Real Time Digital Speed Control System for DC Servo Motor Using LabVIEW 8.5 Package

Areej Alaa Hassen*

Received on:16/6/2009
Accepted on: 5/11/2009

Abstract

This paper describes the design and implementation of a personal computer based closed loop DC motor speed control system Using LabVIEW 8.5 Package for data manipulation and interface control. Tuning the parameters of the PID controller is done using trial and error method by conducting simulation on the system model using Matlab package. This method is used to find the best system response depending on the tuning parameters of the PID controller. These parameters are then implemented in the designed real time digital PID controller system based on LabVIEW package.

Carrying out the task of tuning the PID controller on the real time model requires great effort and time consuming especially in the early stages. Thus the PID controller tuning is firstly carried out on the simulation model in Matlab which is time saving and gives close parameter approximation for applying in the real time system directly. The PID control action in the real time system shows more oscillation in comparison with the PID simulation control action. Simulation and real time results for the speed control of the DC motor experiment were found to have a high degree of agreement in maintaining the desired speed of the motor.

Keywords: Speed Control, LabView, Real Time, PID, Matlab.

التحكم بسرعة محرك تيار مستمر مؤازر في الزمن الحقيقي باعتماد الحقيبة البرمجية LabVIEW 8.5

الخلاصة

هذا البحث يصف تصميم و تنفيذ نظام السيطرة لسرعة محرك التيار المستمر باعتماد حقيبة LabVIEW 8.5 مع حاسوب شخصي من طراز IBM.
هذه الطريقة استخدمت للحصول على أفضل استجابة للنظام اعتمادا على معاملات التوليف للمسيطر PID التي تم الحصول عليها باستخدام محاكاة بواسطة برنامج Matlab ثم أخذت معاملات المسيطر وطبقت في نظام الزمن الحقيقي للمسيطر الرقمي PID باعتماد حقيبة LabVIEW وذلك لتقليل الوقت المطلوب لإيجاد معاملات التوليف المطلوبة في نظام الزمن الحقيقي مباشرة.
إن أداء المسيطر PID في نظام الزمن الحقيقي كان أكثر تذبذبا من أداء المسيطر PID في نظام المحاكاة وكانت نتائج المحاكاة و الزمن الحقيقي لمنظومة السيطرة على سرعة محرك التيار المستمر ناجحة للمحافظة على سرعة المحرك المطلوبة.

Introduction

DC motor drives have occupied A wide spectrum of applications in industry. DC motors are used in machine tools, printing presses, conveyors, fans, pumps, hoists, cranes, paper mills, textile and rolling mills. Small DC motors are used primarily as control devices and

*Control & Systems Engineering Department, University of Technology/ Baghdad

servomotors for positioning and tracking [1].

The neural networks are used in a speed control loop applied to a DC motor and the learning capability of neural networks are implemented an auto-adaptive control structure to learn the dynamic behavior of the SCR-driven DC motor [2].

A direct nonlinear adaptive state regulator is derived based on dynamic neural networks and it is applied to control the speed of a nonlinearized DC motor and the control algorithm is that it covers the situation where the magnetic flux continuously varies [3]. DC series motors are preferred for mechatronic applications requiring high torque/speed ratios. The design and implementation of an open loop DC motor speed control that is based on a micro-controller and IGBT drive stage that can predict the dynamic behavior of systems consisting of mechanical and electronic modules is very desirable [4]. A DC motor drive as well as the corresponding identification and robust stability analysis are used in a robust controller for an experimental laboratory – scale position control system, this technique is applied and then the parameterization by internal model control is also used [5]. Most mechanical systems used in industry are composed of masses moving under the action of position and velocity dependent forces. The nonlinear modeling and identification of a DC motor rotating in two directions together with real time experiments of mechanical systems makes up an essential stage in practical control design and applications [6]. Particular reference is made to the stability of the rectified DC link voltage under load disturbance is used in speed control of an internal combustion engine for series – hybrid electric vehicle applications [7]. The control strategies for precise synchronous control of multi–axis rotating systems used to achieve precise servo system are summarized in two main points: an acceleration control is adopted in an inner loop to ensure robustness of speed of each axis against disturbances, and then a maximum error comparison method has been introduced to minimize not only position synchronous errors but also speed errors in motors used to drive all the axes [8].

Many industrial applications use PID control to maintain constant process variable. The output of PID controllers (proportional- integrative-derivative controllers) is a linear combination of the input, the derivative of the input and the integral of the input therefore it is widely used and enjoys significant popularity, because of its simplicity, effectiveness and robustness [9].

Actually, the trail and error tuned PIDs often perform in a non-optimal way. But even through further fine–tuning is possible and sometimes necessary to provide a good starting point, their usefulness is obvious when no model of the plant is available, and thus no analytic means of tuning a controller exists. This method may also be used when a model is known.

In the work presented in this paper, experimental investigation is carried out for the appropriate tuning parameters of the PID controller that governs the speed of a DC motor using trial and error method to obtain the best system response achievable based on a Matlab simulation package. These parameters are then applied in the designed real time digital PID controller system based on LabVIEW (lab Virtual Instrument Engineering Workbench) package and the results obtained are compared against those of the simulation.

The organization of this paper is as follows: *Section one* contained a brief description of the designed system, *Section two* contains the proposed scheme for designing and implementing a real time speed control system for a DC motor. The analysis of the mathematic model of the system and a description for the hardware design is also covered. *Section three* describes a control methodology for PID controller based on LabVIEW package. *Section four* describes simulation results obtained from Matlab package. *Section five* describes the real time results obtained from the designed system via LabVIEW package. *Section six* contains the conclusions and results discussion.

2- Experimental Arrangement of the Real Time Implementation:

In this section the proposed scheme for designing and implementing the real time speed control of the DC motor is described. During the course of experiments, an IBM compatible PC is used for data acquisition and control in real time. Data acquisition is accomplished through the standard parallel port and the controller is implemented in LabVIEW 8.5 a graphical programming language.

The experimental setup for the real-time speed control of the DC motor is shown in figure (1), the block diagram for this experimental arrangement is shown in figures (2). The setup consists of a DC motor with a DC tacho generator to sense the speed of motor, a DC power supply that provides power to the motor and the rest of the circuitry, and the data acquisition + motor drive board.

The DC motor system can be considered a linear second-order time delay system as the mathematical model describes it.

2-1 Dynamic model of the linear DC Motor drive system:

The linear dynamic equations of the system can be derived using a permanent magnet DC motor model with a two mass model equivalent system [10]:

$$v_a = R_a i_a + L_a \frac{di_a}{dt} + K_b w_m \quad \text{….(1)}$$

$$J_m \frac{dw_m}{dt} + B_m w_m = t_m = K_t i_a \text{…..(2)}$$

Where R_a, L_a, K_b, K_t are the DC motor parameters, and v_a, i_a are the DC motor voltage and current. ω_m is motor speed, τ_m the motor torque, J_m the motor inertia, and B_m the damping coefficient. The modeling of the complete motor drive system includes the nonlinear effect “dead zone” which can be surmounted by adding a small dc drive voltage to the motor in order to minimize its effect as shown in the motor drive electronic circuit design [11].

By taking the Laplace transformation of the equations (1,2) [10].

$$V_a(s) = R_a I_a(s) + L_a S I_a(s) + K_b w_m(s) \text{...(3)}$$

where $K_b w_m(s)$ is e.m.f. equation.

$$V_a(s) - K_b w_m(s) = I_a(s)(R_a + L_a S) \text{..(4)}$$

$$J_m S w_m(s) + B_m w_m(s) = t_m(s) = K_t I_a(s) \text{..(5)}$$

$$w_m(s)(J_m S + B_m) = t_m(s) \quad \text{…..(6)}$$

The block diagram of the DC motor model is show in figure (3).

The parameters of the DC servo motor implemented in the designed system are shown in table (1) [12].

2-2-Hardware Design:

The circuit diagram of the designed electronic circuitry of the digital speed control system is shown in figure (4). It consists of the permanent magnet DC servo motor which has a built in DC Tacho generator with a sensitivity figure equal to $31.5 mV/(rad/\sec)$. The motor has a speed range of operation "0 rad/sec to 157rad/sec". The signal conditioning circuit includes a first order low pass filter that has a cutoff frequency of 20Hz which will remove noise components possible to occur in the sensor outputs especially within mains supply frequency at 50Hz. This filter removers noise components within mains frequency effectively from the tacho generator output prior to delivering it to the level conditioning amplifier with gain equal to "2". This amplifier is built using TLO64 CMOS quad operational amplifier [11, 13]. The output of this amplifier is fed to personal computer through ADC574 the analog to digital converter. It is a 12 bit high speed low power successive approximation converter with LSB equal to 2.44mV/count [14]. TTL devices are used to design the glue logic circuitry required to interface with standard parallel port (Line Printer Port) LPT1 working in SPP mode [15].

Inside the personal computer, software compares the sensed speed signal received via this interface with the set point reference speed. The resulting error is given as an input to the digital PID controller. The PID controller attempts to reduce the error to zero by changing the control action “input voltage to the DC motor” which is in the form of digital data varying within "0Hex – FFHex". These data are sent to the motor drive electronics though the "Parallel Printer Port" to ZN428 which is an 8 bit digital to analog converter (DAC) with range of output voltage equal to "0Volts to 3.75Volts". The output of this DAC is sent to an op-amp based amplifier that has a gain of "5.17". This amplifier is also built using TLO64 CMOS operational amplifier. The output of this amplifier is fed to the motor drive stage to furnish the final output required to drive the DC servo motor. The motor accepts drive voltages ranging from "0V to 18V" which will drive the motor through the speed range of "0 rad/sec to 157 rad/sec".

The electronic circuit includes a section that generates the required offset in the drive voltage needed to minimize the dead zone in the operation of the DC motor [11, 13].

3- Control Methodology Based on LabVIEW Packages:

By varying the input voltage of the DC motor, the output speed can be varied and controlled. The PID control algorithm has been implemented; the details are discussed in the following section.

3-1 The PID Controller :

The input to the actuator or the output ($u_{(t)}$) of the PID controller is the summation of proportional gain, integral action, and derivative action which is expressed as [16]:

$$u_{(t)} = K_P e_{(t)} + K_I \int e_{(t)} dt + K_D \frac{de_{(t)}}{dt} \quad ..(7)$$

Where $e_{(t)}$ is the error between the set-point (w_{des}) and the actual output ($w_{out}(t)$), (KP) Proportional gain, (KI) Integral gain, and (K_D) derivative gain.

The PID parameters affect system dynamics, they are mostly effective in manipulating four major characteristics of the closed-loop step response. These are: Rise Time, Overshoot, Settling Time, and Steady-state Error.

The effect of increasing each of the controller parameters KP, KI and K_D can be summarized in table (2) [16].

3-1-1 Proportional Virtual Instrument:

For proportional controller only, the following equation applies [16]:

$$P=K_p e(t) \qquad ......(8)$$

In order to get the proportional action simply multiply the error e(t) by the constant of proportionality gain K_p. The block diagram for which is shown in figure (5).

3-1-2 Integral Virtual Instrument

The integral action is evaluated by using the trapezoidal method [16]. The mathematical representation of the integral action is as follows [16]:

$$\int_0^n e_{(t)}.dt = \sum_{k=0}^{n} \left(\frac{e(kT)+e(k+1)T}{2}\right)T \; .....(9)$$

Where k =0, 1, 2…n, and T $=\Delta t$ =sampling rate.

The block diagram for the integral virtual instrument (VI) is shown in figure (6). When the integral action of the virtual instrument is called, the while loop executes once because the conditional terminal is connected to Boolean constant (false). The error is passed through the error control. The error value is added to the value of the error from the past iteration. This is done by shift register with an initial value equal to zero. Then the value of the summation is divided by 2 and multiplied by the integral gain K_I and the sampling time Δt.

3-1-3 Derivative Virtual Instrument:

The derivative action is evaluated by using the backward difference method [16]. The mathematical representation of the derivative action is as follows [16]:

$$\frac{de_{(t)}}{dt} = \frac{e(kT)-e(k-1)T}{T} \; .....(10)$$

The block diagram of derivative VI is shown in figure (7). The past value of the error is subtracted from the current error value, then the result is divided by the sampling time Δt, and the final result, after division, is multiplied with the derivative gain K_d. using the select function, a condition to prevent the division by zero (in case that Δt was zero) is included. If Δt is larger than zero, the calculated value passes out to the derivative action indicator.

3-1-4 Δt Virtual Instrument

This virtual instrument is necessary to calculate the time period between every two samples Δt. This VI is very important because it is needed for real time control. It has only one output and has no input. The block diagram is shown in figure (8). The tick count function returns the internal timer of the CPU in millisecond. In the first iteration of the loop the tick count function gives the internal timer value, and then the initial value stored in the shift register (which is equal to zero in the first iteration) is subtracted from this value. The result is the time in millisecond elapsed between the first call and the second call of this VI.

After multiplying this value by number (desired sampling time), the result is in seconds or milliseconds.

3-2 The Complete PID Virtual Instrument

The PID output is the sum of the outputs of the proportional, integral and derivative actions. The block diagram of the PID VI is shown in figure (9). The value of process variable; the error $e_{(t)}$ is passed to the PID VI through the error controls. The output of the PID passes out through the PID output indicator. the output of the PID is within the range between upper limit value (FF hex) and lower limit value (00 hex) which in turn controls the motor drive voltage from 0 to 18 volts resulting in motor speed variation between zero rad/sec to 157 rad/sec "full speed".

4- Simulation Results

This section discusses the mapping between the real process and the simulation carried out using Matlab (7.3) package. It is shown in figure (10). The proposed PID controller in conjunction with the input voltage to the DC motor unit which has a linear relationship with saturation transfer function, it has a slope equal to (18/255). This slope is used as a mapping gain and to limit the motor drive voltage to 18Volts which is chosen in accordance with the DC motor modeling description.

The open loop step response of the speed of the DC motor is shown in figure (11), when applying a step change in the input voltage of the motor equal to (0.5645) volt, it will increase the motor speed by 1 rad/sec with reference to its speed at the initial condition which is equal to zero rad/sec. The settling time for the speed of the DC motor is equal to 1.6sec.

The Matlab package is used to simulate the operation of the real motor drive system. By applying the block diagram in figure (10) and executing many trial runs to find the best system response by varying the parameters of the PID controller, the PID parameters that produced the best achievable system response are found. PID controller parameters that were obtained from the simulation package are shown in table (3).

The proper tuning of the PID parameters is an essential task required to obtain better control. The parameters of the PID controller have been determined using trial and error method carried out on the simulation. The closed loop time response of the DC motor speed control system with PID-controller in Matlab simulation is illustrated in figures (12-a, b, c) for the initial speed of zero rad/sec.

Figure (12-a) shows the response of the DC motor speed output to a step change, it had no over shoots, the steady-state error was equal to zero and the transient time was approximately equal to 0.5 sec when the desired change in speed was (5, 15, 25,15 and 5) rad/sec.

Figure (12-b) is the response obtained after introducing feedback PID control action. It had few spikes in response to the desired step change in motor speed with very small oscillation. But it can be seen that it has kept the speed of the DC motor within desired range.

Figure (12-c) shows the error between the desired speed output and the actual speed output of the DC motor. It is very small in the transient and becomes very close to zero in steady state.

5- Real Time Results

In this section, the fine tuned parameters of the PID controller obtained from the simulation are applied in the discrete time "Real Time Computer Controller" after converting the continuous PID parameters controller of the table (3) to discrete PID controller using Euler's method [16]:

$$U(z)=(K_P+K_I\frac{T_S}{Z-1}+K_D\frac{Z-1}{T_SZ})E(z)..(11)$$

$$U(k)=U(k-1)+E(k).(K_P+\frac{K_D}{T_S})+E(k-1).$$

$$(K_IT_S-2\frac{K_D}{T_S}-K_P)+E(k-2).(\frac{K_D}{T_S})\ ...(12)$$

The sampling interval for the DC motor speed control was chosen to be (0.1) second using Shannon theorem $T_S=0.1\,\text{sec}$.

Figure (13-a) shows the speed response of the DC motor in real time. It can be seen that it is a fast response with oscillation output value in the range of $(\pm 0.4(rad/\text{sec}))$.

Figure (13-b) shows the response after implementing the feedback PID control action. It has many spikes during the step change in the desired speed and a small oscillation can also be observed. The overall functioning of the controller has kept the speed of the DC motor within the desired value.

Figure (13-c) shows the error between the desired speed output and the actual speed output of the DC motor which was very small in the transient and had a small oscillation figure in steady state equal to $\pm 0.4(rad/\text{sec})$.

6-Conclusions

The real time speed control system consisted of a DC servo motor with a tacho generator that senses motor's speed, and a circuit board that contains data acquisition system components and interface circuitry with an IBM compatible PC along with the motor drive circuits.

The speed of the DC motor has been controlled by varying the voltage input to the DC motor from 0 to 18 volt using digital control action from the personal computer.

Both the real time PID controller implemented in LabVIEW platform and the simulation PID controller using Matlab package were successful in maintaining the desired speed for the dc servo motor.

In the simulation package, the trial runs with different PID parameters were carried out several times in order to find the best system response. And then the best PID control parameters were chosen. These parameters were applied to the PID controller in the discrete time "Real Time Computer Controller" after converting the continuous PID controller parameters to discrete PID controller parameters.

The test results obtained from the real time control action had more oscillation than the simulation control action. The steady-state error in real time was equal to (± 0.4) rad/sec, while that obtained in the simulation was equal to zero. That was because in the real time there were accumulation errors such as those resulting from undesirable characteristics of the speed sensor (tacho generator) "non-linearity, drift, and offset". Other sources that have significant contribution to the resulting steady state error are the offset in the operational amplifier output, and the quantization error of the analog to digital and digital to analogue converters.

7-References

[1] B.K. Bose, "Power Electronics and Motor Drives-Recent Technology Advances" IEEE International Symposium on Industrial Electronics, ISIE, 2002, pp. 22-25.

[2] L. Eduardo and G. Torres "Simulation of a Neural Net Controller for Motor Drives". Mathematics and Computers in Simulation. vol.38, 1995, pp. 311-322.

[3] G.A.Rovithakis and M.A. Christodoulou "Direct Adaptive Regulation Using Dynamic Neural Networks: Application to DC Motors Speed Control", Mathematics and Computers in Simulation. vol.41, 1996, pp. 53-62.

[4] J.Stantana, J.L. Naredo, and F. Sandoval "Simulation and Construction of a Speed Control for a DC Series Motor" Mechatronics, vol. 12, Elsevier Science Ltd 2002, pp. 1145-1156.

[5] E.J. Adam and E.D.Guestrin "Identification and Robust Control of an Experimental Servo Motor". The Instrumentation, Systems, and Automation Society. vol.41, 2002, pp. 225-234.

[6] Tologay Kara and Ilyas " Nonlinear Modeling and Identification of a DC Motor for Bi-directional Operation with Real Time Experiments" Energy Conversion and Management. vol 45, 2004, pp. 1087-1106.

[7] P. Strwart, D. Glagwin, J. Stewart and R. Cowley "Generator Voltage Stabilization for Series-Hybrid Electric Vehicles". The Instrumentation, Systems, and Automation Society. vol.47, 2008, pp. 222-228.

[8] S. Jeong and S. Sang "Precise Position Synchronous Control of Multi-Axis Servo System" Science Direct-Mechatronics, vol. 18, 2008, pp. 129-140.

[9] T. Hagglund, K. Astrom," Automatic Tuning of PID Controllers" in W.S. Levine (Ed), the Control Handbook, CRC Press, Boca Raton, FL, 1996, pp. 817-826.

[10] K Ogata "Modern Control Engineering" 4th Edition, by Addison- Wesley Publishing Company, Inc. 2003.

[11] C.J. Savant and martin S, Roden and Gordan L. Carpenter "Electronic Circuit Design" the Benjamin/Cummings Publishing Company Inc. 1991.

[12] N.Senthil and V.Sandasnam "Design of low cost universal artificial neuron controller for chopper embedded DC drives" Science Direct Application Soft Computer, vol. 17, 2008, pp. 439-445.

[13] Steven J. Faulkenberry "Introduction to Operational Amplifiers with Linear Circuit Applications"John-Wiley Inc. 1986.

[14] Willis J. Tompkins and John G. Webster "Interfacing Sensors to the IBM PC" Prentice – Hall. Inc 1988.

[15] Pei An "PC Interfacing using Centronic, RS232 and Game Ports" British Library

Cataloguing in Publication Data. 1st edition 1998.

[16] Gene F. and J. David “Feedback Control of Dynamic Systems”. Third Edition, by Addison- Wesley Publishing Company, Inc. 1994.

Table (1) Parameters of the DC servo motor [12]

Real parameter of DC motor	***Values***
Armature Resistance R_a	1.0 Ω
Armature Inductance L_a	45mH
Inertia Constant J	0.093 Nm/(rad/s^2)
Damping Constant B	0.008 Nm/(rad/s)
Torque Constant K_t	0.55 Nm/A
Back emf Constant K_b	0.55 V/(rad/sec)
Speed	1500 rpm or 157 rad/sec
DC supply	18 volt
Armature Current i_a	0.24 A

Table (2) Effects of the controller parameters KP, KI, and KD [16]

Response	**Rise Time**	**Overshoot**	**Settling Time**	**Error S-S**
K_p	Decrease	Increase	-	Decrease
K_i	Decrease	Decrease	Increase	Eliminate
K_d	-	Decrease	Decrease	-

Table (3) PID controller parameters that gave the best system response in the Matlab simulation package

Controller	$\boldsymbol{K_P}$	$\mathbf{K_I}$	$\mathbf{K_D}$
PID	**35**	**2.5**	**3**

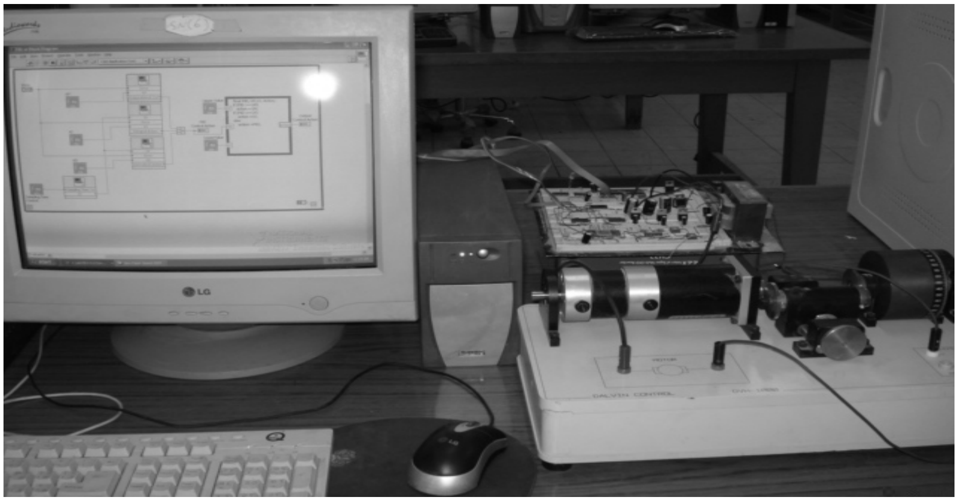


Figure (1) Picture of the experimental setup for the real-time speed control of DC motor.

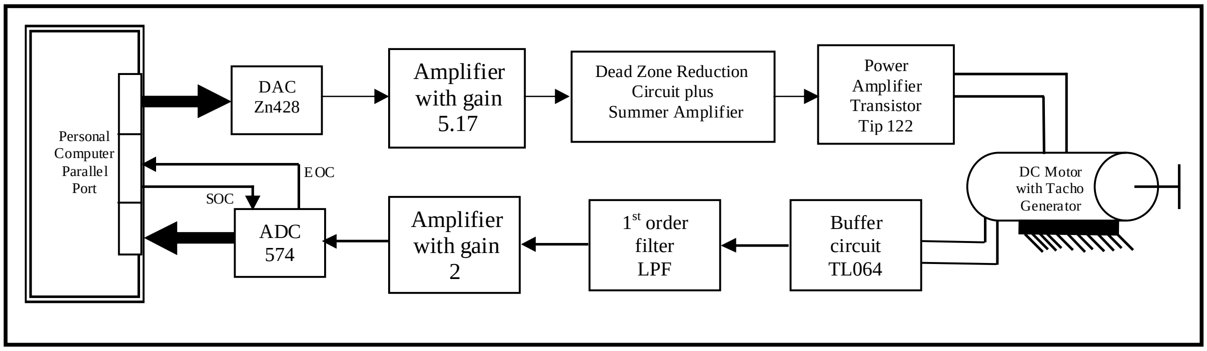


Figure (2) Block Diagram of the experimental of DC motor Speed.

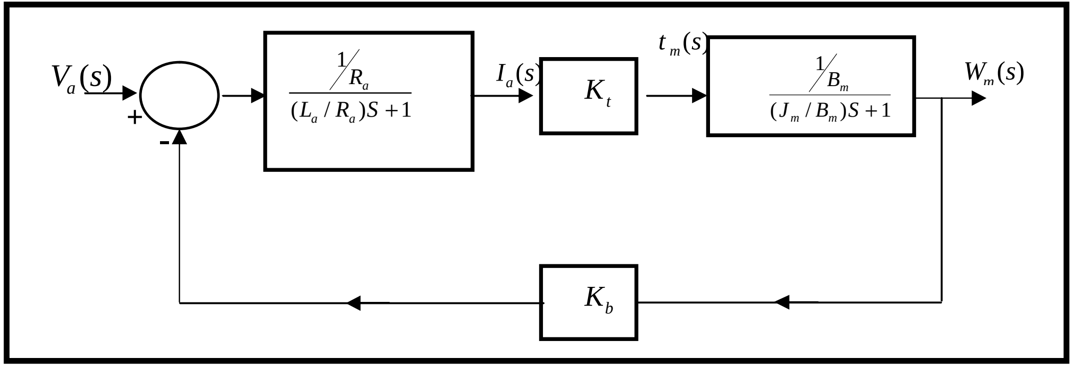
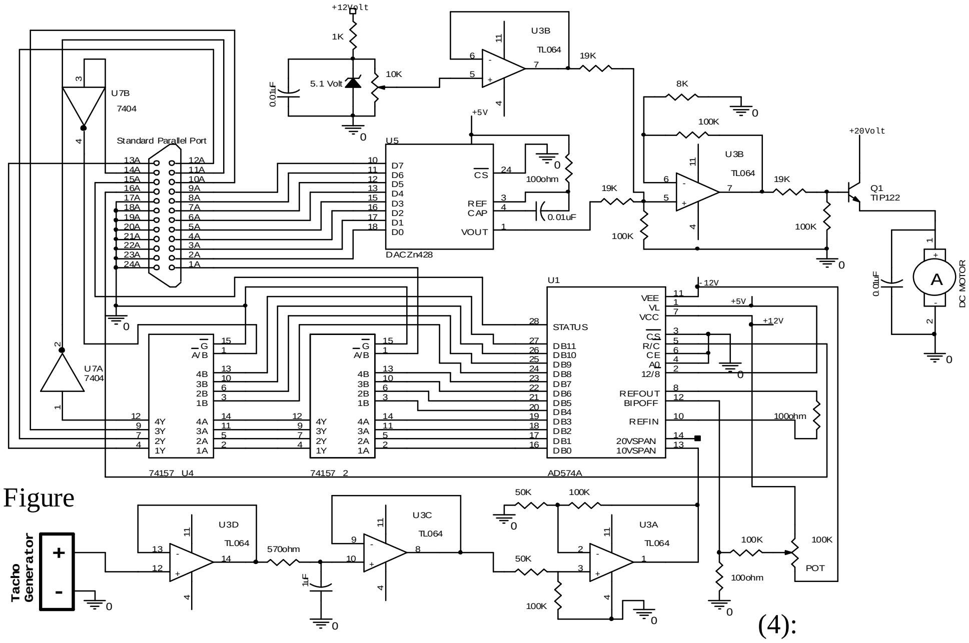


Figure (3) Block diagram of the DC motor model

Where: $K_1 = 1/R_a$ ***and*** $t_1 = L_a/R_a$. $K_2 = 1/B_m$ ***and*** $t_2 = J_m/B_m$.



Figure

(4):

Figure (4) Schematic Diagram of the Digital Speed Control Circuit.

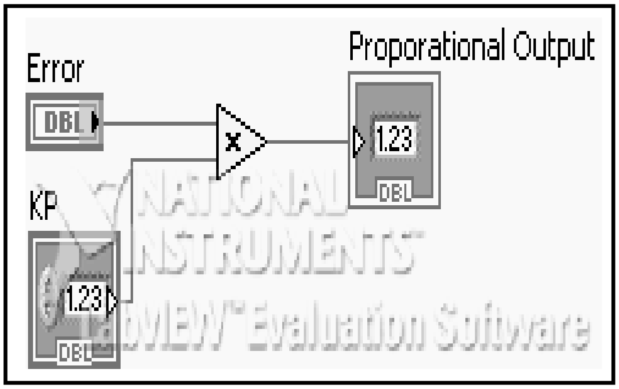


Figure (5) Block Diagram of the Proportional Action Virtual Instrument

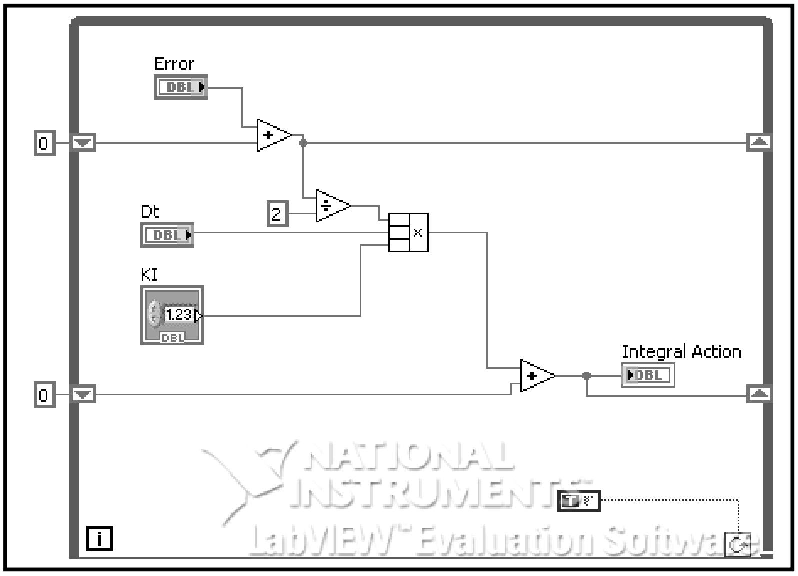


Figure (6) Block Diagram of the Integral Action Virtual Instrument

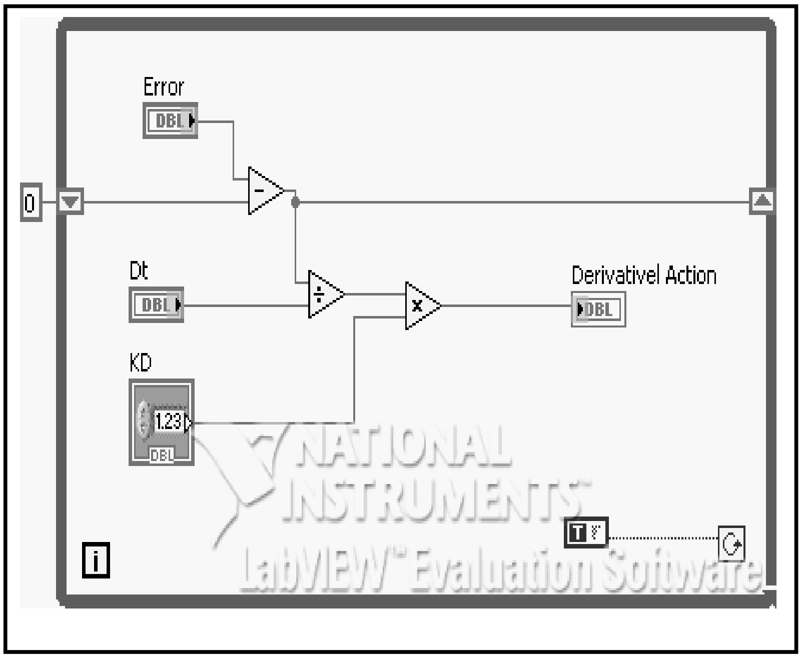


Figure (7) Block Diagram of the Derivative Action Virtual Instrument

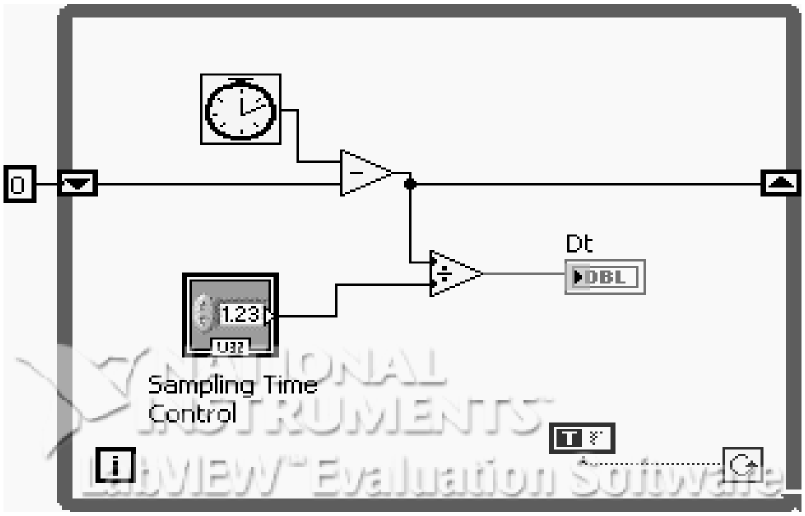


Figure (8): Block Diagram of the Δt Virtual Instrument

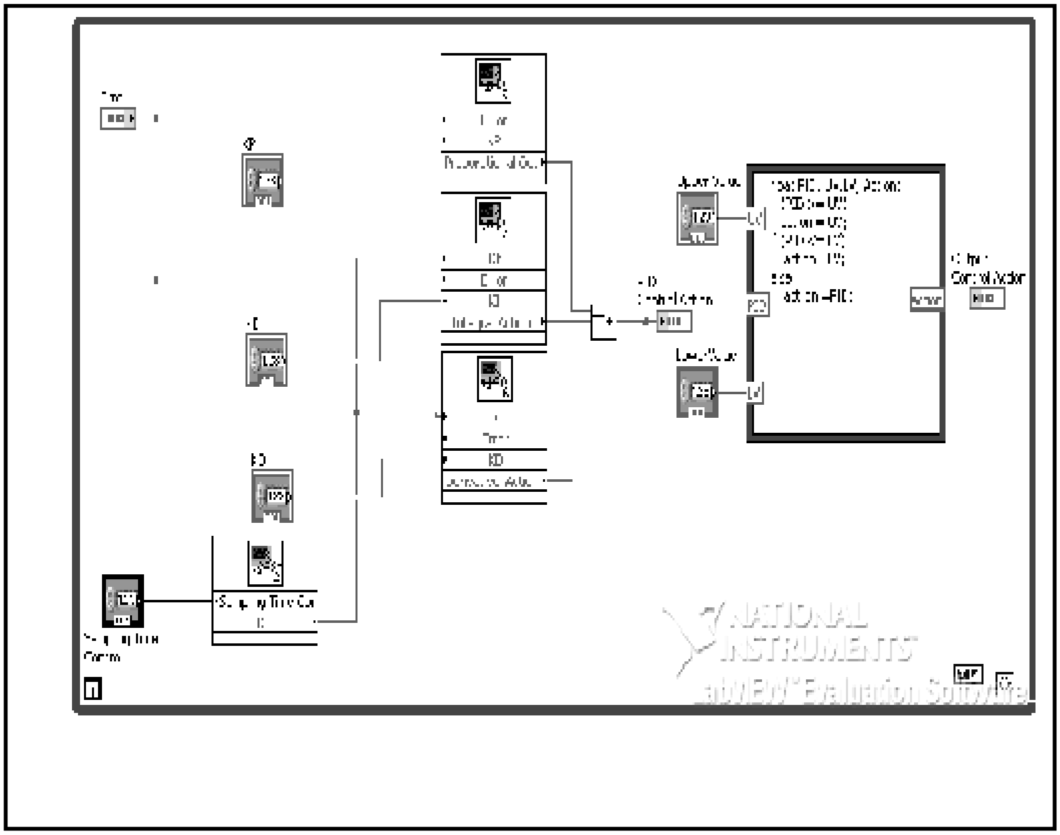


Figure (9) Block Diagram of the PID Virtual Instrument.

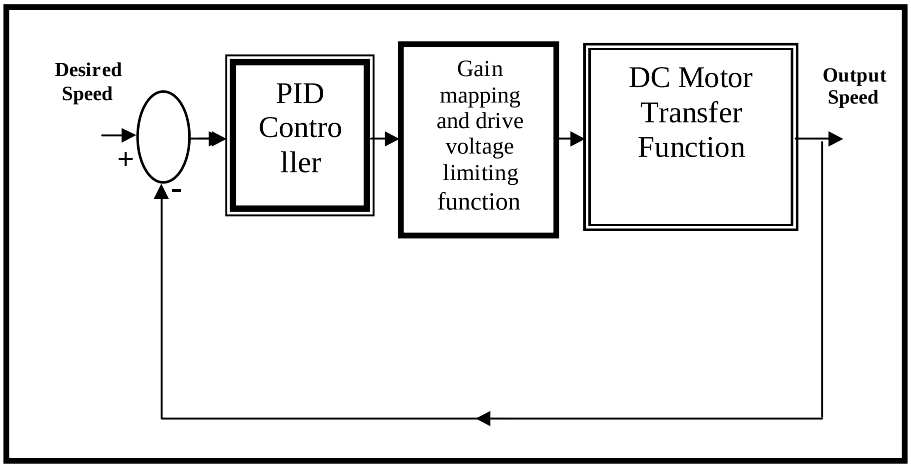

Desired Speed
+
-
PID Contro ller
Gain mapping and drive voltage limiting function
DC Motor Transfer Function
Output Speed

Figure (10) The Mapping Block Diagram between the Real Process and Matlab Simulation.

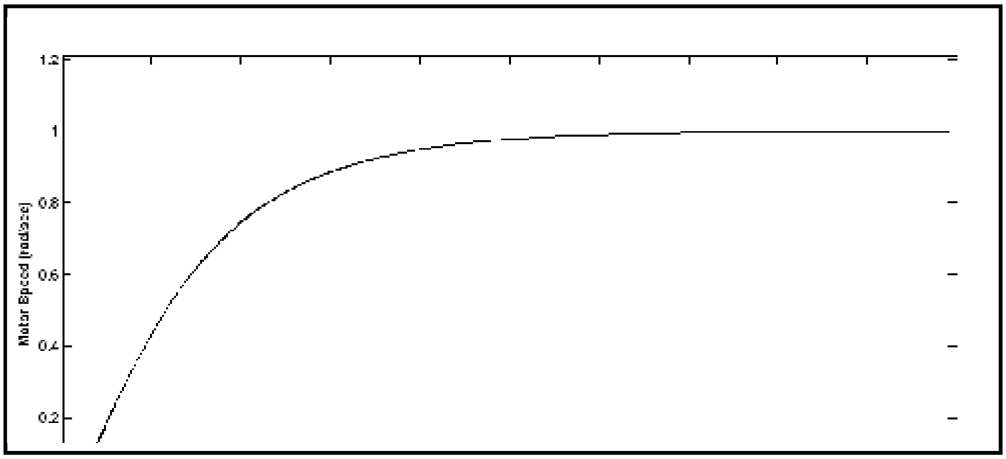


Figure (11) Open loop step response

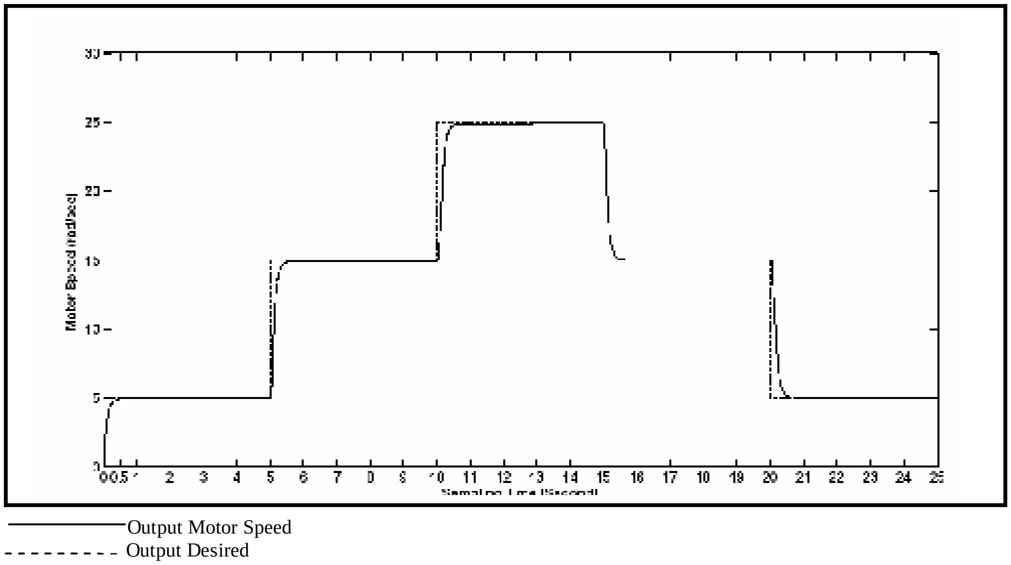


Figure (12-a) The Speed Response of the DC Motor

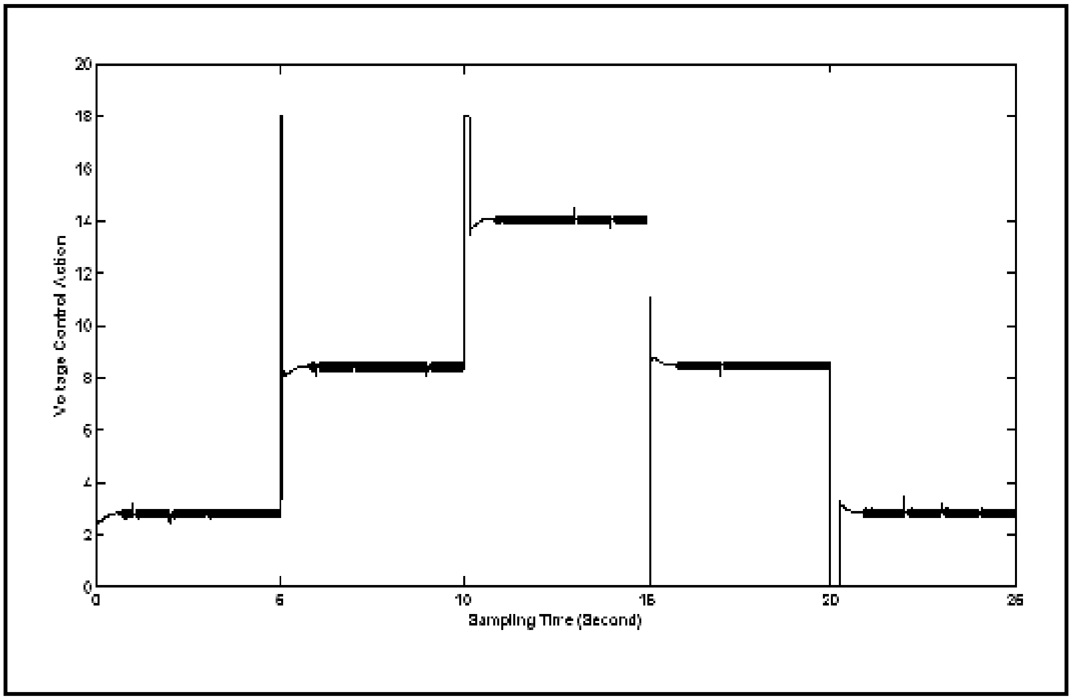


Figure (12-b): The Control Action of the PID Controller

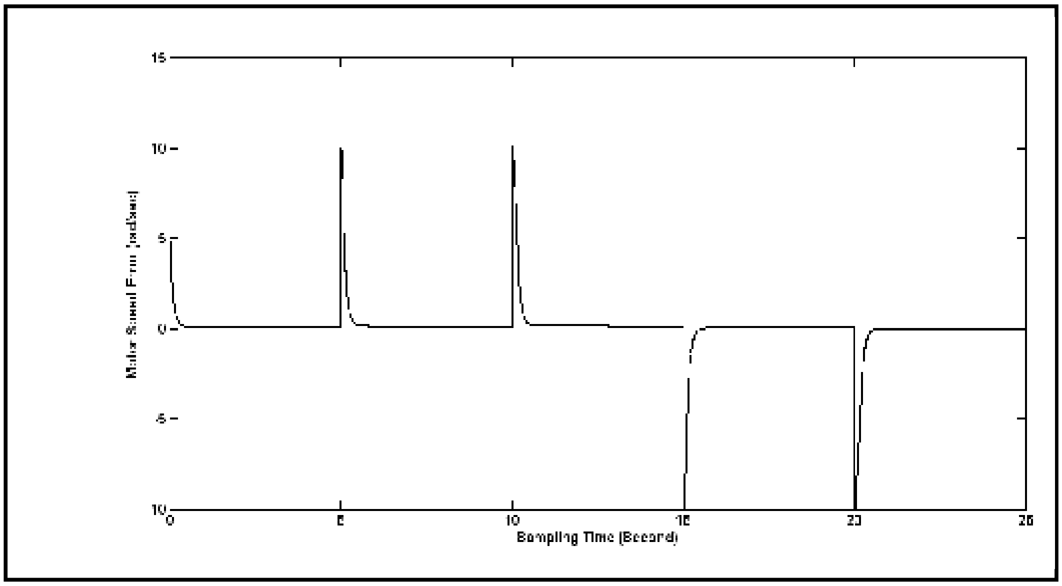


Figure (12-c) Error between the Output Motor Speed & the Desired Speed

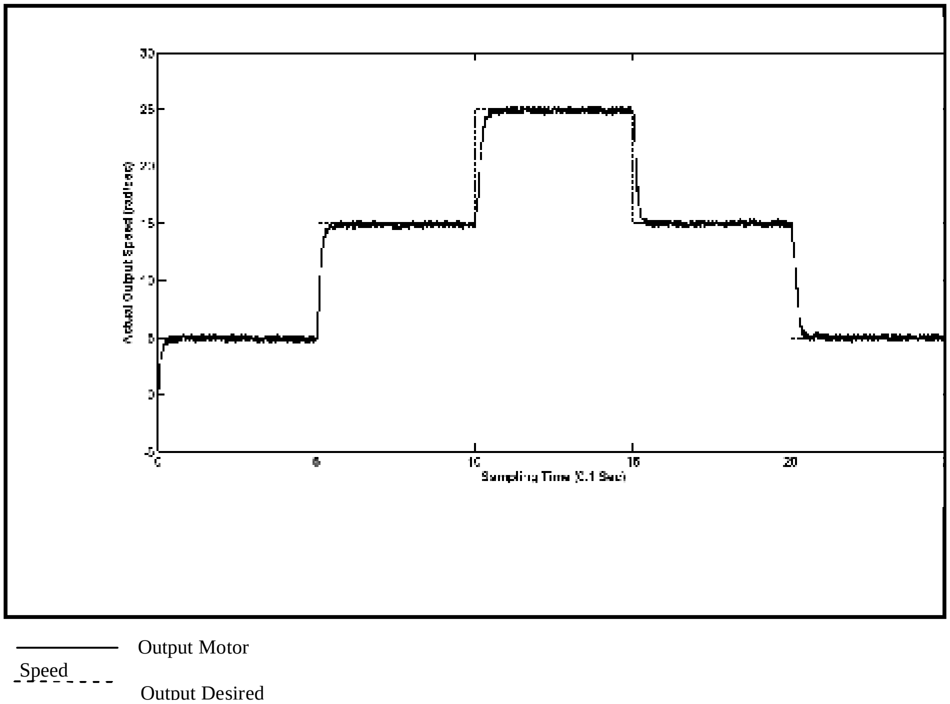


Figure (13-a) The Response of the DC Motor Speed (rad/sec) in Real Time

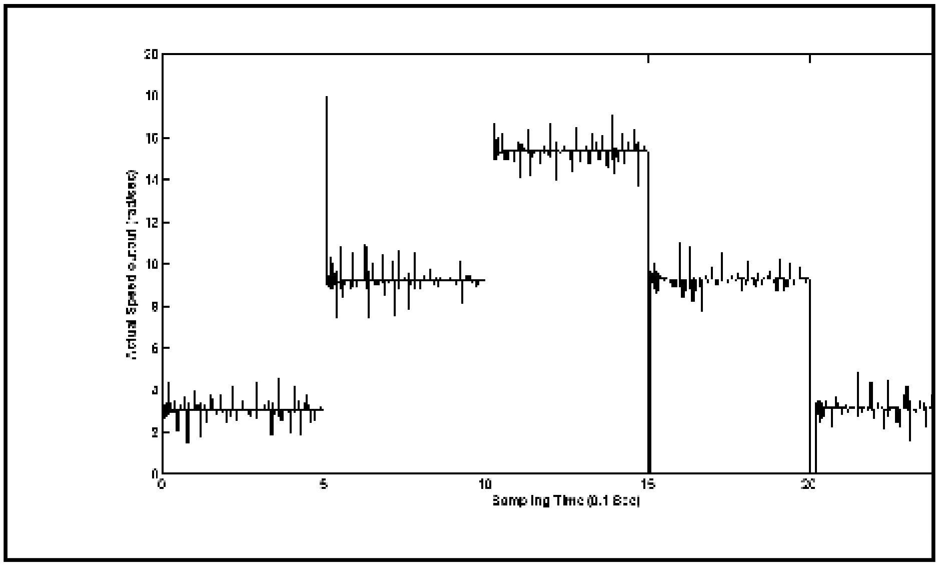


Figure (13-b) The Control Action of the PID Controller in Real Time

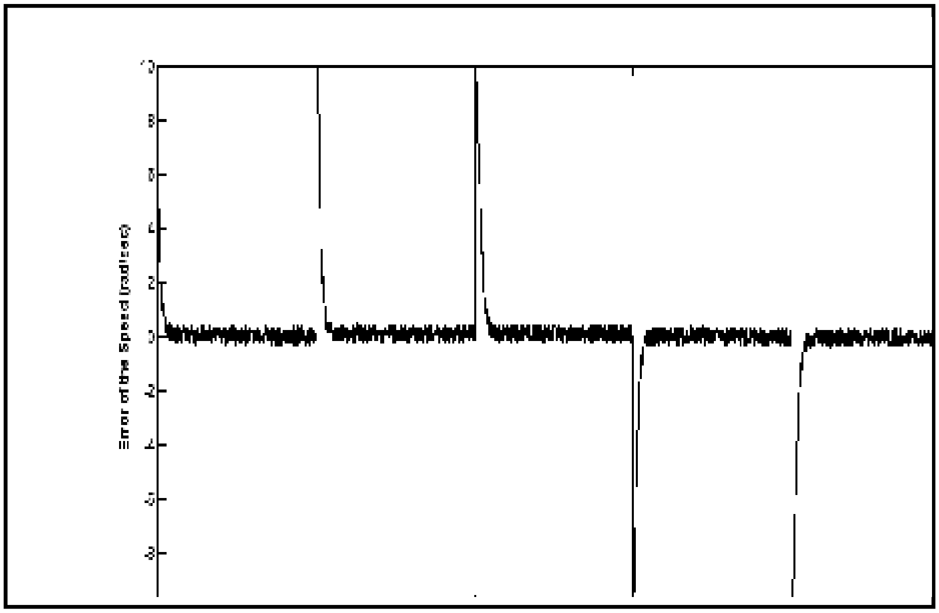


Figure (13-c) The Error between Motor Speed & the Desired Speed in Real Time