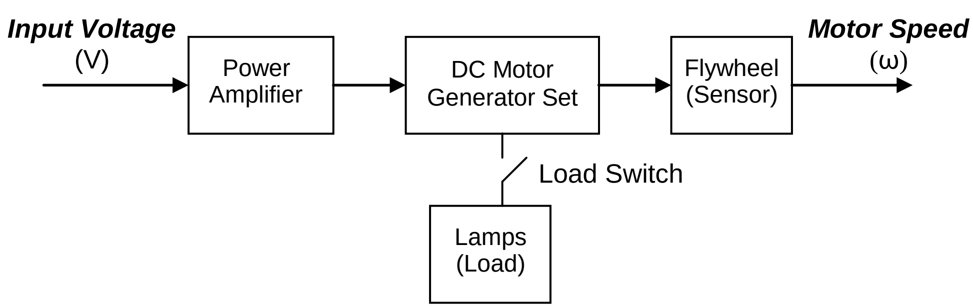


Figure M1.3: DC Motor Speed System

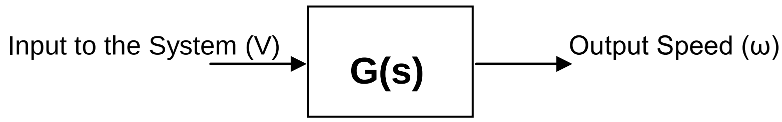


Figure M1.4: System Transfer Function

3.3 Component Description

3.3.1 Motor-Generator Set

The Motor-Generator set (Figure M1.5) comprises two identical 20V DC permanent-field machines which are coupled with Flywheel. The machines have been designed for 20V / 0.47A. The Generator produces an output of **0 …3 V** corresponding to a speed value **n = 0 … 3000 min^{-1}**

3.3.2 Flywheel (Sensor)

The Flywheel has 60 light/dark sections which run past a photo-electric barrier. A digital signal (*revolutions per minute for a counter base T=1 s*) can be tapped at the digital output. A corresponding analog signal can be tapped at the Analog Output via a D/A converter.

3.3.3 Power Amplifier

Power to the Motor is supplied by Power Amplifier. Also, input to the system is always given to the Power Amplifier.

3.3.4 Load Switch

The load switch is provided to connect lamps to the generator and therefore apply load to the DC motor. It can be controlled either manually or automatically.

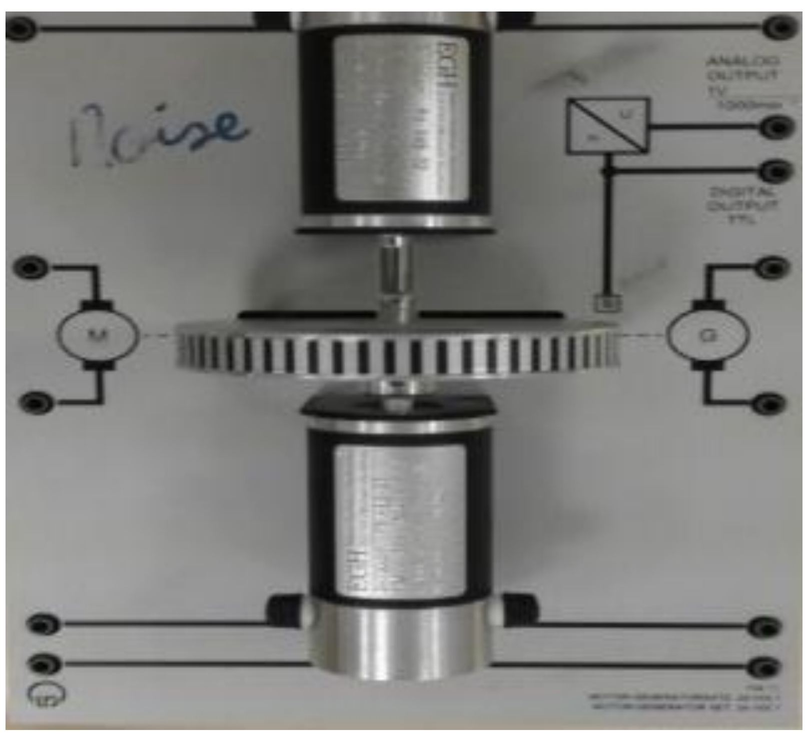


Figure M1.5: The Motor-Generator Set

4. Experimentation Instructions

4.1 Static Performance

- Connect the *"DC Motor Speed System Block Diagram"* (Figure M1.3) to the Power Supply unit and with the "Reference Variable generator".

- Set the ***input voltage*** to 0 V and turn the Power Supply ON.
- Change the input voltage from **0** to **10 V**.
- Make a **Table** and record the *motor speed* against *input voltage*.

Evaluation

- Plot the ***motor speed*** versus ***input voltage****.*
- From the graph, check the ***linearity***.
- From the graph, determine the ***slope*** to find the maximum gain.

4.2 Transient Performance (Open Loop)

- Use *CASSY* Data Acquisition to plot the response.
- ***Step*** Input to the system = **10V** (Use *ON-OFF* switch for the step input).

- At the same time, pull down the switch of the *reference generator* and start CASSY,**F9** Key, (both actions should be done at the same time).

- Plot the response and save your plot on a disk.
- Clearly label the plot and axis.

Evaluation

- From the transient responses, deduce the ***transfer function*** of the Motor. process (identify the parameters **K** and τ from the obtained response).
- How can you tell if the DC Motor Speed is a ***first order*** or a ***second order*** system?

Experiment # M2

Closed-Loop P and PI Control

1. Objectives

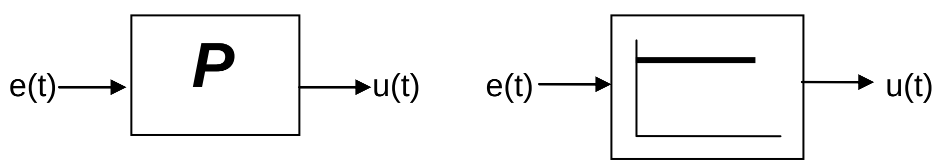
The main targets of the previous experiments were to have an understanding of modeling concepts in terms of *static* and *transient* performance. The DC motor speed process was used in its *open-loop* form. This experiment will introduce the idea of ***closed-loop*** control using ***continuous*** automatic control. Following are the objectives of the experiment:

- Study the effect of ***P*** controller on the DC motor speed.
- Study the effect of ***PI*** controller on the DC motor speed.
- Maintaining DC motor speed close to a *reference* (set point) value.

2. Theory and Background

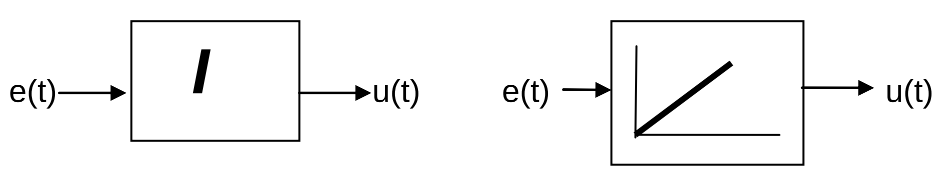
There are many controllers that could be used to perform the closed-loop task. They differ in simplicity, configuration, etc.

The Proportional controller is a "linear and proportional" amplifying element. It produces a manipulated variable u(t) which is proportional to the error signal e(t).



P controller equation: $u(t) = K_P \, e(t)$

The Integral controller produces a manipulated variable u(t) which is proportional to the *integral* of the error signal e(t).



I controller equation: $u(t) = \frac{1}{T_i} \int e(t) \, dt$

The *equation* of the PI controller is given by:

$$u(t) = K_P\, e(t) + \frac{1}{T_i}\int e(t)\, dt$$

where

$$u(t) = \text{Output of PI controller (volts)}$$
$$e(t) = \text{Error signal (volts)}$$
$$T_i = \text{Integral Time}$$
$$K_P = \text{(Dimensionless)}\ \textit{Proportional gain}$$
$$K_i = \frac{1}{T_i}\ (\sec^{-1})\quad \textit{Integral gain}$$

The ***transfer function*** of the PI controller is given by:

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_i}{s} = \frac{K_P s + K_i}{s}$$

where K_P, and K_i are called proportional *gain* and integral *gain* respectively. These parameters (K_P and K_i) are to be selected (tuned) in order to obtain "good" response. Obviously, this choice depends on the system to be controlled.

A *good* response for a *stable* process should have the following characteristics:

1) Good tracking of a reference input.
2) Zero or small steady state error.
3) Fast response.
4) No or little oscillation.

Figure M2.1 shows a block diagram of a closed loop PI controller:

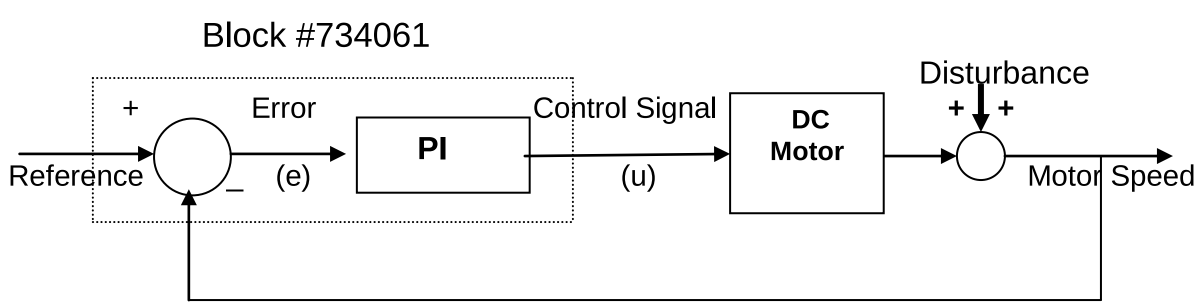


Figure M2.1: Block diagram of a closed loop PI controller

Closed-loop Transfer Function of P-control

Figure M2.2 shows block diagram of a closed-loop ***P-controller:***

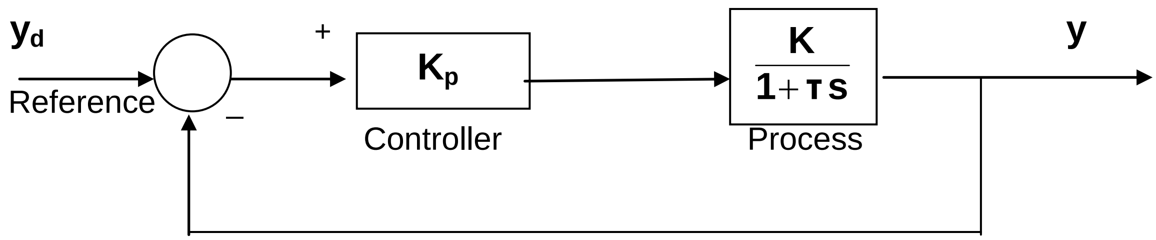


Figure M2.2: Closed loop P-controller

$$\frac{y(s)}{y_d(s)} = \frac{K_p \frac{K}{1+Ts}}{1+K_p \frac{K}{1+\tau s}} = \frac{K_p K}{1+Ts+K_p K}$$

$$= \frac{K_p K \,/(1+KpK)}{1+\left(\frac{T}{1+K_p K}\right)s} = \frac{K_c}{1+T_c s}$$

T_c = Closed– loop Time Constant

K_c = Closed– loop gain

K, T = Process Parameters

Remarks:

1. As $K_p >>$, $K_p \longrightarrow 1$ and $y \longrightarrow y_d$, the steady state error decreases.
2. The closed-loop is faster than open-loop $T_c < T$
3. No overshoot because the closed-loop is 1st order.

3. Experimentation Instructions

- Arrange the experiment according to the block diagram shown in Figure M2.1. Such connection should be obvious by now!
- Set the ***reference*** *input* = 2 V
- Perform the following *tasks* and record the response:

a) ***P-****control*: [Switch OFF K_i]
- Increase K_P in steps of **2, 5, 10,** and **50.**
- Plot the <u>*four*</u> responses and put them on the <u>*same*</u> graph.

b) ***PI-****control*: [Fix K_P = **2**]
- Increase K_i in steps of **0.02, 0.3,** and **1.**
- Plot the above <u>*three*</u> responses and put them on the <u>*same*</u> graph.
- Try and observe the response for K_P = **50** and **Ki** = **1**

c) <u>***Disturbance*** *response of the Motor Speed* ***(PI)****:*</u>
- Set K_P **= 5** and K_i **= 0.03**
- Plot a response until it reaches <u>*steady state*</u> then apply a *Load Disturbance* on the system and <u>*wait*</u> until it reaches steady state again.
- Observe the *behavior* of this response!

4. Evaluation

- Clearly label all the plots.
- VERY CAREFULLY examine the above responses.
- Analyze and discuss the effects of <u>*each*</u> response of ***P*** and ***PI*** control and choose the <u>*best*</u> one.
- Take into consideration (among other things) ***speed, steady state error, overshoot, oscillation,*** **etc.**
- Discuss the effect of **Disturbance**.
- Deduce the “Closed-loop Transfer Function” for ***PI-control*** and explain ***why*** we get oscillation *or* overshoot in the response *and* under what conditions?

Experiment # M3

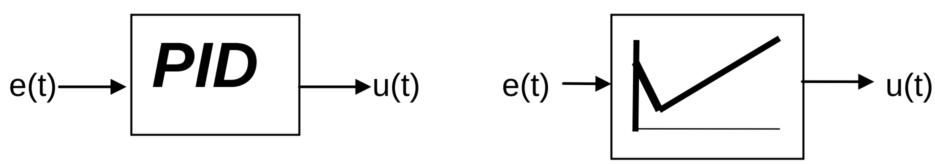
PID Design for the Speed Control System

1. Introduction

By now, you should have acquired the fundamentals of the Speed Control System and the basics of open-loop and closed-loop controls. In the previous experiment, ***PI*** control of the motor speed was examined. In this experiment, a third controller term called ***D*** (derivative) is added.

The primary objective of this experiment is to design a "good" PID controller that maintains the motor speed at a reference value despite of *disturbance*. The design is going to be performed using trial and error along your knowledge of how ***each*** gain is expected to affect the performance of the system.

The PID controller produces a control action u(t) as a function of the error e(t) given by:



The *equation* of the PID controller is given by:

$$u(t) = K_P\, e(t) + \frac{1}{T_i}\int e(t)\, dt + T_v \frac{de(t)}{dt}$$

where

$$K_D = T_v \quad (\text{sec}) \qquad \textit{Derivative gain}$$

The ***transfer function*** of the PID controller is given by:

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_i}{s} + K_D s = \frac{K_P s + K_i + K_D s^2}{s}$$

Figure M3.1 shows a block diagram of a closed loop PID controller:

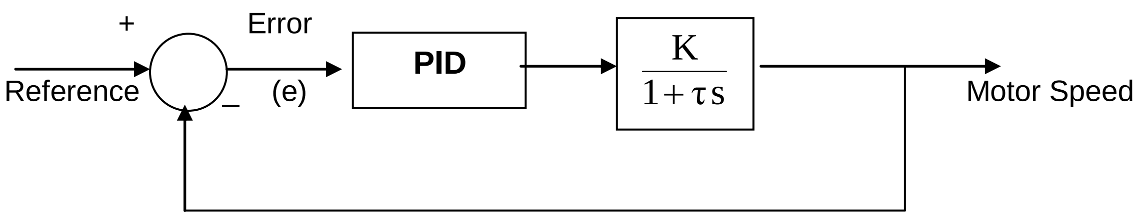


Figure M3.1: Block diagram of a closed loop PID controller

3. Experiment

- Arrange the experiment according to the block diagram shown in Figure M31. <u>*Such connection should be obvious by now!*</u>
- Set the ***reference*** *input* = 2 V
- Perform the following *tasks* and record the responses:

a) <u>***PD-****control*</u>:
- Plot the response at **K_P = 5.**
- Fix **K_P = 5** and <u>*select*</u> **K_D** to obtain ***fast*** response with **no** overshoot.
- Put the ***both*** responses on the <u>*same*</u> graph and ***compare***.

b) <u>Use ***MATLAB Simulink*** toolbox to build the model in Figure M3.1.</u>
- Change the parameters **K_P**, **K_i**, and **K_D** to get the <u>*best*</u> performance.
- Plot the response on *scope.*
- Save all your trials.

Your design objectives should be:
- Good tracking of the reference input.
- Fast response
- No steady state error
- No oscillations

c) <u>***PID-****control*</u>:
- Apply the <u>*best*</u> choice of **K_P**, **K_i**, and **K_D** of *part(b)* to provide the “best” possible performance for the motor speed.
- Save the obtained results.

5. Evaluation

- Discuss the effect of **D** controller.
- Discuss the ***steps*** that lead to your <u>final</u> **PID** design.
- From your response, calculate the ***steady state error***, the ***settling time***, and the ***time constant*** of the system.

Experiment # T1

Static and Transient Performance

1. Objectives

In this experiment, the students will learn the basic operation of a temperature controlled system and will also learn the static and transient behavior of the temperature process. Following are the objectives of the experiment:

- Study the effect of ***flap, ventilator-motor****,* and ***input power*** over the
- temperature output.
- Evaluation of the “step response” of the temperature process.
- Determination of the linear model (transfer function) of the temperature process.

2. Theory and Background

Temperature systems are widely used in real life whether in industrial applications or at home. The underlying concepts of operation is basically the same.

2.1 System Description

Figure T1.1 shows the block diagram and the system description of the temperature process.

- The temperature controlled system as an “oven” contains a halogen lamp (1) 12 V and is equipped with a heat sink.
- The oven temperature is measured with the sensor (2) and converted via transducers in either 2 mA/10oC or 1 V/10oC (8).
- The operation mode is selected using the switch (8). The output is (9) and the controlled variable is T.
- The entire arrangement consists of a ventilator-motor (fan) (3) and an adjustable flap (openning) (5) arranged in a transparent channel (6). The power of the ventilator is set with a potentiometer (4).
- The heating power (7) is supplied by the power amplifier via a diode, so that It is guaranteed that the control loop is operated only in the first quadrant (+V; +I) and that positive feedback is not transformed into negative feedback in the controller.

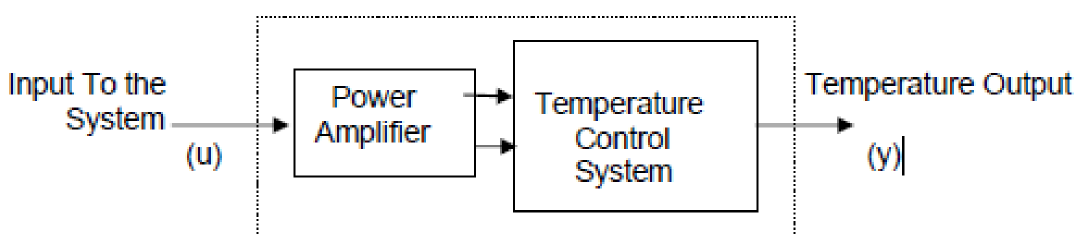


Figure T1.1: System Description and Block Diagram of Temperature Control

- ***Static test***

Static test is performed to check the *linearity* of the equipment under test. If we want to test the "flap" we need to keep the ventilator-motor, and input power at constant values and vary the flap scale.

- ***A first order linear description of the system can be given by***

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

$$\text{where} \quad K = \frac{\Delta y(\infty)}{\Delta u(\infty)} = \frac{\text{Steady State change in output}}{\text{Steady State change in input}}$$

y(∞) = final (steady state) value of the output
u(∞) = input to the system
T = the time for which the output reaches 63% of its steady state value and it is called the time constant.
L = pure time delay (dead time). The time required for the system to start responding to the input change.

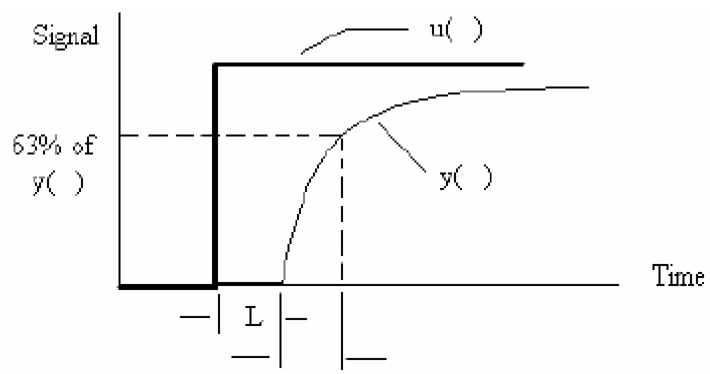


Figure T2.2 Representation of transient response

3. Components and appliances

1 Reference variable generator 73402
1 Power amplifier 73413
1 Temperature controlled system 73412
1 Power Supply +/- 15 V 72686
1 CASSY-interface with Computer

4. Experimentation Instructions

4.1 *Static Performance*

- Connect the "Temperature Control System Block Diagram" (Figure T1.1) to the Power Supply unit and with the "Reference Variable generator".
- Set the output switch to 1V/10oC.

- Make sure that the temperature of the process settles at room temperature.

1. Effect of "flap" over temperature

- Set the *refererence variable generator* to 8V input.
- Set the *ventilator-motor potentiometer* at scale 3.
- Switch the lamp ON.
- Make a Table and record the oven temperature at *flap* scalesm4, 3, 2, 1 and 0.

2. Effect of "ventilator-motor" over temperature

- Keep the *input power* to 8V.
- Set the *flap* to 2 scale divisions.
- Make a Table and record the oven temperature at *ventilator-motor potentiometer* scales 8, 6, 4, 3 and 2.

3. Effect of "input power" over temperature

- Keep the *flap* at scale 2 and *ventilator potentiometer* at scale 3.
- Make a Table and record the temperature of the system at inputs 0, 2, 4, 6, and 8 Volts.

4.2 *Transient Performance*

- Use CASSY Data Acquisition to plot the response.
- Cool down the system at *room temperature*.
- Set the ***flap*** at scale 2 and ***ventilator potentiometer*** at scale 3.
- ***Step*** Input to the system = 10V
- At the same time, pull down the switch of the *reference generator* and start the CASSY, **F9** Key (both actions should be done at the same time).
- Record the transient response of the process.
- Print and save your plot on *disk*.

5. Evaluation

- Plot ***temperature*** *versus* ***flap***
- Plot ***temperature*** *versus* ***ventilator-motor***
- Plot ***temperature*** *versus* ***input power***
- Label your scales very carefully.
- From the transient responses, deduce the ***transfer function*** of the temperature process (identify the parameters **K, T** and **L** from the obtained response).

Experiment #T2

Closed-Loop Two-Position Control

1. Objectives

The main targets of the previous experiments were to have an understanding of modeling concepts in terms of *static* and *transient* performance. The temperature process was used in its *open-loop* form. This experiment will introduce the idea of *closed-loop* control. Following are the objectives of the experiment:

- Study the *two-position* (discontinuous) controller.
- Examine the effect of *hysteresis* on the temperature process.
- Maintaining process temperature close to a *reference* (set point) value.

2. Theory and Background

There are many controllers that could be used to perform the closed-loop task. They differ in simplicity, configuration, etc. ***The two-position controller*** recognizes only two states and has only two actions: ON or OFF.

In two-position controllers, there is an *extensive and continuous* switching back and forth between the two states. This continuous switching may damage the equipment. To reduce the switching rate we can equip the two-position controller with **hysteresis**. ***The Hysteresis*** is a differential *gap* between the two states (ON-OFF) of the two position controller. The lamp will switch ON or OFF when the output reaches the two given levels of the hysteresis.

The *switching rate* of two-position controller can be influenced directly using *hysteresis*. Thus choosing the *right* value for the hysteresis varies from problem to problem; here the *type* of controlled system plays an important role. Figures T3.1 and T3.2 show block diagrams of *open-loop* and *closed-loop* configurations. The fundamental differences between the two cases need to be understood before any appreciation of control systems is achieved.

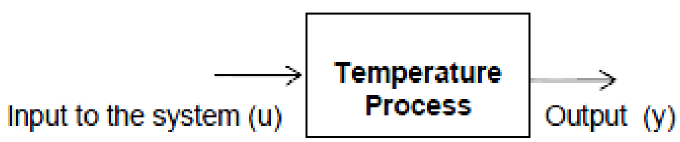


Figure T3.1: Open-loop block diagram

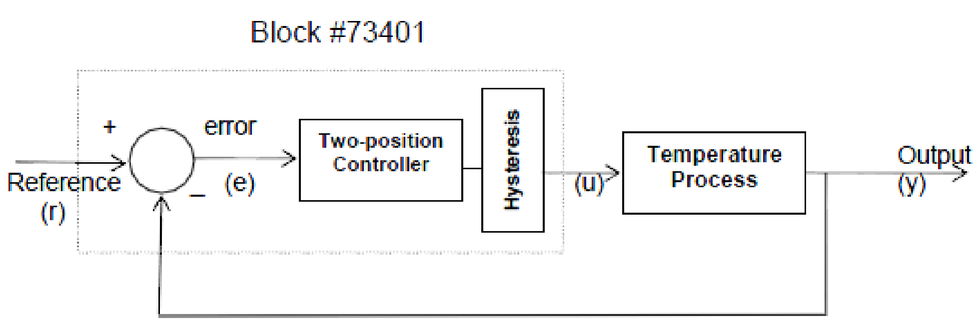


Figure T3.2: Block diagram of a closed loop two-position controller.

By examining the block diagram, we can easily deduce that:

$$e(t) = r(t) - y(t)$$

If there is no switching hysteresis (h = 0), the two states of the controller are:

$$\begin{cases} e > 0 & \Rightarrow u : ON \\ e < 0 & \Rightarrow u : OFF \end{cases}$$

However, if there is switching hysteresis (±h), the two states are:

$$\begin{cases} e > +h \Rightarrow & u : ON \\ e < -h \Rightarrow & u : OFF \\ |e| < h \Rightarrow & u : remains\ unchanged\ either\ ON\ or\ OFF \end{cases}$$

3. Experimentation Instructions

3.1 Closed-Loop Two-Position Control

- Set up the experimental arrangement as shown in the block diagram of
- Figure T3.2.
- Connect the *reference generator* to the two-position controller *Block* (#73401) and then connect to the temperature process.
- *Note:* You should be capable of making the connection without the help of a circuit diagram by now!
- Set the *reference input* to 5 V.
- Set the *ventilator-motor potentiometer* at scale 3.
- Set the *flap* to 2 scale divisions.
- Plot Output **(y)** *with* the Output of two-position controller **(u)** for the following values of **hysteresis:** 1) h = **0** V 2) h = ± **0.5** V 3) h = ± **1.5** V
- Observe the behavior of the lamp!

Note: Make sure to record data using CASSY for at least *one* or *two* cycles.

3.2 Open-Loop

- Keep the *same* settings as before (switch off).
- Disconnect the feedback path from the output to the input of the controller.
- Set the switching **hysteresis** to 0 V and plot the *reference* and the *output* on the same graph.

4. Evaluation

- All the plots *must* be Labeled.
- Discuss the difference *between* the ***open-loop*** and the closed-loop ***two-position controller***.
- Discuss the *output of the controller* and the *behavior of the lamp*.
- Analyze *all* the responses and choose the *best* one.
- Explain any interesting observations.

Experiment #T3

Proportional Control

1. Objectives

The previous experiment examined the two-position (ON/OFF) controller which is a *discontinuous* controller. In this experiment *continuous* controller is examined. It is called the proportional (P) controller. Following are the objectives of the experiment:

- Study the effect of ***P***-controller on the temperature output.
- Examine the difference between "open-loop" and "closed-loop ***P***- controller".
- Study the effect of ***Disturbance*** on the system response.

2. Theory and Background

The P controllers are used for simple process control operations. They are easy to design electronically as analog controller or even digital. P controller produces manipulated variable u(t) which is proportional to the error signal e(t). Figure T3.1 shows a block diagram of such control technique.

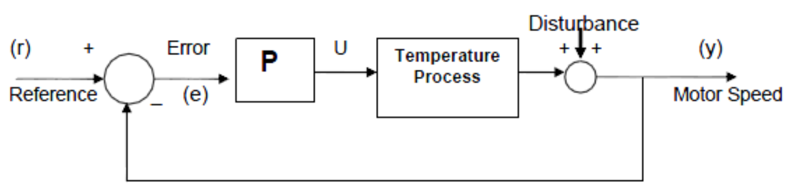


Figure T3.1: Block diagram of Proportional Control

Assuming that the temperature process has a transfer function G(s) =

$$G(s) = \frac{K_1}{1+\tau s}$$

The closed loop transfer function is then given by:

$$\frac{KG(s)}{1+KG(s)} = \frac{KK_1}{\tau s + (KK_1 + 1)}$$

The pole of the open-loop system is at s = -1/ τ while the pole of the closed-loop is at

$$s = -\frac{1+KK_1}{\tau}$$

Note that when K = 0 both poles are the same.

K is called "proportional" gain and it can be chosen to put the poles of the closed loop system at different locations. Therefore, we can choose a value of K that may improve the performance.

3. Experimentation Instructions

Set up the experiment according to the block diagram shown in Figure T3.1.

- Set the *reference input* to **5 V**.
- Set the *ventilator-motor potentiometer* at scale **3**.
- Set the *flap* to **2** scale divisions.
- Turn the switch ON and plot the *reference* and the *output* on the same graph.
- *Remember to* ***always*** *let the system cool down.*

3.1 P-control

- Plot four responses for a **closed-loop** proportional gain: **K_P = 5, 20, 40 and 70**
- ***Open-loop:*** Repeat above for **K_P = 70** but disconnect the feedback.
- Put all the *five* responses on the *same* graph.

3.2 Disturbance Rejection

- Set **K_P=20.** Plot a response until it reaches *steady state* then *suddenly* change the speed of ***ventilator motor*** and *wait* until it reaches steady state again.
- Observe the *behavior* of this response!
- Repeat the *Disturbance* for the ***Flap*** position.

4. Evaluation

- Comment, analyze and discuss the effects of *each* response and choose the ***best*** one.
- Evaluate the performance in terms of ***speed, steady state error, overshoot, oscillation*** **etc.**
- Discuss the effect of **Disturbance** on the system behavior.

EXPERIMENT # S1

CRUISE CONTROL SIMULATION

1. Objective

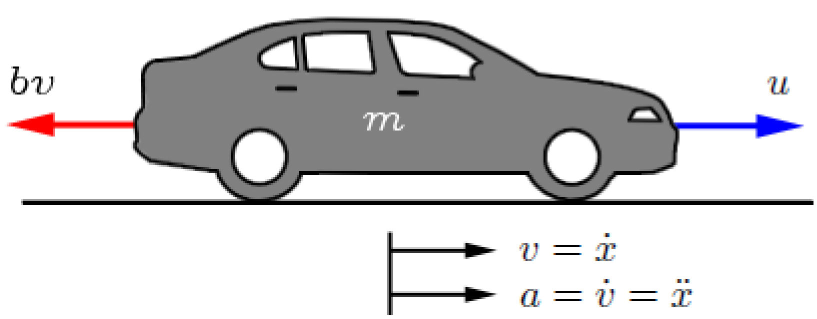
In this experiment, the students will perform MATLAB simulation in order to analyze the performance of cruise control system in open loop as well as in closed-loop by using classical PID control. Following are the objectives of this experiment:

- Determination of the linear model (transfer function) of the cruise control system
- Open loop analysis of cruise control system in MATLAB
- PID controller design and closed-loop analysis of cruise control system in MATLAB in order to achieve desired control objectives

2. System Description

Automatic cruise control is an excellent example of a feedback control system found in many modern vehicles. The purpose of the cruise control system is to maintain a constant vehicle speed despite external disturbances, such as changes in wind or road grade. This is accomplished by measuring the vehicle speed, comparing it to the desired or reference speed, and automatically adjusting the throttle according to a control law.

3. System Modeling



We consider here a simple model of the vehicle dynamics, shown in the free-body diagram (FBD) above. The vehicle, of **mass m**, is acted on by a **control force, u**. The force **u** represents the force generated at the road/tire interface. For this simplified model, we will assume that we can control this force directly and will neglect the dynamics of the powertrain, tires, etc., that go into generating the force. The **resistive forces, bv**, due to rolling resistance and wind drag, are assumed to vary linearly with the **vehicle velocity, v**, and act in the direction opposite the vehicle's motion. **b** is **damping coefficient**.

With these assumptions we are left with a first-order mass-damper system. Summing forces in the x-direction and applying Newton's 2nd law, we arrive at the following system equation:

$$\mathrm{m}\,\dot{\mathrm{v}} + \mathrm{b}\,\mathrm{v} = \mathrm{u}$$

we are interested in controlling the speed of the vehicle so the output is **v** which is speed of vehicle and input is control force which is **u**. Taking the Laplace transform of the governing differential equation and assuming zero initial conditions, we will find the transfer function of the cruise control system as:

$$\mathrm{m\,s\,V(s) + b\,V(s) = U\,(s)}$$

$$\mathrm{(m\,s + b)\,V(s) = U(s)}$$

$$G(s) = \frac{V(s)}{U(s)} = \frac{1}{ms + b}$$

The parameters used in this example are as follows:

m (vehicle mass) = 1000 kg
b (damping coefficient) = 50 N.s/m
u(nominal control force) = 500 N

4. PID Controller

The PID controller produces a **control action u(t)** as a function of the **error e(t)** given by:

$$u(t) = K_P\, e(t) + \frac{1}{T_i}\int e(t)dt + T_v \frac{d\, e(t)}{dt}$$

The transfer function of PID controller is given by:

$$C(s) = \frac{U(s)}{E(s)} = K_P + K_D\, s + \frac{K_I}{s} = \frac{K_D\, s^2 + K_P\, s + K_I}{s}$$

5. Performance Specifications

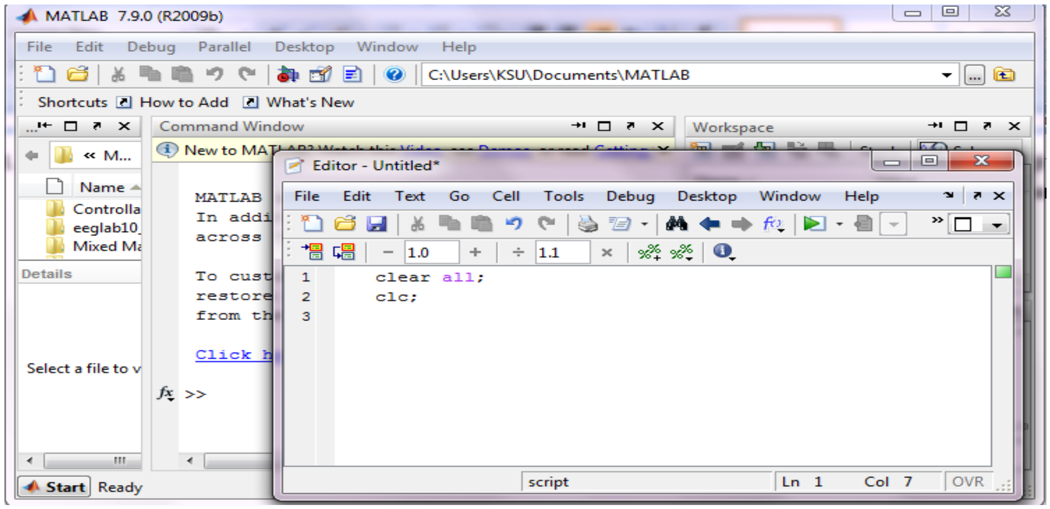
The next step is to come up with some design criteria that the compensated system should achieve. When the engine gives a 500 Newton force (u), the car will reach a maximum velocity (v) of 10 m/s (it needs to be verified from open loop step response of the system). An automobile should be able to accelerate up to that speed (10 m/s) in less than 5 seconds. In this application, a 10% overshoot and 2% steady-state error on the velocity are sufficient.

Keeping the above in mind, we have proposed the following design criteria for this problem:

- ✓ Rise time < 5 s
- ✓ Overshoot < 10%
- ✓ Steady-state error < 2%

6. Simulation Instructions

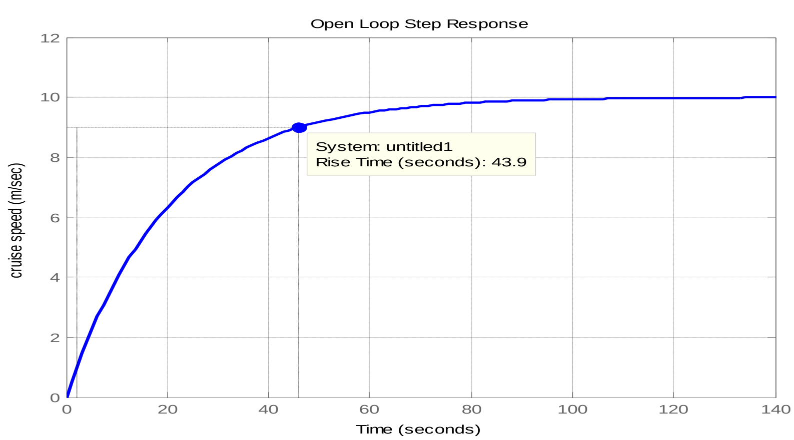
1. Start MATLAB software
2. Create new M-file to write the code (Click *File* > *New* > *Blank M File*)



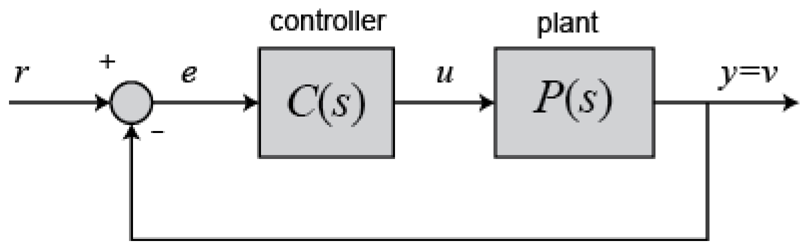
3. To simulate the open loop response, write the following code in the M-File :

```
clear all; % Clear all variables
clc; % clear screen

m = 1000; % m (vehicle mass) =  1000 kg
b = 50;    % b (damping coefficient) =  50 N.s/m
num = [1];
den = [m b];
TF_cruise = tf(num, den); % Transfer function: G(s)=1/(ms+b)

u=500; % u (nominal control force) =  500 N
step(500* TF_cruise); % Plot the open loop step response
grid on;
ylabel('cruise speed (m/sec)');
title('Open Loop Step Response');
```

4. Save and run your code ! See the open loop response.



5. For closed loop system, we should add a controller to the system :



Modify your previous code with the following code :

```
Kd=0;Kp=10;Ki=1; % Controller gains
C = tf([Kd Kp Ki],[1 0]); % TF of the controller
                          % C(s)= (KD s^2  +  KP  s + KI)/s
T = feedback(C*TF_cruise,1); % TF of the closed loop system
r=10; % Reference speed of 10 m/s
step(r*T);  % Step response of the closed loop system
```

6. Change the values of proportional, integral and differential gains (*Kp, Ki, Kd*) until the desired performance objectives are achieved.

7. Evaluation

- Find out step response of the open loop cruise control system when the engine gives an input force of 500 N. Comment on your plot with respect to desired performance objectives as mentioned in section 5.

- Find out step response of the closed loop cruise control system in the presence of Proportional (P) controller only. Change the value of proportional gain and comment on your achieved plots accordingly.

- Find out step response of the closed loop cruise control system in the presence of Proportional – Integral (PI) controller first and then PID controller. Change the values of Proportional, Integral and Differential gains and comment on your achieved plots accordingly w.r.t. desired performance objectives.

EXPERIMENT # S2

DC MOTOR SPEED CONTROL

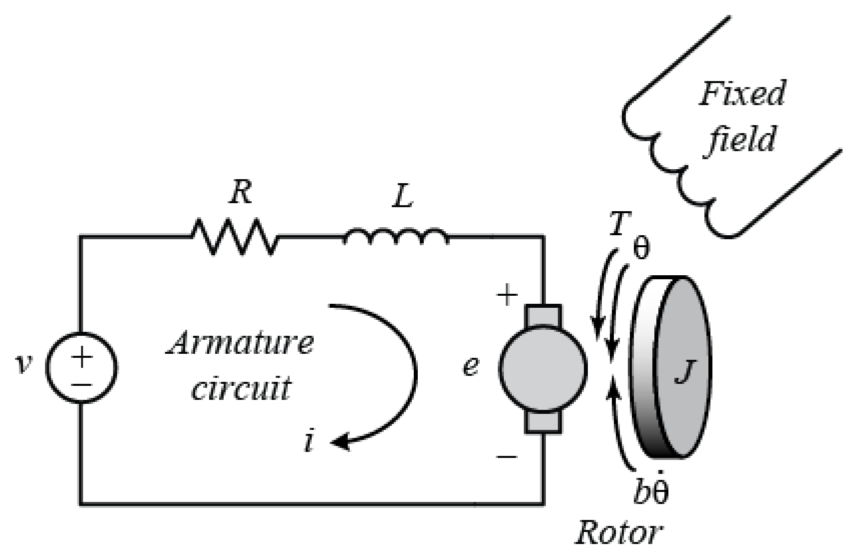
1. Objective

In this experiment, the students will perform MATLAB simulation for analyzing the performance of speed control of a DC motor in open loop as well as in closed-loop by using classical PID control. Following are the objectives of this experiment:

- Determination of the linear model (transfer function) of a DC motor
- Open loop analysis of speed control of a DC motor in MATLAB
- PID controller design and closed-loop analysis of speed control of a DC motor in MATLAB in order to achieve desired control objectives

2. System Description

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in the following figure.



For this example, we will assume that the input of the system is the **voltage source (V)** applied to the motor's armature, while the output is the rotational speed of the shaft ($\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}$). The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.

The physical parameters for our example are:

(J) moment of inertia of the rotor 0.01 kg.m^2
(b) motor viscous friction constant 0.1 N.m.s
(Ke) electromotive force constant 0.01 V/rad/sec

(Kt)	motor torque constant	0.01 N.m/Amp
(R)	electric resistance	1 Ohm
(L)	electric inductance	0.5 H

3. System Modeling

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. In this example we will assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor **Kt**as shown in the equation below. This is referred to as an armature-controlled motor.

$$T = K_t \cdot i$$

The back emf, **e**, is proportional to the angular velocity ($\boldsymbol{\omega}$) of the shaft by a constant factor Ke.

$$e = K_e \cdot \omega$$

In SI units, the motor torque and back emf constants are equal, that is, **Kt = Ke**; therefore, we will use **K** to represent both the motor torque constant and the back emf constant.

From the figure above, we can derive the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law.

$$J\,\dot{\omega} + b\,\omega = K \cdot i$$

$$L\,\frac{di}{dt} + R\,i = V - K\,\omega$$

By taking Laplace transform of above two equations:

$$J\,s\,\omega(s) + b\,\omega(s) = K \cdot I(s)$$

$$L\,s\,I(s) + R\,I(s) = V(s) - K\,\omega(s)$$

$$(L\,s + R)\,I(s) = V(s) - K\,\omega(s)$$

$$I(s) = \frac{V(s) - K\,\omega(s)}{L\,s + R}$$

Put it in first equation:

$$J\,s\,\omega(s) + b\,\omega(s) = K \cdot \frac{V(s) - K\,\omega(s)}{L\,s + R}$$

$$(J\,s + b)\,\omega(s) = K \cdot \frac{V(s) - K\,\omega(s)}{L\,s + R}$$

$$(J\,s + \;\; b)\;(L\,s + R)\;\omega(s) = K\;(V(s) - K\;\omega(s))$$
$$(J\,s + \;\; b)(L\,s + R)\;\omega(s) = K\;V(s) - K^2\;\omega(s)$$
$$(J\,s + \;\; b)(L\,s + R)\;\omega(s)\; + \;K^2\;\omega(s) = K\;V(s)$$

$$\frac{\omega(s)}{V(s)} \;=\; \frac{K}{(J\,s + \;\; b)(L\,s + R)\;\; + \;K^2}$$

4. Performance Specifications

First consider that our uncompensated motor rotates at 0.1 rad/sec in steady state for an input voltage of 1 Volt (it needs to be verified from open loop step response of the system). Since the most basic requirement of a motor is that it should rotate at the desired speed, we will require that the steady-state error of the motor speed be less than 1%. The desired speed is 1 rad/sec for an input voltage of 1V. Another performance requirement for our motor is that it must accelerate to its steady-state speed as soon as it turns on. In this case, we want it to have a settling time less than 2 seconds. Also, since a speed faster than the reference may damage the equipment, we want to have a step response with overshoot of less than 5%.

In summary, for a unit step command in motor speed, the control system's output should meet the following requirements.

- Desired motor speed is 1 rad/sec for an input voltage of 1V
- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 1%

5. Evaluation

- Find out step response of the open loop DC motor speed control system for an input voltage of 1V. Comment on your plot with respect to desired performance specifications as mentioned in section 4.

- Find out step response of the closed loop DC motor speed control system in the presence of Proportional (P) controller only. Change the value of proportional gain and comment on your achieved plots accordingly.

- Find out step response of the closed loop DC motor speed control system in the presence of Proportional – Integral (PI) controller first and then PID controller. Change the values of Proportional, Integral and Differential gains and comment on your achieved plots accordingly w.r.t. desired performance specifications.

EXPERIMENT # S3

DC MOTOR POSITION CONTROL

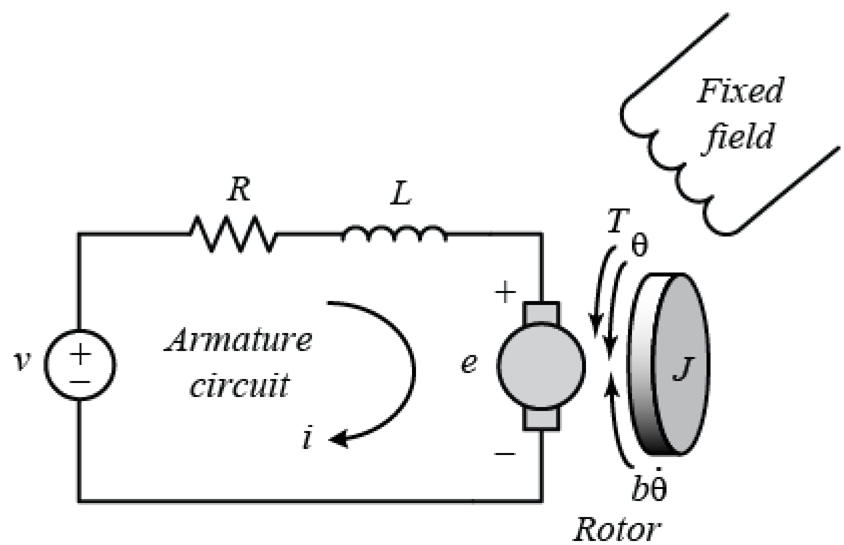
1. Objective

In this experiment, the students will perform MATLAB simulation for analyzing the performance of a DC motor position control system in open loop as well as in closed-loop by using classical PID control. Following are the objectives of this experiment:

- Determination of the linear model (transfer function) of a DC motor position
- Open loop analysis of a DC motor position control system in MATLAB
- PID controller design and closed-loop analysis of a DC motor position control system in MATLAB in order to achieve desired control objectives

2. System Description

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in the following figure.



In this example, we assume that the input of the system is the voltage source (**V**) applied to the motor's armature, while the output is the position of the shaft (**theta**). The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.

For this example, we will assume the following values for the physical parameters. These values were derived by experiment from an actual motor in Carnegie Mellon's undergraduate controls lab.

(J)	moment of inertia of the rotor	3.2284E-6 kg.m^2
(b)	motor viscous friction constant	3.5077E-6 N.m.s
(Kb)	electromotive force constant	0.0274 V/rad/sec
(Kt)	motor torque constant	0.0274 N.m/Amp
(R)	electric resistance	4 Ohm
(L)	electric inductance	2.75E-6H

3. System Modeling

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. In this example we will assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor **Kt**as shown in the equation below. This is referred to as an armature-controlled motor.

$$T = K_t \cdot i$$

The back emf, **e**, is proportional to the angular velocity ($\dot{\theta}$) of the shaft by a constant factor Ke.

$$e = K_e \cdot \dot{\theta}$$

In SI units, the motor torque and back emf constants are equal, that is, **Kt = Ke**; therefore, we will use **K** to represent both the motor torque constant and the back emf constant.

From the figure above, we can derive the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law.

$$J\ddot{\theta} + b\dot{\theta} = K \cdot i$$

$$L\frac{di}{dt} + R\,i = V - K\dot{\theta}$$

By taking Laplace transform of above two equations:

$$J\,s^2\,\theta(s) + b\,s\,\theta(s) = K \cdot I(s)$$

$$L\,s\,I(s) + R\,I(s) = V(s) - K\,s\,\theta(s)$$

$$(L\,s + R)\;I(s) = V(s) - K\,s\,\theta(s)$$

$$I(s) = \frac{V(s) - K\,s\,\theta(s)}{L\,s + R}$$

Put it in first equation:

$$J\,s^2\,\theta(s) + b\,s\,\theta(s) = K \cdot \frac{V(s) - K\,s\,\theta(s)}{L\,s + R}$$

$$s\,(J\,s + \;b)\,\theta(s) = K\;\cdot\;\frac{V(s) - K\,s\,\theta(s)}{L\,s + R}$$
$$s\,(J\,s + \;b)\,(L\,s + R)\,\theta(s) = K\,(V(s) - K\,s\,\theta(s))$$
$$s\,(J\,s + \;b)(L\,s + R)\,\theta(s) = K\,V(s) - K^2\,s\,\theta(s)$$
$$s\,(J\,s + \;b)(L\,s + R)\,\theta(s)\; + \;K^2\,s\,\theta(s) = K\,V(s)$$
$$s\,[(J\,s + \;b)(L\,s + R)\; + \;K^2]\,\theta(s) = K\,V(s)$$

$$\frac{\theta(s)}{V(s)}\; = \;\frac{K}{s\,[(J\,s + \;b)(L\,s + R)\; + \;K^2]}$$

4. Performance Specifications

We will want to be able to position the motor very precisely, thus the steady-state error of the motor position should be zero when given a commanded position. The other performance requirement is that the motor reaches its final position very quickly without excessive overshoot. In this case, we want the system to have a settling time less than 40 ms and an overshoot smaller than 16%.

If we simulate the reference input by a unit step input, then the motor position output should have:

- Settling time less than 40 milliseconds
- Overshoot less than 16%
- No steady-state error

5. Evaluation

- Find out step response of the open loop DC motor position control system for an input voltage of 1V. Comment on your plot with respect to desired performance specifications as mentioned in section 4. Is open loop system stable ?

- Find out step response of the closed loop DC motor position control system in the presence of Proportional (P) controller only. Change the value of proportional gain and comment on your achieved plots accordingly.

- Find out step response of the closed loop DC motor position control system in the presence of Proportional – Integral (PI) controller first and then PID controller. Change the values of Proportional, Integral and Differential gains and comment on your achieved plots accordingly w.r.t. desired performance specifications.