SPEED CONTROL OF DC MOTOR USING PI CONTROLLER

MOHD AZRI BIN ABD MUTALIB

A report submitted in partial fulfillment
of the requirements for the award of the degree of
Bachelor of Electrical Engineering

Faculty of Electrical Engineering
Universiti Malaysia Pahang

MAY, 2008

ACKNOWLEDGEMENT

First of all I am grateful to ALLAH S.W.T for blessing me in finishing my final year project (PSM) with success in achieving my objectives to complete this project.

Secondly, I want to thank my family for giving morale support and encouragement in completing my project and also throughout my study in Universiti Malaysia Pahang (UMP) as they are my inspiration to success. I also would like to thank my supervisor Mr Mohd Syakirin Bin Ramli for guiding and supervising my final year project. He has been very helpful to me in finishing my project and I appreciate every advice that he gave me in correcting my mistakes. I apologize my supervisor if any mistakes and things that I done wrong while doing my project. The credit also goes to Assiociate Prof. Nasir B Shaikh Abd Rahman as the dean of Electrical Engineering Faculty and Mr Reza Ezuan Bin Samin as the coordinator of final year project with their cooperation and guide about my final year project from the beginning.

Last but not least, I want to thank all my friends that have gave me advice and encouragement in completing my project. Thank you very much to all and May ALLAH bless you.

ABSTRACT

The development of technologies affects the demands of industries at the present time. Thus, automatic control has played a vital role in the advance of engineering and science. In today’s industries, control of DC motors is a common practice. Therefore, implementation of DC motor controller is required. There are many types of controller that can be used to implement the elegant and effective output. One of them is by using a PI controller. PI stands for Proportional and Integral Controllers which are designed to eliminate the need for continuous operator attention thus provide automatic control to the system. Cruise control in a car and a house thermostat are common examples of how controllers are used to automatically adjust some variable to hold the measurement (or process variable) at the set-point. This project is focusing on implementing PI controller to control speed of a dc motor. The overall project is divided into two parts. The first part is concern on the simulation using MATLAB simulink where the dc motor is modeled and PI controller is tuned using Ziegler-Nichols rules and software tuning. The second part is implementing the simulation. This part is divided into another two parts, Graphical User Interface (GUI development and hardware interfacing. GUI is built using National Instrument LabVIEW software with implementation of PI controller. An oscilloscope also had been build there. Hardware interfacing part is built with Mitsumi dc mini-motors M31E-1 Series, speed sensor and analog to digital converter, DAC8032. As the result PI controller is capable to control the speed of dc motor followed the result from simulation.

ABSTRAK

Perkembangan teknologi masa kini telah memberikan tekanan kepada industri. Maka, sistem kawalan automatik telah memainkan peranan yang penting sejajar dengan kemajuaan sains dan kejuruteraan. Dalam era pembangunan industri masa kini, kawalan *dc motor* adalah amalan biasa dilakukan. Oleh yang demikian, perlaksanaan sistem kawalan *dc motor* amatlah diperlukan. Terdapat pelbagai jenis sistem kawalan *dc motor* yang boleh digunakan dalam mendapatkan keluaran (*output* yang mantap dan efektif. Antaranya termasuklah PI Controller. PI adalah singkatan dari *Proportional* dan *Integral Controller*. Sistem kawalan ini dibangunkan untuk memenuhi keperluan sistem kawalan tanpa henti. Kawalan *Cruise* dan *thermostat* di rumah adalah contoh yang biasa digunakan dalam menunjukan bagaimana sistem kawalan digunakan untuk mengawal secara automatik pembolehubah yang tertentu untuk mengekalkan nilai tetapannya. Projek ini memberi penekanan kepada perlaksanaan sistem kawalan PI dalam mengawal kelajuan dc motor. Secara keseluruhannya, projek ini terbahagi kepada dua. Bahagian yang pertama menceritakan mengenai proses simulasi menggunakan perisian MATLAB simulink di mana *dc motor* telah dimodelkan dan sistem kawalan PI dilaras mengikut prinsip Ziegler-Nichols dan disusuli dengan larasan secara perisian. Bahagian yang kedua adalah melaksanakan simulasi tersebut. Bahagian ini juga terbahagi kepada dua bahagian iaitu membangunkan *Graphical User Interface* (GUI) dan litar elektronik. GUI dibangunkan dengan menggunakan perisian National Instrument LabVIEW dengan melaksanakan sistem kawalan PI. Osiloskop juga turut dibangunkan disini. Pembangunan litar elektronik pula dijalankan dengan Mitsumi *dc mini-motors*, siri M31E-1, sensor kelajuan dan penukar nilai digital kepada analog, DAC8032. Sebagai keputusannya, sistem kawalan PI adalah mampu untuk mengawal kelajuan *dc motor* mengikut keputusan simulasi.

TABLE OF CONTENTS

3 METHODOLOGY

4 RESULTS AND DISCUSSIONS

5 CONCLUSION AND RECOMMENDATIONS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PI	-	Proportional Integral
DAC	-	Digital to Analog Converter
DAQ	-	Data Acquisition
dc	-	Direct Current
GUI	-	Graphical User Interface
LabVIEW	-	Laboratory Virtual Instrument Engineering Workbench
GND	-	Ground
OSC	-	Oscillator
LED	-	Light Emitting Diode
MSB	-	Most Significant Bit
LSB	-	Least Significant Bit
PC	-	Personal Computer

LIST OF SYMBOLS

μ - Micro
k - Kilo
V - Volts
Ω - Ohm
mH - MiliHenry
N - Newton
m - Meter
rad - Radian
s - Second
dc - Direct Current
K_p - Proportional gain
K_I - Integral gain
R - Desired Input Value
Y - Actual Output
e - Error Signal
u - Signal
L - Delay Time
T - Time Constant
∞ - Infinity
Kcr - Critical Gain
Pcr - Corresponding Period
% - Percent
Va - Voltage Source
La - Inductance
emf - Electromotive Force

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Dc motors have long been the primary means of Electrical Traction. DC motor is considered as a SISO system having torque/speed characteristics compatible with most mechanical loads [1]. Hence, dc motors are always a good proving ground for advanced control algorithm because the theory is extendable to other types of motors.

Normally closed loop operation with PI controllers in the inner current loop and the outer speed loop is employed for speed control**;** the design of PI controller is generally carried out using time/frequency domain analysis. The speed response of the drive with PI controllers designed with the above techniques may be satisfactory but not necessarily be the best, since they do not pose any constraint on settling time, overshoot / undershoot etc. In any classical PID control problem, the required controller parameters should be optimally designed. Despite the method of Zeigler-Nichols ultimate cycle tuning scheme, these parameters can be optimally obtained via Genetic Algorithms**.**

Genetic Algorithm's are general purpose optimization techniques which use a direct analogy of natural evolution where stronger individuals would likely be the winners in a competing environment [2]. Application of genetic algorithm in Electrical Machines anti control systems is quite new. This paper explains the application of genetic algorithm for the design of a PI Controller for speed control DC motor.

1.2 Project objective

The objective of this project is to design PI controller, develop Graphical User Interface (GUI) with implementing PI controller and compare the result from simulation and experiment in controlling speed of dc motor.

1.3 Project scope

This project is to design and construct a PI controller which is performing to control a speed of DC motor. As a machine's performance is a vital factor for a big production line, this project will examine the efficiency and performance of a Dc motor with and without controller methodology. Thus, the focuses of this project are as stated below:

- i) Create the flow diagram of project
- ii) Do the offline simulation to get the expected result with and without controller.
- iii) Design a Graphical User Interface (GUI) with implementing PI controller.
- iv) Do the online simulation and compare the result to offline simulation before.
- v) Result analysis.

CHAPTER 2

LITERATURE REVIEW

2.1 PI Controller

2.1.1 Proportional Band (P)

With proportional band, the controller output is proportional to the error or a change in measurement (depending on the controller).

Controller output = (error)*100/ (proportional band) (2.1)

With a proportional controller offset (deviation from set-point) is present. Increasing the controller gain will make the loop go unstable. Integral action was included in controllers to eliminate this offset [3].

2.1.2 Integral (I)

With integral action, the controller output is proportional to the amount of time the error is present. Integral action eliminates offset.

Controller output = (1/intergeral) (Integral of) e(t) d(t) (2.2)

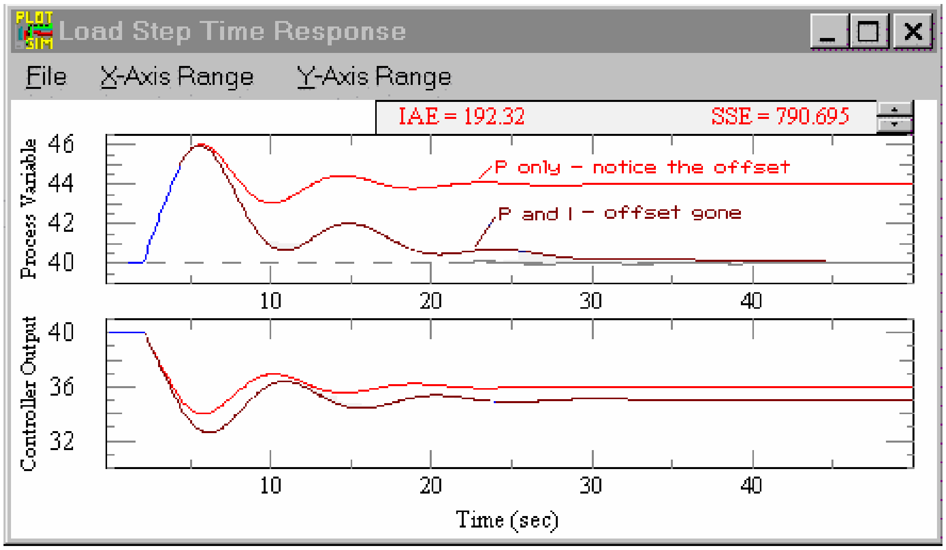


Figure 2.1: Load Step Time Response

In Figure 2.1, notice that the offset (deviation from set-point) in the time response plots is now gone. This is done for integral action which eliminated the offset. The response is somewhat oscillatory and can be stabilized by adding derivative action [3].

Integral action gives the controller a large gain at low frequencies that results in eliminating offset and "beating down" load disturbances. The controller phase starts out at –90 degrees and increases to near 0 degrees at the break frequency. This additional phase lag is what you give up by adding integral action.

2.1.3 The two-term controller

The transfer function of the PI controller looks like the following:

$$\frac{K_p s + K_I}{s} = K_p + \frac{K_I}{s} \qquad (2.3)$$

K_p = Proportional gain

K_I = Integral gain

The variable (e) represents the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PI controller, and the controller computes the integral of this error signal. The signal (u) just past the controller is now equal to the proportional gain (Kp) times the magnitude of the error plus the integral gain (Ki) times error [2].

$$u = K_P e + K_I \int e\,dt \tag{2.4}$$

This signal (u) will be sent to the plant, and the new output (Y) will be obtained. This new output (Y) will be sent back to the sensor again to find the new error signal (e). The controller takes this new error signal and computes its derivative and it's integral again. This process goes on and on.

2.1.4 The characteristics of P and I controllers

A proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (K_i) will have the effect of eliminating the steady-state error, but it may make the transient response worse. Effects of each of controllers K_p and K_i on a closed-loop system are summarized in the table shown in Table 2.1.

Table 2.1: Effect of K_p and K_i

Controller respond	**Rise Time**	**Overshot**	**Settling Time**	**S-S error**
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate

Note that these correlations may not be exactly accurate, because K_p and K_i are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you are determining the values for K_p and K_i [4].

2.1.5 Tuning Rules Of PI Controller

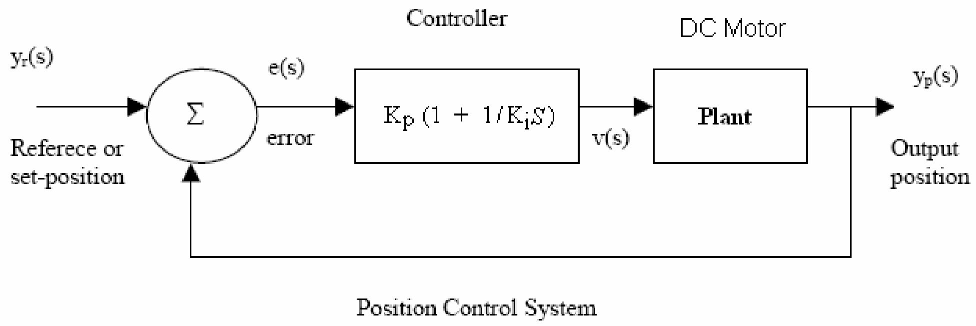


Figure 2.2: Position Control System

Figure 2.2 shows the position of PI controller in dc motor speed control system. If the mathematical model of the plant (dc motor) can be derived, then it is possible to apply various design techniques for determine parameters of the controller that will meet the transient and steady-state specification of the close loop system.

The process of selecting controller parameter to meet given performance specification is known as controller tuning. Ziegler and Nichols suggested rules for tuning PI controller (mean to set the values of K_p and K_i) based on the experimental step response or based on the value of K_p that result is marginal stability when only proportional control action is used [5]. Ziegler-Nichols rules, which are briefly presented in the following, are useful when mathematical models of plans are not known. These rules can, of course, be applied to design of system with known mathematical models. Such rules suggest a set of values of K_p and K_i that will give a stabile operation of the system. However, the resulting system may exhibit a large maximum overshoot in step response, which is unacceptable. In such a case, we need series of fine tunings until an acceptable result is obtained. In fact, the Ziegler-Nichols tuning rules give an educated guess for parameter values and provide a stating point for fine tuning, rather then giving the final settings for K_p and K_i in a single shot.

Ziegler and Nichols proposed rules for determining values of proportional gain, K_p and integral time, K_i based on the transient response characteristics of given plant. Such determination of the parameter of PI controller or tuning of PI controller can be made by engineers on-site by experiments on the plants. Numerous tuning rules for PI controllers have been proposed since the Ziegler-Nichols proposal. They are available in the literature and from the manufactures of such controllers. There are two methods called Ziegler-Nichols tuning rules; the first method and the second method.

First method

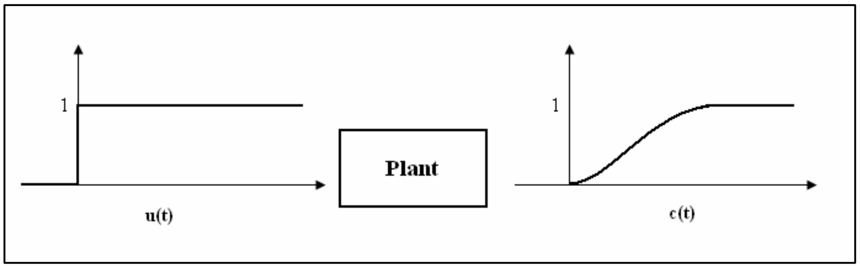


Figure 2.3: Response to a unit-step input

In the first method, we obtain experimentally the response of the plant to a unit-step input, as shown in figure 2.3. If the plant involves neither integrator(s) nor dominant complex-conjugate poles, then such a unit-step response curve may look S-shape as shown in figure. This method applies if a response to a step input exhibit an S-shaped curve. Such step-response curves may be generated experimentally of from dynamic simulation of the plant [2].

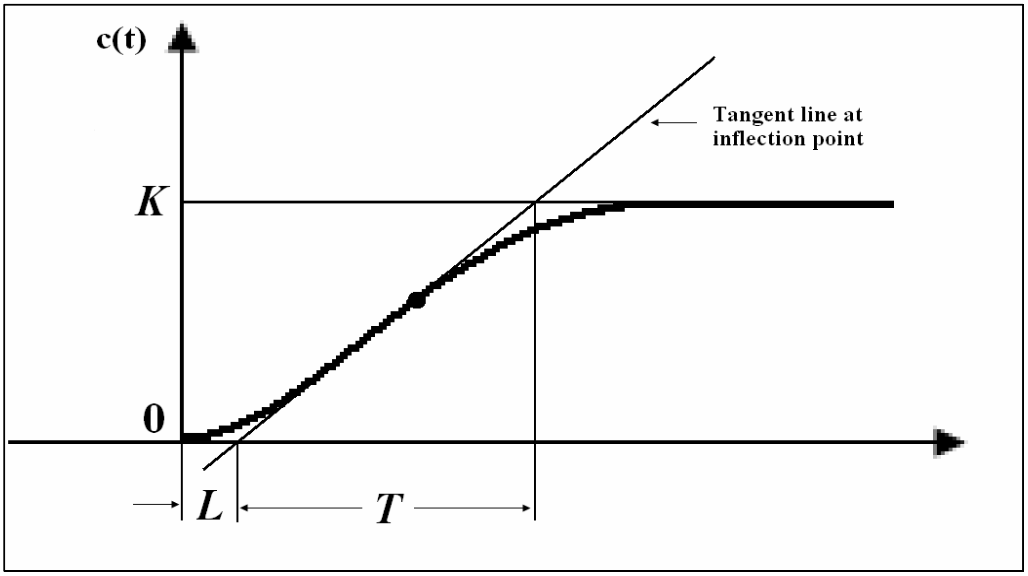


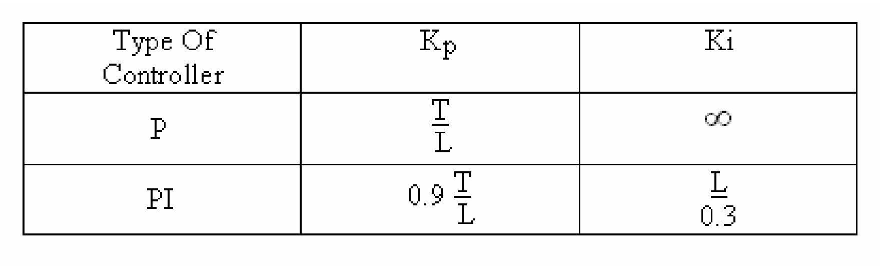
Figure 2.4: Response with tangent line

The S-shape curve may be characterized by two constants, delay time L and time constant T. The delay time and constant time are determined by drawing a tangent line (Figure 2.4) at the inflection point of the S-shaped curve and determining the intersections of the tangent line with the time axis and line c(t) = K as shown in figure. The transfer function C(s) /U(s) may then be approximated by fist-order system with transport lag as follows;

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1} \qquad (2.5)$$

Ziegler and Nichols suggested setting the value of K_p and K_i according to the formula shown in Table 2.2:

Table 2.2: Ziegler and Nichols' setting value for KP and Ki



Type Of Controller	K_p	Ki
P	$\frac{T}{L}$	∞
PI	$0.9\,\frac{T}{L}$	$\frac{L}{0.3}$

Notice that the PI controller tune by the first method of Ziegler-Nichols rules gives

$$G(s) = Kp\left(1 + \frac{1}{KiS}\right) \qquad (2.6)$$

$$= 0.9\frac{T}{L}\left(1 + \frac{1}{L/0.3}\right)$$

$$= 0.9\frac{T}{L}\left(1 + \frac{0.3}{L}\right)$$

The first method explains about how to tune the PI controller in the first order system only. But this project is about second order close loop system. A study need to be done to tune the second order system in the second method.

Second Method

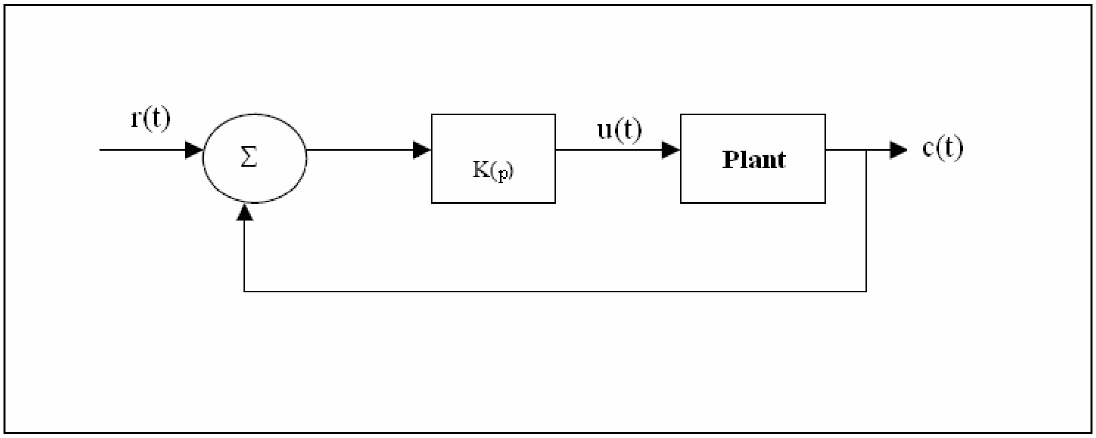


Figure 2.5: Second Method Control System

In the second method, $K_i = \infty$ was set. Using the proportional control action only (see figure), increase K_p from 0 to a critical value Kcr at which the output first exhibits sustained oscillations. (If the output does not exhibit sustained oscillations for whatever value of K_p may take, then this method does not apply) Thus, the critical gain Kcr and the corresponding period Pcr are experimentally determined (see figure). Zeigler and Nichols suggested that we set the value of the parameters K_p and K_i according to the formula shown in the Table 2.3.

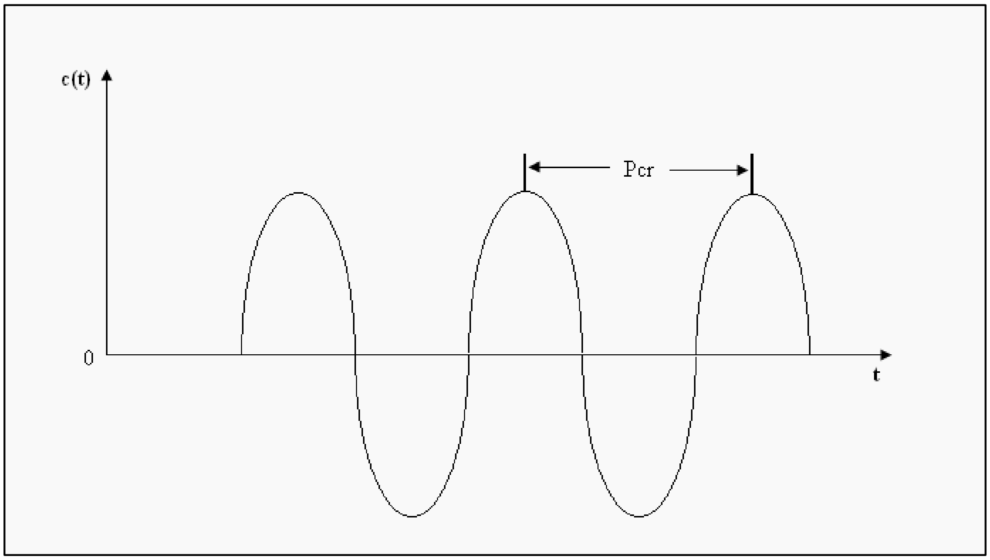


Figure 2.6: Position of Pcr

Table 2.3: Second method of Zeigler-Nichols rules

Type Of Controller	Kp	Ki
P	0.5*K*cr	∞
PI	0.45*K*cr	$\frac{1}{1.2}$ Pcr

Notice that the PI controller tuned by the second method of Zeigler-Nichols rules in Table 2.3 gives.

$$G_c(s) = Kp\left(1 + \frac{1}{KiS}\right)$$

$$= 0.45\ Kcr\left(1 + \frac{1}{1.2\ Pcr\ S}\right)$$

2.2 Dc Motor

2.2.1 History of Dc Motor (Plant)

Electric motors exist to convert electrical energy into mechanical energy. This is done by two interacting magnetic fields; one stationary and another attached to a part that can move [7].

DC motors have the potential for very high torque capabilities; although this is generally a function of the physical size of the motor, are easy to miniaturize, and can be "throttled" via adjusting their supply voltage. DC motors are also not only the simplest, but the oldest electric motors.

The basic principles of electromagnetic induction were discovered in the early 1800's by Oersted, Gauss, and Faraday. In 1819, Hans Christian Oersted and Andie Marie Ampere discovered that an electric current produces a magnetic field. The next 15 years saw a flurry of cross-Atlantic experimentation and innovation, leading finally to a simple DC rotary motor [8].

Because of the work of these people, DC machines are one of the most commonly used machines for electromechanical energy conversion. Converters which are used continuously to convert electrical input to mechanical output or vice versa are called electric machines as shown in Figure 2.7. An electric machine is therefore a link between an electrical system and a mechanical system. In these machines, the conversion is reversible. If the conversion is from mechanical to electrical, the machine is said to act as a generator. If the conversion is from electrical to mechanical, the machine is said to act as a motor. Therefore, the same electric machine can be made to operate as a generator as well as a motor.

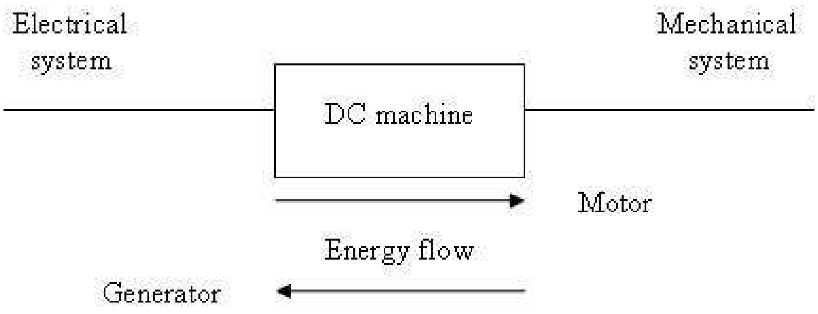


Figure 2.7: Electromechanical energy conversion

DC machines may also work as brakes. The brake mode is a generator action but with the electrical power either regenerated or dissipated within the machine system, thus developing a mechanical braking effect. It also converts some electrical or mechanical energy to heat, but this is undesired.

The major advantages of DC machines are easy speed and torque regulation. The major parts of any machine are the stationary component, the stator, and the rotating component, the rotor.

2.2.2 Principles of operation.

In any electric motor, operation is based on simple electromagnetism [9]. A current carrying conductor generates a magnetic field which when placed in an external magnetic field; it will experience a force proportional to the current in the conductor and to the strength of the external magnetic field. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding.