Digital PID Control System for DC Servo Motor Using VHDL Code

Jayaraman .K

Assistant Professor, Department Electronics and Instrumentation, Chennai India

Abstract: ***This project investigates based PID motion control systems for small, self- adaptive systems. The closed loop position control of DC servo motor is performed using PWM signal. VHDL based PID motion control system provides an efficient and cheap method. Proportion (P)–increases gain margin, increase system response speed. Integration (I)-minimizes steady state error Differentiation (D)–increases system stability. VHDL: HARDWARE DESCRIPTION LANGUAGES (HDL'S) are used to describe hardware for the purpose of simulation, modeling, testing design and documentation of digital systems. PWM PULSE WIDTH MODULATION which circuit works by making a square wave. Advantage of pulse width modulation is that the pulses reach the full supply voltage and will produce more torque in a motor***

Keywords: component; formatting; style; styling; insert (key words)

1. Introduction

The PID controller provides,
P – Increases gain margin, increase system response speed
I – Minimizes steady state error
D – Increases system stability

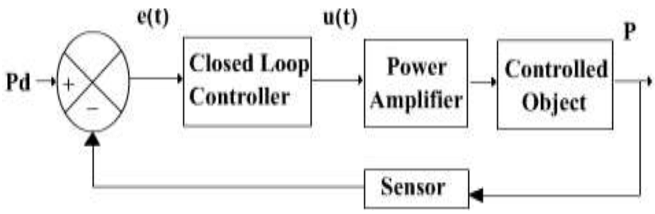
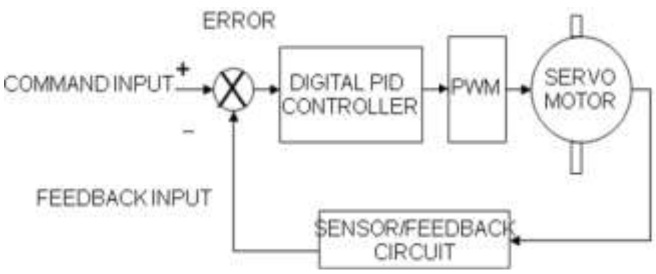


Figure: Closed loop control system

A closed loop control system is shown in figure which is used to control a device such as a servo motor. P and P_d correspond to the controlled variable (e.g. rotational position) and its desired value, which is provided at a higher control level. The goal is to eliminate the error between P and P_d. The value of P is measured by the sensor, which is compared with P_d to generate the error e(t). The output to the controlled device, u(t), from the closed-loop controller is a function of e(t).Typically this is a weak signal that requires amplification.

2. Block Diagram of PID Control System



PID Control Algorithm

- A closed-loop control system is used to control a device such as a servo motor .
- P and Pd correspond to the controlled variable (e.g. rotational position).
- Pd Goal is to eliminate the error between P and.
- Value of P is measured by the sensor, which is compared with Pd to generate the error e(t).
- The output to the controlled device u(t).
- Closed-loop controller is e(t).

$$u(t) = kp[e(n) + 1/Ti\int_0^t e(t)dt + Td * de(t)/dt$$

- For a small **sample interval T,** this above equation can be turned into a difference equation by discretization .

$$u(n) = kp(e(n)) + Ki\sum_{j=0}^{n} e(j) + Kd(e(n) - e(n-1)]$$

- A difference equation can be implemented by a digital system, either in hardware or software.
- The derivative term is simply replaced by a first-order difference expression and the integral by a sum, thus the **difference equation** is given as:

PID Control Algorithm

$$u(t) = kp[e(n) + 1/Ti\int_0^t e(t)dt + Td * de(t)/dt$$

u(t)=The output to the controlled device
Kp = proportional gain
Kd = derivative gain
e = error
KI = integral gain
Ti-integral time constant,
Td-derivative time constant

$$u(n) = kp(e(n)) + KpT/Ti\sum_{j=0}^{n} e(j) + KpTd/T(e(n) - e(n-1)]$$

$$u(n) = kp(e(n)) + Ki\sum_{j=0}^{n} e(j) + Kd(e(n) - e(n-1)]$$

$$u(n-1) = kp(e(n-1)) + Ki\sum_{j=0}^{n-1} e(j) + Kd(e(n-1) - e(n-2)$$

$$\Delta u(n) = u(n) - u(n-1)$$

$$\Delta u(n) = kp(e(n)) + Ki\sum_{j=0}^{n} e(j) + Kd(e(n) - e(n-1)] - kp(e(n-1)) + Ki\sum_{j=0}^{n} e(j) + Kd(e(n-1) - e(n-2)]$$

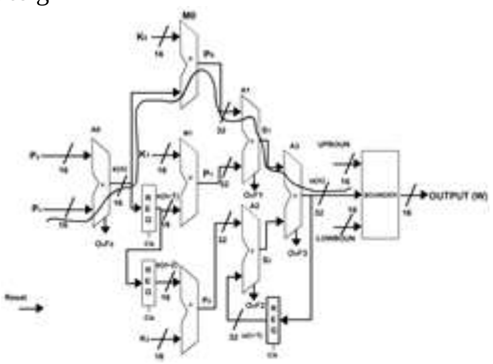
$$\Delta u(n) = kp(e(n) - e(n-1)) + Ki(\sum_{j=0}^{n} e(j) - \sum_{j=0}^{n-1} e(j)) + kd(e(n) - e(n-1) - ((e(n) - e(n-2)))$$

$$\Delta u(n) = kp(e(n) - e(n-1) + kd(e(n) - e(n-1) - e(n-1) + e(n-2)) +$$
$$Ki[e(0) + e(1) + ...e(n)) - (e(0) + e(1) + ...e(n-1))$$
$$\Delta u(n) = Kp(e(n) - e(n-1) + kd(e(n) - 2e(n-1) + e(n-2)) + Ki(e(n))$$

$$\Delta u(n) = Kp(e(n)) - Kpe(n-1) + kd(e(n)) - Kd2e(n-1) + Kde(n-2)) + Ki(e(n))$$
$$u(t) = kp[e(n) + 1/Ti\int_0^t e(t)dt + Td * de(t)/dt$$
$$\Delta u(n) = (Kp + Ki + kd)(e(n) + (-Kp - 2Kd)(e(n-1) + Kde(n-2)$$
$$\Delta u(n) = Ko(e(n) + K1(e(n-1) + K2e(n-2))$$
step input are
$$K0 = (Kp + Ki + kd)$$
$$K1 = (-Kp - 2Kd)$$
$$K2 = Kd$$
$$\Delta u(n) = u(n) - u(n-1)........equation(1)$$
from equvation(1) we find
$$u(n) = \Delta u(n) + u(n-1)$$
$$u(n) = u(n-1) + Ko(e(n) + K1(e(n-1) + K2(n-2))$$

PID DESIGN

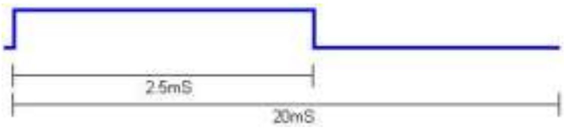
- The design requires 4 adders and 3 multipliers.
- Signal clk is used to control sampling frequency.
- Encoder counter value, represents the current position P.
- The negation of P, P neg, is generated by bit-wise complementing and adding 1.
- Latched at register REG0, thus becomes e(n−1)
- e(n−2)and u(n − 1) are recorded at REG 1and REG2 by latching (n − 1) and u(n) respectively.
- p0,p1,p2,s1 ,s2 are temporary variables.
- *e(n) = pd + (-p)*
- *p0 = k0 * e(n)*
- *p1 = k1* e(n-1)*
- *p2 = k2 * e(n-2)*
- *s1 = p0 + p1*
- s2 = p2 + u(n-1)
- u(n) = s1 + s2
- PID design



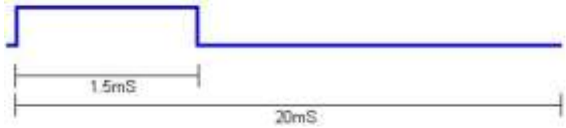
Operation of Servomotor

A typical servo consists of a DC motor, a gear head, a potentiometer for position feedback, and a small circuit to read the pot and position the motor. Position Pulse width

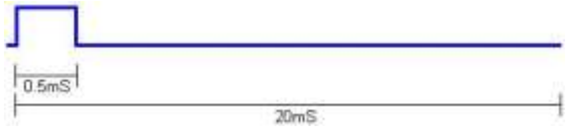
Maximum 2.5 mS



Centre 1.5mS



Minimum 0.5Ms



For example, sending a 1.5 ms pulse to the servo, tells the servo that the desired position is 90 degrees. In order for the servo to hold this position, the command must be sent at about 50Hz or every 20ms.

Once the servo has received the desired position (via the PWM signal) the servo must attempt to match the desired and actual positions. It does this by turning a small, geared motor left or right. If, for example, the desired position is less than the actual position, the servo will turn to the left. On the other hand, if the desired position is greater than the actual position, the servo will turn to the right

Specifications of Servo Motor

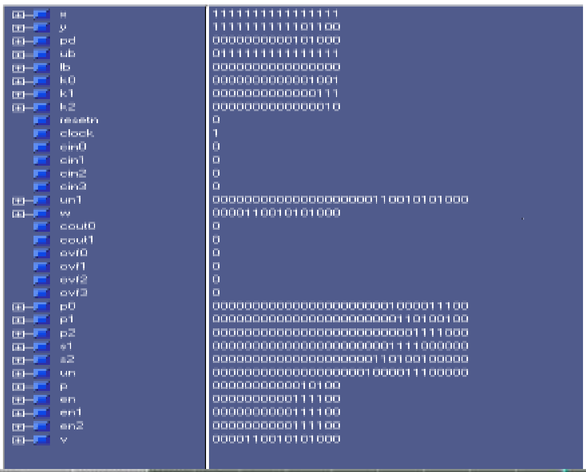
Voltage: 24V DC
Current : 1.0 Amp
Speed: 1500 rpm
Armature Inductance: 2.2mH
Torque: 1.50 N cm
Torque constant: 12.44 N cm/A Moment of Inertia : 0.25 Kg cm^2

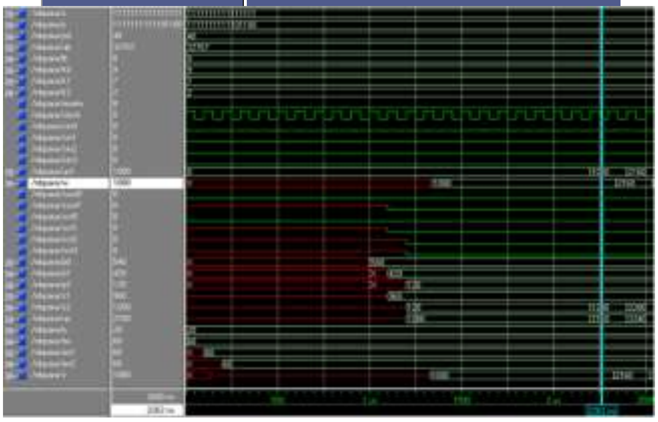
General Equations of Motor

- V=IR+L dl/dt+E
- T= j dw/dt+Bw+Tf
- where, V= voltage in volts,
- T= Torque in N cm,ØØØØ
- I= current in Amperes, J= Moment of Inertia in Kg-cm^2
- R= Resistance in ohms, B= Viscous coefficient of friction in Nm-s/rad,
- L= Inductance in henry, Tf= Load Torque in N cm,
- w=angular velocity in rad/s
- **The** position transfer function obtained by taking laplace transform of above equation is

(Ø)s/ v(s)=6.374/s(12.421s+1)

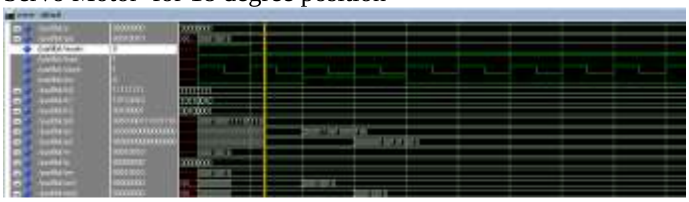
Simulation Output



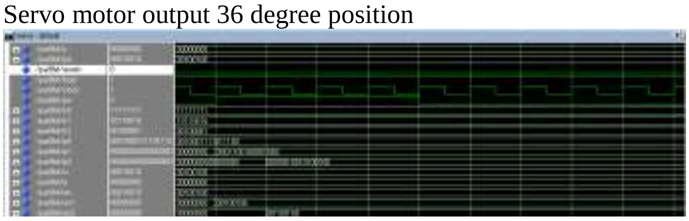


Degree Position	Binary value	Clock pluses
0	00000000	0
18	00010010	02
36	00100100	04
54	00110110	06
72	01001000	08
90	01011010	10
108	01101100	12
126	01111110	14
144	10010000	16
162	10100010	18
180	10110100	20

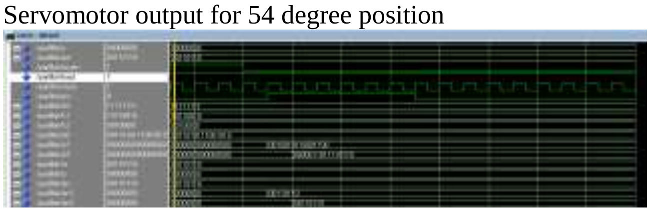
Servo Motor for 18 degree position



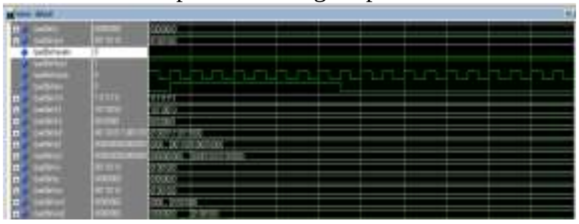
Servo motor output 36 degree position



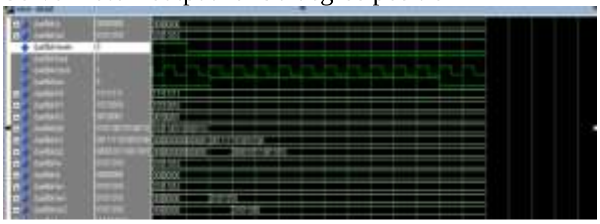
Servomotor output for 54 degree position



Servo motor output for 72 degree position

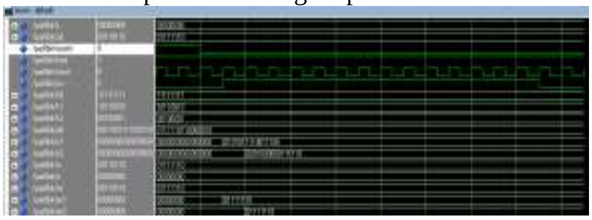


Servo motor output for 90 degree position

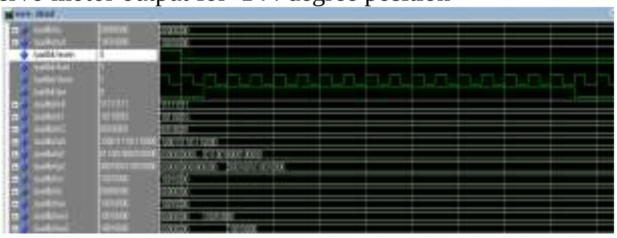


Servo motor output 108 degree position

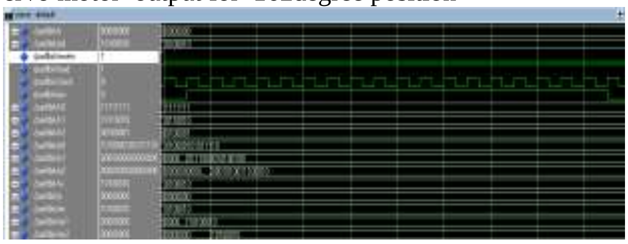
Servo motor output for 126 degree position



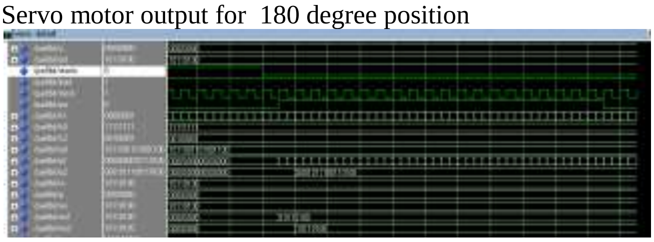
Servo motor output for 144 degree position



Servo motor output for 162degree position



Servo motor output for 180 degree position



References

[1] Wei Zhao, Byung Hwa kim, Amy C. Larson and Richard M. Voyles, "FPGA implementation of closed loop control system for small scale robot" in

Proceedings 12th International conference on advanced robotics - ICAR 05, pages 70–77, 2005

[2] Daijin Kim, "An implementation of fuzzy logic controller on the reconfigurable FPGA system," IEEE Transactions on Industrial Electronics, Vol. 47, No. 3, 2000

[3] Panagiotis Vouzis, Leonidas G. Bleris, Mayuresh V. Kothare and Mark Arnold, "Towards a co-design implementation of a system for model predictive control" in Proceedings, Annual Meeting, American Institute of Chemical Engineers, Cincinnati Convention Center, Cincinnati, OH, November 2005

[4] National Instruments: http://www.ni.com, FPGA based control: Millions of transistors at your command, 2004

[5] Xilinx Comparing and Contrasting FPGA and Microprocessor System Design and Development, July 2004

[6] Wikipedia: http://www.wikipedia.org, Field programmable gate array, 2005

[7] Charles H. Roth Jr., "Digital system design using VHDL" Bross/Cole, 1918