The University of Alabama at Birmingham

School of Engineering

Department of Mechanical Engineering

in Collaboration with

Center for Advanced Automotive Technology

DC Motor Control: Lifting Device



***Prepared By:***

**Vehicle and Robotics Engineering Laboratory**

**[](http://www.uab.edu/engineering/home)Mechanical Engineering Department**

**School of Engineering**

**University of Alabama at Birmingham, USA**

**2017**

**Vladimir V. Vantsevich**

**Professor and VREL Director**

**Jesse R. Paldan**

**Research Assistant**

# Table of contents

* Objective ……………………………………………………………..…………………...3
* Control Systems Overview…………………...…………………………….....…………..3
* PID Control…………………….…..……………………………………………….……..5
* DC Motor Model…………………………………………………………………………..7
* Lifting Device Model……….…………………………………..………………………..10
* Control System Computer Modeling…………………………………………………….14
* References………………………………………………………………………………..19
* Assignment ……………………………………………………………...….……………20

## Objective

1. Learn about control systems and PID controllers
2. Learn the mathematical model of a DC motor
3. Understand a computer model to demonstrate the use of a PID controller to control a motor

### Control Systems Overview

##### Closed Loop Systems

In a typical control system, the process variable is the system parameter that needs to be controlled, such as temperature (°C), pressure (psi), or flow rate (liters/minute). A sensor is used to measure the process variable and provide feedback to the control system. The set point is the desired or command value for the process variable, such as 100 degrees Celsius in the case of a temperature control system. At any given moment, the difference between the process variable and the set point is called the error. The error is used by the control system algorithm (compensator) to determine a control signal. The control signal is used to control the actuator output that drives the system (plant). For instance, if the measured temperature process variable is 100 °C and the desired temperature set point is 120 °C, then the actuator output specified by the control algorithm might be to drive a heater. Driving an actuator to turn on a heater caused the system to become warmer and results in an increase in the temperature process variable. This is called a closed loop control system because the process of reading sensors to provide constant feedback and calculating the desired actuator output is repeated continuously and at a fixed loop rate as illustrated in Fig. 1.

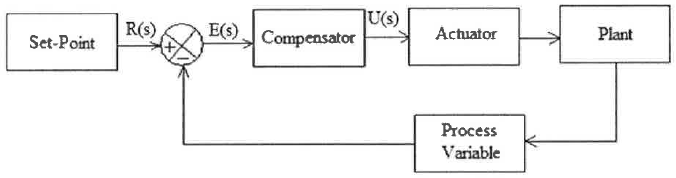


Figure : Block diagram of a closed-loop system

In many cases, the actuator output is not the only signal that influences the system. For instance, in a temperature chamber there might be a source of cool air that sometimes blows into the chamber and disturbs the temperature. Such a term is referred to as a disturbance. A control system is usually designed to minimize the effect of disturbances on the process variable.

##### 1.2 Control System Performance Characteristics

Control system performance is often measured by applying a step function as the set point command variable and then measuring the response of the process variable. Commonly, the response is quantified by measuring defined waveform characteristics such as the following shown in Fig. 2:

**Rise Time:** The rise time is the amount of time the system takes to go from 10% to 90% of the steady-state, or final, value.

**Percent Overshoot:** Percent overshoot is the amount that the process variable overshoots the final value, expressed as a percentage of the final value.

**Settling Time:** Settling time is the time required for the process variable to settle to within a certain percentage (commonly 5%) of the final value.

**Steady-State Error:** Steady-State error is the final difference between the process variable and the set point.

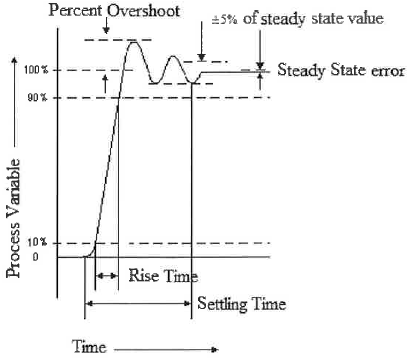


Figure : Response waveform characteristics

### PID Control

##### 2.1 Definition of PID Control

PID control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity. As the name suggests, the PID algorithm consists of three basic coefficients: proportional, integral, and derivative. The values of these coefficients are varied to get optimal response. The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output.

##### Proportional Response

The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the error term. The proportional gain (P) determines the ratio output response to the error signal. For instance, if the error term has a magnitude of 10, a proportional gain of 5 would produce a proportional response of 50. In general, increasing the proportional gain will increase the speed of the control system response. However, if the proportional gain is too large, the process variable will begin to oscillate. If P is increased further, the oscillations will become larger and may even oscillate out of control. This is called instability.

##### Integral Response

The integral component (I) sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the steady-state error to zero. A phenomenon called the integral windup results when integral action saturates a controller without the controller driving the error signal toward zero.

##### Derivative Response

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative (D) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall system response. Most practical control systems use very small derivative time because the derivative response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or the control loop rate is too slow, the derivative response can make the control system unstable. The expression for a PID controller is given by

(1)

where u = control signal

e = error signal

P = proportional Gain

D = differential gain

I = integral gain

##### 2.2. Determining PID Controller Gains

One way of getting the gains of a PID controller is by the trial and error method. In this method, the integral and derivative terms are set to zero first and the proportional gain is increased until the output of the loop oscillates. As one increases the proportional gain, the system becomes faster, but care must be taken not to make the system unstable. Once proportional gain has been set to obtain a desired fast response, the integral term is increased to stop the oscillations. The integral term reduces the steady state error, but increases overshoot. Some amount of overshoot is always necessary for a fast system so that it could respond to changes immediately. The integral term is tweaked to achieve a minimal steady state error. Once the proportional and integral gains have been set to get the desired fast control system with minimal steady state error, the derivative term is increased until the loop is acceptably quick to its set point. Increasing the derivative term decreases overshoot and yields higher gain with stability but would cause the system to be highly sensitive to noise. Often it is necessary to tradeoff one characteristic of a controls system for another to better meet their requirements.

Another way to getting the gains of a PID controller is by reducing the closed loop characteristic equation to a second order equation and comparing the coefficients of this equation with the standard characteristic equation of a second order system which is given by

(2)

where the damping coefficient and the frequency can be calculated based on the required system performance characteristics such as settling time, percent overshoot, and steady state error. This method would take us close to the required PID gains. A slight tuning may still be needed to meet the system performance characteristics.

### DC Motor Model

##### 3.1 Permanent Magnet DC Motors

Permanent Magnet (PM) motors are the most commonly used DC motors. A Permanent Magnet DC Motor converts direct current electrical energy into mechanical energy through the interaction of two magnetic fields. One field is produced by a permanent magnet assembly; the other field is produced by an electrical current flowing in the rotor windings. These two fields result in a torque which tends to rotate the rotor. As the rotor turns, the current in the winding is commutated to produce a continuous torque output. Figure 3 below shows a three pole PM DC Motor.

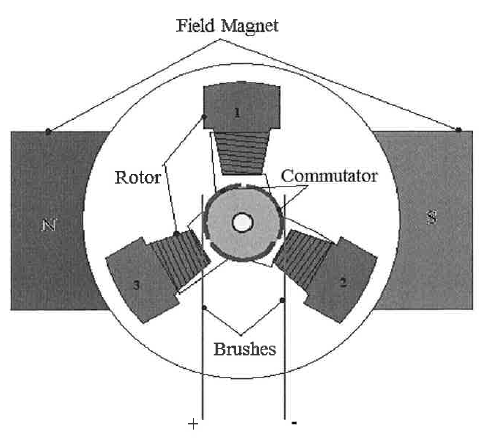


Figure : Permanent Magnet DC Motor

The commutator acts as half of a mechanical switch and rotates with the rotor or armature as it turns. The second half of the mechanical switch is completed by the brushes. These brushes typically remain stationary with the motor’s housing but ride (or brush) on the rotating commutator. As electrical energy is passed through the brushes and consequently through the armature a torsional force is generated as a reaction between the PM field and the armature causing the motor’s armature to turn. As the armature turns, the brushes switch to adjacent bars on the commutator. This switching action transfers the electrical energy to an adjacent winding on the armature which in turn perpetuates the torsional motion of the armature.

The operation of a PM DC Motor is based on the law of electromagnetism where a current-carrying conductor generates a magnetic field. When it is placed in an external magnetic field, it will experience a force proportional to the current in the conductor at the strength of the external magnetic field. The direction of the magnetic field around the current carrying conductor can be found out using the right hand rule as shown in Fig. 4 below.

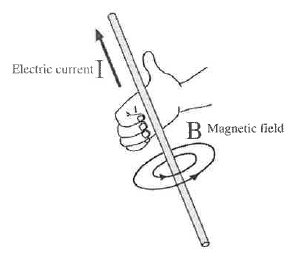


Figure : Right hand rule

If a current carrying wire is wound around a core, the core would behave like a magnet. The poles of the magnet can be determined using the right hand rule. If the direction of the current is changed, the north and south poles of the magnet will reverse. Since opposite polarities attract while like polarities repel, the winding of the armature is designed to harness this magnetic interaction to generate rotational motion.

##### 3.2 DC Motor Math Model

The equivalent circuit model for an ideal DC Motor is depicted in Fig. 5 below.

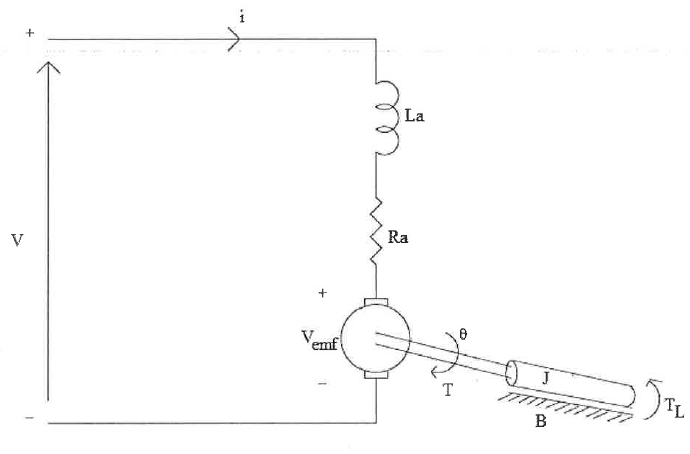


Figure : DC Motor equivalent circuit model

From Kirchhoff’s voltage law, the expression for the applied voltage is given as

(3)

where , and are the armature resistance, armature inductance, and back emf. From Newton’s Law of motion, the expression for the motor torque is given as

(4)

where , , and are the motor inertia, viscous friction coefficient, and the load torque on the motor. The motor torque is related to the motor electric current in a linear function

(5)

where is the motor torque constant. The back emf can be described as a linear function of motor rotational velocity as

(6)

where is the speed constant. Equations (3) – (6) can be used to get the relationship between the applied voltage and the motor displacement as

(7)

The torque constant and speed constant have the same value in SI units, so

(8)

From Eq. for a no-load condition,

(9)

At zero initial conditions, the above equation can be written in the Laplace domain as

(10)

### Lifting Device Math Model

##### 4.1 Problem Statement

The following scenario demonstrates the control of a motor to perform a task, in this case lifting a weight to a desired location. Figure 6 shows the setup: the motor is coupled to a pulley (A) of radius r. The pulley raises or lowers mass m when the motor rotates. The goal is to design a PID controller for controlling the position of a permanent magnet DC motor to lift a load of 1.124 kg. The motor and other system parameters are as follows:

Armature resistance, R = 20.5 Ω

Armature Inductance, H = 168 μH

Motor constant, K = 0.032 Nm/A (or V/rad/sec)

Gear ratio between motor and pulley, G = 49

Radius of the pulley, r = 0.0022 m

The rotor and pulley inertia and viscous friction coefficient are neglected. The string is assumed to be inextensible and friction between the string and the pulleys are also neglected.

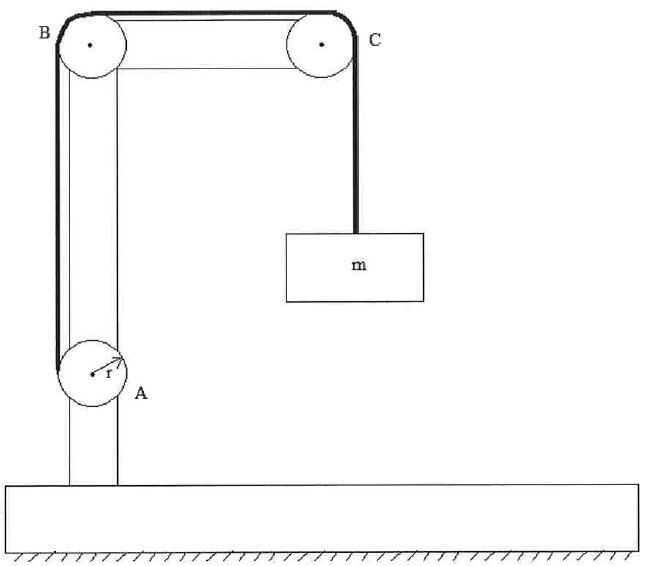


Figure : Pulley and weight system

##### Mathematical Modeling of the Open Loop System

As seen in Fig. 6, pulley A is coupled to the geared PM DC motor while pulleys B and C are idlers supporting the string. When pulley A rotates by an angle in the counterclockwise direction, the mass m will move up by a distance . Figure 7A and 7B show the free body diagrams of the pulley A and mass m respectively.



A B

Figure : Pulley (A) and Mass (B) Free Body Diagrams

For the moving mass, using Newton's Law results in

(11)

and for the rotating pulley after neglecting inertia and damping losses results in

(12)

Hence, the load on the motor will be

(13)

The relationship between the angular displacement of the motor shaft and gear output shaft is

(14)

From Eq. (14) and Eq. (15), the result for is

(15)

From Eq. (8) and Eq. (15), the result is

(16)

(17)

After neglecting inertia and damping losses in the motor equation, Eq. (17) reduces to

(18)

Applying a Laplace transform to Eq. (18) at zero initial conditions gives

(19)

(20)

Eq. (20) represents the open loop transfer function of the plant which can be represented by the block diagram as shown in Fig. 8. The term is the voltage required to balance the constant torque developed due to the gravitational force .

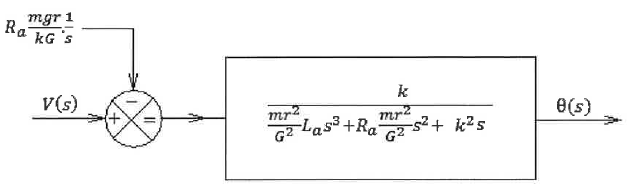


Figure : Open loop transfer function

**Mathematical Modeling of the Closed Loop System**

Next, a PID controller is introduced in the system which will ensure that the desired position is achieved with the given system performance requirements. The block diagram of the closed loop system is represented in Fig. 9.

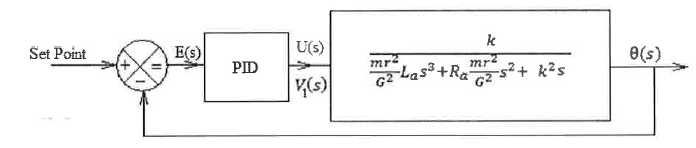


Figure : Closed loop transfer function

where . The transfer function of the PID can be obtained from the Laplace transform of Eq. (1) as

(21)

(22)

From Eq. (20) and Eq. (22), the closed loop transfer function can be obtained as

(23)

(24)

From Eq. (24) the characteristics of the closed loop system can be written as

(25)

### Control System Computer Modeling

##### 5.1 Computer Model

The mathematical model of the system and controller can be used as the basis for a computer model of the system. Computer models are used to conduct simulations which model a real system to predict its outcome. The control parameters can then be tested and optimized before a real system is tested. The computer model here is constructed using Matlab and Simulink.

##### 5.2 Script File

A Matlab script file (extension \*.m) contains a sequence of commands which execute computations. One use of a script file is to set up numeric values of parameters to use in later computations. For example, the command

m = 1.5;

Assigns the value of 1.125 to variable ‘m’. This variable then becomes part of the Matlab workspace and can be used in calculations. For example,

2\*m

will return a value of 3.0, or 2\*1.5.

plot(x,y)

creates a plot of variable y vs. x.

Figure 10 is the contents of the file “lifting\_model\_script.m” used for this lab. Commands are written in the script window (A). The run button (B) is used to run the script. Text following a % mark is a comment. They are ignored by the program but are used to convey information to the user.

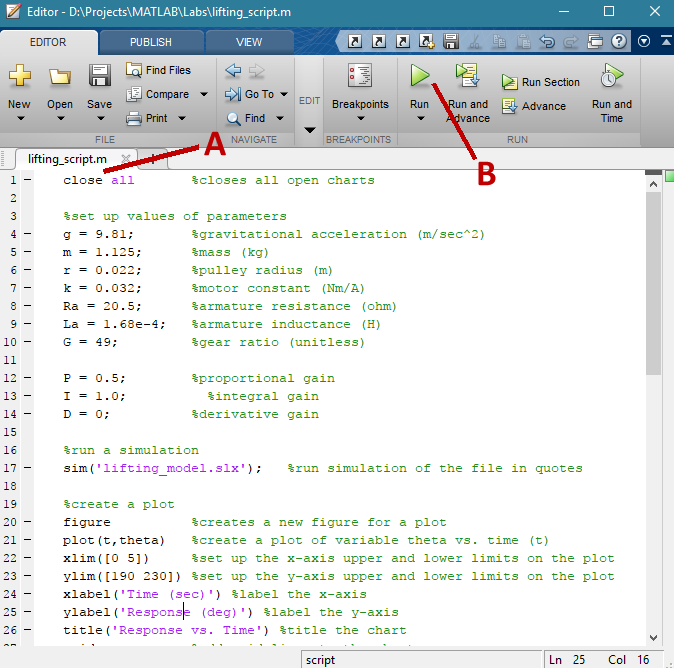


Figure : Matlab Script window

The “sim” command runs a Simulink model, described in the next section.

##### 5.3 Simulink File

The second component of the model, “lifting\_model.slx”, contains a block diagram of the model (Fig. 11). Here, mathematical models are implemented using a graphical model where signals (represented by arrows) pass from block to block.

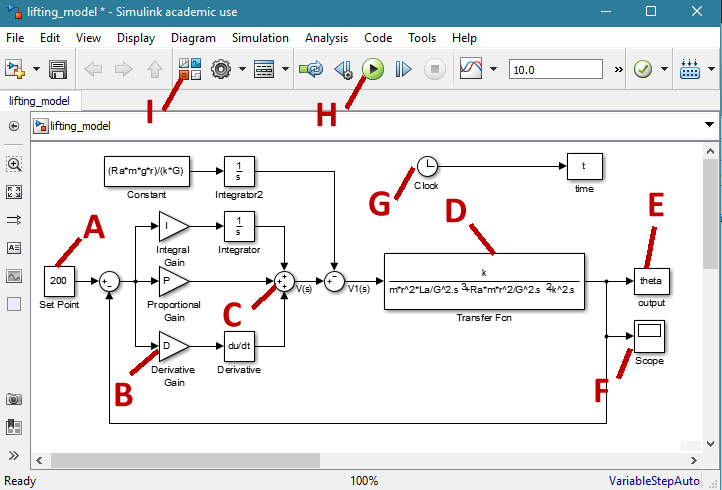


Figure : Simulink Model

The blocks labeled “constant” and “set point” are constant blocks (A). Signals originate from these source blocks. Gain blocks (B) multiply the signal by the value inside the block. The circular summation blocks (C) add or subtract multiple signals into one. The Transfer Function (D) block contains the transfer function of the model from Eq. (20). The output block (E) saves the signal to the workspace with the variable name in the block, while the scope (F) displays a graph of the connected signal. A clock (G) outputs the simulation time. The simulation is run by pressing the run button (H). Button (I) opens the library browser (Fig. 12), which is a list of all blocks which may be added to build a model. Blocks can be added by clicking and dragging them from the list (A) into the model. Blocks may be searched for by entering a name in the search window (B), or by clicking on categories in window (C).

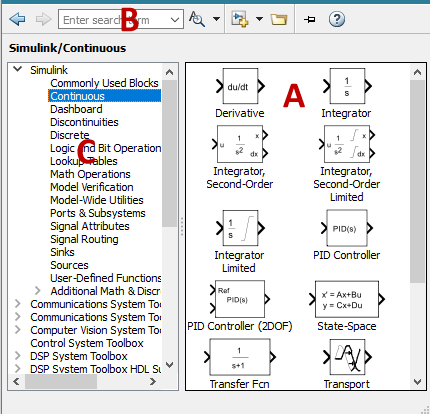


Figure : Library Browser

Then the connections between blocks can be created by right clicking the arrow on a block and dragging to another block (Fig. 13). A red arrow is represents a signal that is not connected. Right-click the red arrow and drag to another block to complete the connection.



Figure : Block Connections

Double-click any block to open a dialogue window in which the block’s value can be changed. There are two ways to specify block values:

1. Numerically: a number can be entered directly as the block value. For example, entering “5” in a gain block will multiply the signal passing through the block by 5.
2. Symbolically: a variable may also be entered into the block. The block will take on the value of the variable stored in the workspace. For example, a gain with a value of “m” will multiply the signal by 1.125, since 1.125 was previously assigned to m in the script. The advantage of this is that if the value of m needs to be changed, the change can be made once in the script. When the script is then run again, the value will be updated without needing to change m everywhere it occurs in the model.

For summation blocks, the number and type of connections is set by entering a list consisting of + (sum), – (subtract), and | (blank space). For example, ++- will create a block with two + connections and one – connection.

##### 5.3 Working with Charts

Figure 14 shows a Matlab figure window. The “data cursor” button (A) is used to label a point on the curve (B). Press the button, then click a point on the curve to create a label. Multiple labels can be created by holding the shift key and clicking on the curve. The “axis properties” choice in the edit menu allows the plot’s x and y axis limits and titles to be changed.

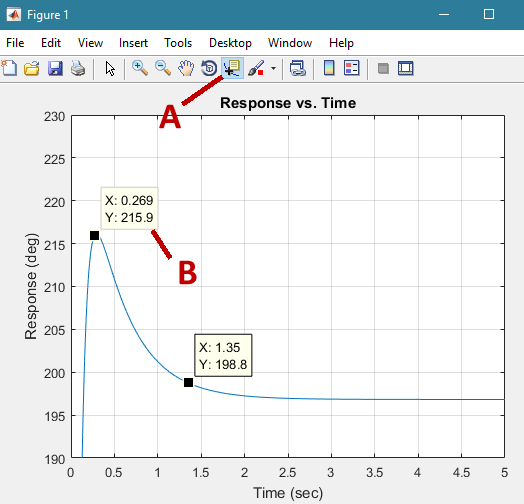


Figure : Figure window

# References

Thomas, G, “Position Control of a Permanent Magnet Brushed DC Motor”, Lawrence Technological University, 2009.

"DC Motors", PC Control Learning Zone, 2008.

<http://www.pc-control.co.uk/dc-motors.htm>

Seale, E., "Principles of Operation", DC Motors - How they work, in 4 parts, 2001.

<http://www.solarbotics.net/starting/200111\_dcmotor/200111\_dcmotor2.html>

Behnen, A., "Magnetic and Electric Fields", 2009.

<http://ffden-2.phys.uaf.edu/212\_fall2003.web.dir/Aaron\_Behnen/mag\_elecfields.html>

National Instruments, "PID Theory Explained", NI White Papers, 2011.

<http://www.ni.com/white-paper/3782/en/>

## Assignment:

The files “lifting\_script.m” and “lifting\_model.slx” contain the script and block diagram files for the lifting device model. Some blocks and connections are incomplete. Complete the model by referring to Fig. 11. Add values to gains P, I, and D. Create a plot of the system response. Determine the steady-state error and % overshoot when the set point value is 200 deg, and the PID gains are P = 0.5, I = 7.0, and D = 0.