PID Control of DC motor using Particle swarm Optimization (PSO) Algorithm

Mahbubeh Moghaddas[1]**, MohamadReza Dastranj**[2]**, Nemat Changizi**[3]**, Modjtaba Rouhani**[4]

Islamic Azad University, Gonabad Branch, Iran, Moghaddasm.m@gmail.com
Islamic Azad University, Gonabad Branch, Iran, Mohamadrezadastranj@gmail.com
Islamic Azad University Gonabad Branch, Iran, Nemat.changizi@gmail.com
Islamic Azad University Gonabad Branch, Iran, M.rouhani@ieee.org

Abstract

Wide amplitude, DC motor's speed and their facile control cause its great application in industries. Generally the DC motors gain speed by armature voltage control or field control. The suggestion method in this paper is using PSO Algorithm for regulation parameter PID control of DC motors. The Algorithm PSO using by defining the fitness functions so that the minimum error and overshoot design is easy to implement.

Keywords: nonlinear, optimal, classical PID controller, DC motor, PSO Algorithm

1. Introduction

There are variety methods for Inverted Pendulum control that are presented since now. The presented methods for Inverted Pendulum control are divided generally in three groups. Classic methods

---

1, MSC Student Islamic Azad University Gonabad Branch, Iran
2, MSC Student Islamic Azad University Gonabad Branch, Iran
3, MSC Student Islamic Azad University Gonabad Branch, Iran
4,Assistant Professor Islamic Azad University Gonabad Branch, Iran

such as PID and PI controllers [1,2]. Modern methods (adaptation-optimum...) [3, 4, 5] . Artificial methods such as neural networks and fuzzy and Genetic Algorithm and PSO [6, 7, 8] theory are the presented methods for Inverted Pendulum angle control.

The design method in linear control comprise based on main application the wide span ' of frequency, linear controller has a weak application, because it can't compensate the nonlinear system effect completely.

2. Model of DC Motor

The direct current motors are different kinds and several methods are presented for controlling of their speed. In this essay DC motor was chosen for speed control and by controlling the supply voltage was controlled it in nominal less speed.
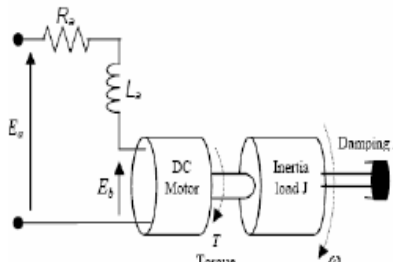The electric circuit of the armature and the free body diagram of the rotor are shown in fig. 1

Figure 1: The structure of a DC motor

$$V_t = R_a I_a + L_a \frac{dI_a}{dt} + E_a \tag{1}$$

$$T = J\frac{d\omega}{dt} + B\omega - T_i \tag{2}$$

$$T = K_T I_a \tag{3}$$

$$E_a = K_a \omega \tag{4}$$

$$\frac{d\omega}{dt} = \varphi \tag{5}$$

With the following physical parameters:

Ea: The input terminal voltage (source), (v);

Eb: The back emf, (v);

Ra: The armature resistance, (ohm);

Ia: The armature current (Amp);

La: The armature inductance, (H);

J: The moment inertial of the motor rotor and load,(Kg.m2/s2);

T: The motor torque, (Nm)

w : The speed of the shaft and the load (angular velocity),(Rad/s);

f : The shaft position, (Rad);

B: The damping ratio of the mechanical system, (Nms);

T k: The torque factor constant, (Nm/Amp);

B k: The motor constant (v-s/rad).

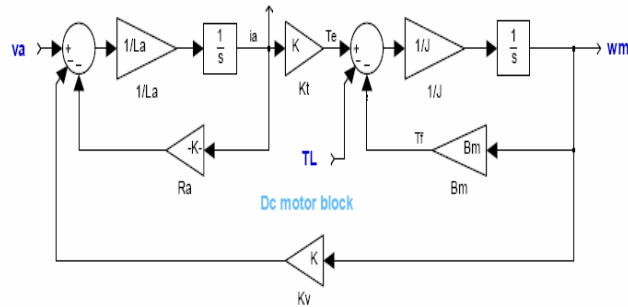Block diagram of a DC motor is shown in fig. 2 [8].

Figure 2.The block diagram of a DC motor

3. Particle swarm Optimization (PSO) Algorithm

Since the introduction of the particle swarm optimizer by James Kennedy and Russ Eberhart in 1995 [9], Numerous variations of the basic algorithm have been developed in the literature. Each researcher seems to have a favorite implementation - different population sizes, different neighborhood sizes, and so forth. In this paper we examine a variety of these choices with the goal of defining a canonical particle swarm optimizer, that is, an off-the shelf algorithm to be used as a good starting point for applying PSO.

The original PSO formulae defined each particle as a potential solution to a problem in D-dimensional space,

With particle i represented Xi=(xi1,xi2,...,xiD). Each particle also maintains a memory of its previous best position, Pi=(pi1,pi2,...,piD), and a velocity along each dimension, represented as Vi=(vi1,vi2,...,viD).At each iteration the P vector of the particle with the best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position (P) is considered the cognition component, and the portion influenced by the best in the neighborhood is the social component [10, 11].

In Kennedy's early versions of the algorithm, these formulae are:

vid = vid+j1*rand()*(pid - xid)+j2*rand()*(pgd - xid)   (6)

xid = xid+vi   (7)

Constants j1 and j2 determine the relative influence of the social and cognition components, and are often both set to the same value to give each component (the cognition and social learning rates) equal weight.

Angeline, in [1], calls this the learning rate. A constant, Vmax, was used to arbitrarily limit the velocities of the particles and improve the resolution of the search.

In [9] Eberhart and Shi show that PSO searches wide areas effectively, but tends to lack local search precision.

Their solution in that paper was to introduce w, an inertia factor that dynamically adjusted the velocity over time, gradually focusing the PSO into a local search:

vid = w*vid+j1*rand()*(pid - xid)+j2*rand()*(pgd - xid)   (8)

More recently, Maurice Clerc has introduced a constriction factor [12] K, that improves PSO's ability to Constrain and control velocities. In [13], Shi and Eberhart found that K, combined with constraints on Vmax Significantly improved the PSO performance. K is computed as:

K = 2

| 2 - j - j 2 - 4j |

Where j=j1+j2,j>4, and the PSO is then:

vid = K(vid+j1*rand()*(pid - xid)+j2*rand()*(pgd - xid))

To test the various parameter settings, we start with the PSO settings Shi and Eberhart used in [13]: 30 particles, j1 and j2 both set to 2.05, Vmax set equal to Xmax, and incorporating Clerc's constriction factor. We assume, in absence of evidence otherwise, that the neighborhood is global, and particles are updated synchronously (That is, g best is determined between iterations).

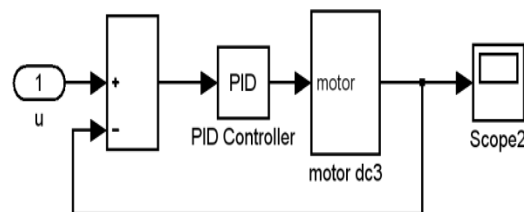At first we control the DC motor by PID controller in fig.3

Figure 3.The block diagram of a PID controller dc motor
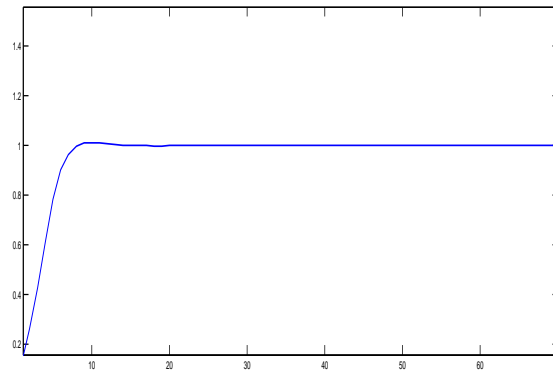
The results are based on fig.4

Figure 4.Simulated results PID controller of DC motor

8. References

[1].Paul I-HaiLin,Santai Hwang and John Chou, "COMPARISON ON FUZZY LOGIC AND PID CONTROLS FOR A DC MOTOR POSITION CONTROLLER" Indiana-Purdo University Fort Wayne.
[2]. J.Tang and R.Chassaing, "PID Controller Using theTMS320C31 DSK for Real-Time DC Motor Control" Proceedings of the 1999 Texas Instruments DSPS Fest, Houston, Texas, August 1999
[3].Y.P. Yang, C.H. Cheung, S.W. Wu, J .P Wang "OPTIMAL DESIGN AND CONTROL OF AXIAL-FLUX BRUSHLESS DC WHEEL MOTOR FOR ELECTRICAL VEHICLES" Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal, July 9-12, 2002.
[4]Hyun Cheol Cho, Kwon Soon Lee and Sami M. Fadali REAL-"TIME ADAPTIVE SPEED CONTROL OF DC MOTORS WITH BOUNDED PERIODIC RANDOM DISTURBANCE" ICIC International °c 2009 ISSN 1349-4198.
 [5].M.Fallahi, Member, IAENG, S.Azadi, Member, IAENG. "Adaptive Control of a DC Motor Using Neural Network Sliding Mode Control" Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong
[6]Jang J. S. R. Adaptive network based fuzzy inference systems. IEEE Transactions on systems man and cybernetics 1993, p 665-685.
[7]Boumediene ALLAOUA*, Abdellah LAOUFI, Brahim GASBAOUI, and Abdessalam
ABDERRAHMANI ''Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization'' Issue 15, July-December 2009
[8]. M.Fallahi, Member, IAENG, S.Azadi, Member, IAENG."Robust Control of DC Motor Using Fuzzy Sliding Mode Control with PID Compensator" Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong
J.Tang and R.Chassaing

[9] Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization, IEEE International Conference on Neural

Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948.

[10] Eberhart, R. and Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory, Proc. Sixth International

Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, 39-43.

[11] Kennedy, J. (1997), the Particle Swarm: Social Adaptation of Knowledge, IEEE International Conference on

Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ, 303-308

[12] Clerc, M. (1999) the swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings, 1999 ICEC, Washington, DC, pp 1951-1957.

[13] Eberhart, R.C., and Shi, Y. (2000), Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, 2000 Congress on Evolutionary Computing, vol. 1, pp. 84-88.