JAYA ALGORITHM BASED TUNING OF PID CONTROLLER

1*ANKIT K. SAHU, 2JAGADISH K. BOKAM, 3V. P. SINGH, 4S. P. SINGH

Dept. of Electrical Engg., National Institute of Technology, Raipur, India
E-mail: 1sanikumar005@gmail.com, 2jagadishjaggu01@gmail.com, vinaymnit@gmail.com, sugandhgcet@gmail.com

Abstract— In this work, Jaya based PID controller tuning is presented for position control of DC servo-motor. The performance index considered is integral of squared error (ISE) of unit step input. The controller settings are obtained by minimizing the ISE using Jaya algorithm. The Jaya algorithm is used due to being simple and free from algorithm specific parameters. The simulation results are obtained for proposed method as well as for other existing techniques. The results show that the Jaya based tuning provides satisfactory response.

Keywords— Integral of squared error, Jaya algorithm, PID controller, Position control, Servo-motor.

I. INTRODUCTION

The DC servo-motors are widely used as actuators for position and motion control of direct drive in many industrial applications like mechanical motion, robotic arm, automation process, etc. The extensive use of these motors is facilitated due to easy adaptability with any control strategy. The classical proportional-integral-derivative (PID) controllers are most preferable ones for various engineering applications due to their easy availability and ease in implementation. Ziegler-Nichols (ZN) tuning rule [1] is one of the most applied methods for tuning of controller parameters. Other than this, integral of squared time weighted error (ISTE), Kessler Landau Voda (KLV), Pessen integral of absolute error (PIAE), no overshoot rule (NOOV), Luus-Tacconis, igler-Nichols (MT-ZN), some overshoot rule (SOOV) and refined Ziegler-Nichols (R-ZN) [2] rules are also well known methods which are used for tuning the PID controllers. Literature survey reveals that the rule based tuning provides only good dynamic response. Hence, various optimization based tuning methods such as PI controller tuning based on teacher-learner-based-optimization (TLBO) [3] Luus-Jaakola based PID controller tuning [4], PID controller tuning using particle swarm optimization [5], PID controller design for three tank system using TLBO [6], PID controller tuning based on genetic algorithm [7], etc., have been proposed in literature for improving the dynamic response. This work proposes Jaya algorithm based PID tuning for level control of three tank system. The Jaya is recently proposed by Rao [8] which is simple to understand and has no algorithm specific parameters. The integral of squared error of step input is minimized for obtaining the set of controller parameters. The simulation results are presented for proposed tuning method and other existing methods. A comparative study is also presented to illustrate the effectiveness of proposed method.

This work is organized as follows: section 2 discusses the DC servo-motor and structure of PID controller, section 3 gives the tuning method, Jaya algorithm is described in section 4, section 5 provides the simulation parameters along with results obtained and the conclusion is given in section 6.

II. CLOSED LOOP CONTROL

The closed loop position control of DC servo-motor is given in Fig. 1.

![Fig. 1. Closed loop system.](image)

The transfer function of DC servo-motor [9] is given as

$$G(s) = \frac{K}{\delta(L_1s + R_1)(J_3s + B_3) + K\delta}$$

where, $K$, $K_1$, $J_0$, $B_0$, $R_0$ and $L_0$ are electromotive force constant, back EMF constant, moment of inertia of motor, viscous friction coefficient, armature resistance and armature inductance, respectively. The $C(s)$ in Fig. 1 is controller. The proportional-integral-derivative (PID) controller considered in this work is given as

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

where, $K_p$ is proportional gain, $T_i$ is integral time constant and $T_d$ is derivative time constant.

Table 1 shows the controller settings [2] due to Ziegler-Nichols (ZN) criterion, Pessen integral of absolute error (PIAE) criterion, no overshoot rule (NOOV) and some overshoot rule (SOOV). The parameters, $K_p$ and $T_i$ represent, respectively, the ultimate gain and the period corresponding to ultimate gain.

<table>
<thead>
<tr>
<th>S.N.</th>
<th>ZN</th>
<th>PIAE</th>
<th>NOOV</th>
<th>SOOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>0.6$K_n$</td>
<td>0.7$K_n$</td>
<td>0.2$K_n$</td>
<td>0.33$K_n$</td>
</tr>
<tr>
<td>$T_i$</td>
<td>0.5$T_u$</td>
<td>0.4$T_u$</td>
<td>0.5$T_u$</td>
<td>0.5$T_u$</td>
</tr>
<tr>
<td>$T_d$</td>
<td>0.125$T_u$</td>
<td>0.15$T_u$</td>
<td>0.33$T_u$</td>
<td>0.33$T_u$</td>
</tr>
</tbody>
</table>
III. TUNING METHOD

The integral of squared error (ISE) of unit step input is considered as tuning criterion in this work. The ISE is given as

$$J = \int_{0}^{\infty} e^2(t) dt$$

which is further re-written in terms of alpha and beta parameters as

$$J = \frac{1}{2} \sum_{i=1}^{n} \alpha_i^2$$

where, $\alpha_i$ and $\beta_i$ for $i = 1, 2, ..., n$ are defined in [10]. The ISE given by (4) is minimized using Jaya algorithm to obtain the controller settings.

IV. JAYA ALGORITHM

Recently, Rao [8] proposed Jaya algorithm. The algorithm is based on the concept that a solution should move towards the best one and should move away from the worst solution.

Suppose, there is a total $R$ solutions in the population considered and the dimension of the problem is $C$. The $j$th dimension, $j = 1, 2, ..., C$ of $i$th solution, $i = 1, 2, ..., R$, can be denoted as $X_{i,j}$. The solutions in Jaya are updated using

$$newX_{i,j} = X_{i,j} + r_j \left( \frac{X_{best,j} - X_{i,j}}{|X_{best,j} - X_{worst,j}|} \right) - r_2 \left( \frac{X_{i,j} - X_{worst,j}}{|X_{i,j} - X_{worst,j}|} \right)$$

where, $newX_{i,j}$ is updated solution and $r_1$ and $r_2$ are random numbers in the range $[0,1]$. $X_{best,j}$ and $X_{worst,j}$ represent, respectively, the best and worst solutions of the population. $|X_{i,j}|$ denotes the absolute value of $X_{i,j}$. If $newX_{i,j}$ is better solution, then its value is updated in $X_{i,j}$ otherwise $X_{i,j}$ is retained. This completes one iteration of Jaya algorithm. This process stops when termination criterion meets.

V. RESULTS AND DISCUSSION

For the system given in (1), the objective function [10] for ISE of step input is given as

$$J = \frac{1}{2} \left[ \left( \frac{B_3}{A_1} \right)^2 \left( \frac{B_3}{A_1} - \alpha_1 A_1 \right)^2 \left( \frac{B_3}{A_1} - \alpha_2 A_1 \right)^2 \left( \frac{B_3}{A_1} - \alpha_3 A_1 \right)^2 \right]$$

where,

$$A_1 = T_1 I_0 J_0$$

$$A_2 = T_1 \left( \frac{L_a B_a - R_a J_0}{K_p K_T} \right)$$

$$A_3 = T_1 \left( \frac{K_a}{K_p K_T} \right)$$

$$A_4 = K_p K_T$$

$$B_1 = T_1 I_0 J_0$$

$$B_2 = T_1 \left( \frac{L_a B_a + R_a J_0}{K_p K_T} \right)$$

$$B_3 = T_1 \left( \frac{B_a R_a - K_p K_T}{K_p K_T} \right)$$

$$B_4 = 0$$

$$\alpha_1 = \frac{A_2}{A_1} - \alpha_1 A_1 \alpha_2 = \frac{A_2}{A_1} - \alpha_2 A_1 \alpha_3 = \frac{A_2}{A_1} - \alpha_3 A_1$$

$$\beta_1 = \frac{B_3}{A_1} - \beta_1 A_1 \beta_2 = \frac{B_3}{A_1} - \beta_2 A_1 \beta_3 = \frac{B_3}{A_1} - \beta_3 A_1 \beta_4 = - \beta_1$$

The parameters of DC servo-motor [9, 10] considered in this work are

$$K_p = 0.01, \quad I_0 = 0.01, \quad I_q = 1,$$

$$L_a = 0.5, \quad J_a = 0.01, \quad B_a = 0.1$$

The values of controller parameters obtained using Jaya algorithm are given in Table II. Table II also provides the setting proposed in [10] and obtained due to ZN, PI, SOOV, and NOOV rules.
Table II: Controller parameters

<table>
<thead>
<tr>
<th>S.N</th>
<th>Proposed</th>
<th>TLBO based PID</th>
<th>ZN</th>
<th>PIAE</th>
<th>NOOV</th>
<th>SOOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>2.1396</td>
<td>990.90</td>
<td>72.07</td>
<td>8.08</td>
<td>24.02</td>
<td>39.63</td>
</tr>
<tr>
<td>$T_i$</td>
<td>-7.3103</td>
<td>127.84</td>
<td>0.70</td>
<td>0.56</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>$T_d$</td>
<td>231.1865</td>
<td>900.90</td>
<td>0.17</td>
<td>0.21</td>
<td>0.46</td>
<td>0.16</td>
</tr>
</tbody>
</table>

The step responses of system with various controller settings given in Table II are shown in Fig. 2. Table III tabulates time domain specifications of responses.

Table III: Time domain specifications

<table>
<thead>
<tr>
<th>S.N</th>
<th>Proposed</th>
<th>TLBO based PID</th>
<th>ZN</th>
<th>PIAE</th>
<th>NOOV</th>
<th>SOOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling time (sec.)</td>
<td>1.22</td>
<td>0.65</td>
<td>3.77</td>
<td>4.56</td>
<td>8.90</td>
<td>5.50</td>
</tr>
<tr>
<td>Peak overshoot (%)</td>
<td>51.84</td>
<td>96.25</td>
<td>0.39</td>
<td>18.20</td>
<td>59.07</td>
<td>27.87</td>
</tr>
<tr>
<td>Peak time (sec.)</td>
<td>0.0998</td>
<td>0.0083</td>
<td>0.8340</td>
<td>0.7312</td>
<td>1.0190</td>
<td>1.1099</td>
</tr>
</tbody>
</table>

CONCLUSION

In this work, a Jaya based PID controller tuning is proposed for position control of DC servo-motor. The controller settings are obtained by minimizing the integral of squared error of step input. The Jaya algorithm is used for minimizing the ISE and it is found that Jaya based PID tuning provides promising results.

REFERENCES