Design and Implementation of Discrete Augmented Ziegler-Nichols PID Controller

Vedika V. Patki, D.N. Sonwane, Deepak D. Ingole and Vihangkumar V. Naik
Department of Instrumentation & Control Engineering, College of Engineering, Pune, India-411 005
Email: patki.vedika@gmail.com
dns.instru@coep.ac.in, dipingole21@gmail.com,
naikvihang@gmail.com

Abstract—Although designing and tuning a proportional-integral-derivative (PID) controller appears to be conceptually intuitive, but it can be hard in practice, if multiple (and often conflicting) objectives such as short transient and high stability are to be achieved. Traditionally Ziegler Nichols is widely accepted PID tuning method but it’s performance is not accepted for systems where precise control is required. To overcome this problem, the online gain updating method Augmented Ziegler-Nichols PID (AZNPID) was proposed, with the amelioration of Ziegler-Nichols PID’s (ZNPID’s) tuning rule. This study is further extension of [1] for making the scheme more generalized. With the help of fourth order Runge-Kutta method, differential equations involved in PID are solved which significantly improves transient performance of AZNPID compared to ZNPID. The proposed augmented ZNPID (AZNPID) is tested on various types of linear processes and shows improved performance over ZNPID. The results of the proposed scheme is validated by simulation and also verified experimentally by implementing on Quanser’s real time servo-based position control system SRV-02.

Index Terms—Discrete PID controller, Ziegler-Nichols’ Auto-tuned PID controller, AZNPID

I. INTRODUCTION

With its three-term functionality covering treatment to both transient and steady-state responses, proportional-integral-derivative (PID) control offers the simplest and yet most efficient solution to many real-world control problems. Since the invention of PID control, in 1910 and the Ziegler–Nichols’ (Z-N) straightforward tuning methods in 1942 [2, 3], the popularity of PID control has grown tremendously. Advances in digital technology, the science of automatic control now offers a wide spectrum of choices for control schemes. However, an extensive survey on the regulatory controllers used in industries reveals that 97% of them are of PID structure [4]. Particularly at lowest level, as no other controllers match the simplicity, clear functionality, applicability, and ease of use offered by the PID controller. Its wide application has stimulated and sustained the development of various PID tuning techniques, sophisticated software packages, and hardware modules.

Though many tuning methods have been proposed for PID controllers so far, none of the method could replace Ziegler–Nichols’ (Z-N) PID tuning rule due to its good initial settings and ease of use[4]. This rule performs satisfactorily for first order system, but they fail to provide acceptable performance for higher order and non-linear systems due to large overshoots and poor load regulation [5].

To overcome these drawbacks many auto-tuning methods are proposed [6-13]. In [6] Mixed Fuzzy- PID control algorithm is applied for digital missile rudder servo system for controlling two closed loops, position loop and speed loop. Paper [7] proposes a GPC (generalized predictive control) implicit PID control algorithm applied to the heat exchanging station control system, which has features of long time delay, time varying and non-linearity. Also in [8] PID is combined with General Predictive Control (GPC) such that the computational tasks of GPC are reduced by taking reciprocal of number instead of inverse of matrix. In [9] PID algorithm is combined with Model Predictive Control (MPC), the restricted objective function is altered to PID form. Through feed forward action ameliorative dynamic of control is achieved in [10]. A comparative study of three PID algorithms i.e. Time Constrain, Observer-based algorithm and Heuristic algorithm is done for Send-on-Delta Sampling in [11]. Based on the fractional-order PID control algorithm and dynamic matrix control (DMC) algorithm, the fractional order PID dynamic matrix control (FOPID-DMC) algorithm is proposed in [12], for better dynamic performance and stability. In [13] state of art of PID control is discussed with design, application, performance and future scope of PID.

Most of the times while working with the real time systems, dynamics of the system are not completely known. In such circumstances to deal with variable operating conditions online gain modification scheme is proposed in [4], which simultaneously adjusts proportional, integral and derivative gains depending on the instantaneous error (e) and change in error (Δe) of the controlled variable. The study is further extended in [1] for getting better result’s for AZNPID (Augmented Ziegler-Nichols (ZN) tuned PID controllers) algorithm by application of Runge-Kutta to solve differential equations in PID. As mentioned earlier this paper is extension of above study. In this paper modified AZNPID algorithm is applied to second, third and fourth order processes along with its hardware verification. Stability aspects of processes are also discussed with the help of Nyquist stability criteria.

The rest of the paper is organized as: Section II gives an overview of PID and Auto-tuned PID controller. Section III, illustrates the experimental results for different systems. In Section IV, results for application of AZNPID on QUANSER’s SRV-02 plant are discussed.
II. Designing of Auto-Tuned PID Controller

The Conventional PID is shown in Fig. 1. For hardware implementation, discrete PID controller is considered.

A. Discrete PID Controller

Discrete form of a conventional ZN PID can be described as,

$$u(k) = K_p \left[ e(k) + \frac{t_s}{t_i} e(j) + \frac{t_d}{t_d} [e(k) - e(k - 1)] \right]$$  \hspace{1cm} (1)

Where, $t_s$ is sampling period, $n$ is index and $j$ is denoting to time constant. $K_p$ is proportional gain, $t_i$ is integral time constant or reset time, $t_d$ is derivative time constant and $e$ is the error between the reference and process output $y$ for instance $k$. Discrete time integral gain $K_i$ and derivative gain $K_d$ obtained as,

$$K_i = K_p \frac{t_i}{t_s}$$

$$K_d = K_p \frac{t_d}{t_s}$$

The values of $K_p$, $K_i$ and $K_d$ are decided using any standard rule. For this study, we have used Ziegler-Nichols Tuning rule [14]. The values of critical gain $K_c$ and corresponding time period $P_m$ are determined experimentally or can be calculated using bode plot. Form those values $K_p$, $t_i$ and $t_d$ will be determined as given in Table I.

<table>
<thead>
<tr>
<th>Gain Coefficient</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional Gain ($K_p$)</td>
<td>0.6 $K_c$</td>
</tr>
<tr>
<td>Integral time ($t_i$)</td>
<td>0.5 $P_m$</td>
</tr>
<tr>
<td>Derivative time ($t_d$)</td>
<td>0.125 $P_m$</td>
</tr>
</tbody>
</table>

Values of $K_p$, $K_i$ and $K_d$ are modified depending on value of $\beta(k)$ after every sample instant as,

$$K_p = K_p(1 + k_i \Delta e(k))$$  \hspace{1cm} (6)

$$K_i = K_i(1 + k_i \beta(k))$$  \hspace{1cm} (7)

$$K_d = K_d(1 + k_i \beta(k))$$  \hspace{1cm} (8)

$$u_{auto}(k) = K_p e(k) + K_i \sum_{i=0}^{k} e(i) + K_d \Delta e(k)$$  \hspace{1cm} (9)

where,

$$\Delta e(k) = e(k) - e(k - 1)$$

$$e(j) = e(k) - e(k - 1)$$

In (6) to (9) $k_i$, $k_i$, $k_i$ are three positive constants which gives variation in process value to achieve desired output. For this study, performance indices are chosen as $k_i=1$, $k_i=1$, $k_i=12$. Equation (2) to (5) indicates, when process output is less than set point ($ref^n$), value of $K_p$ is greater than $K_p$ so that the process achieves set point quickly. $K_p$ is greater than $K_p$ so as to avoid oscillations in process output. At the same time $K_p$ is less than $K_p$ to correct offset errors, which plays a minor role when the process is reaching to set point. As the process reaches to set point, the values of $K_p$, $K_i$, $K_d$ increases, this eventually increases control $u$ ($u_{auto} > u$). This brings back process to desired set point.

B. Augmented Ziegler-Nichols PID Controller

Fig. 2 describes the basic structure of AZNPID. We get initial values of $K_p$, $K_i$, and $K_d$ from Table I. The $e(k)$ and $\Delta e(k)$ is calculated as [4]

$$e(k) = ref^n - y(k)$$  \hspace{1cm} (2)

$$e_N(k) = \frac{e(k)}{abs(ref^n)}$$  \hspace{1cm} (3)

$$\Delta e_N(k) = e_N(k) - e_N(k - 1)$$  \hspace{1cm} (4)

$$\beta(k) = e_N(k) \Delta e_N(k)$$  \hspace{1cm} (5)

A. Steps for implementation of AZNPID algorithm:

- **Step 1**: Obtain gain margin and phase margin of system either using bode plot method or Ziegler Nichol’s -Continuous Oscillation Method.
Step 2: In ZN Continuous Oscillation Method, critical gain of system is \( K_c \) and its corresponding period of oscillation is \( P_c \).

Step 3: Applying Zeigler-Nichols formula, the values of \( K_c \), \( K_{m} \), \( K_{d} \) is obtained from table I.

Step 4: Choosing the appropriate values of constants \( K_{m1} \), \( K_{m2} \) and AZNPID is applied to the system using (6) to (9). \( K_{13} \)

III. EXPERIMENTAL RESULTS

The robustness and stability of AZNPID is tested for second order, third order and fourth order systems.

A. Second Order System

The transfer function for second order system, with dead-time is given as

\[
G_P = \frac{e^{-0.2s}}{s(1 + s)}
\]  

(10)

Referring Fig. 3, one can observe that, after application of AZNPID there is considerable reduction in peak overshoot of the second order system. For testing purpose, the noise with 0.1% magnitude and sinusoidal disturbance of amplitude 1 with frequency 1 Hz is applied to the system and response is observed in Fig. 4. Stability of the system is analyzed by using Nyquist plot, for the above system is shown in Fig. 5.

B. Third Order System

Consider a third order non-minimum process,

\[
G_P = \frac{(1 - 0.1s)}{(1 + s)^3}
\]  

(11)

From Fig. 6, one can observe that there is considerable improvement in the transient performance of the system with AZNPID compared to ZNPID. Fig. 7 shows effect of noise and external disturbances for third order system. Fig. 8 shows stability analysis of the system using Nyquist plot. Fig. 12 shows effect of variation of performance indices for the system above.

C. Fourth Order System

Consider the fourth order linear system as

\[
G_P = \frac{(7s + 1)}{(s^4 + 10s^3 + 35s^2 + 50s + 24)}
\]  

(12)

For fourth order system, from Fig. 9, one can observe that, as order of system increases, oscillations of ZNPID also increased but there is significant effect of modified derivative gain \( K_{d} \) in AZNPID. Due to this there is also a considerable change in settling time and peak overshoot. Fig. 10 shows effect of noise and external disturbances for fourth order system. Fig. 11 shows stability analysis of the system using Nyquist plot.

IV. CASE STUDY

To demonstrate the performance of the proposed algorithm, Quanser’s SRV-02 plant is used for experimentation.
Figure 6. Comparison of responses of PID and AZNPID for Third order system

Figure 7. Comparison of responses of PID and AZNPID in presence of a) noise and b) disturbance for Third order system

Figure 8. Nyquist Diagram for Third Order System

The digital servo rig has two parts, namely hardware and software. The hardware units are mechanical units and digital unit. Mechanical unit consists of power amplifier, dc servo motor and tacho-generator, absolute and incremental digital encoders, input and output potentiometers [15, 16]. The digital unit carries ADC and DAC for signal conversion, switching, multiplexing, encoder circuits, and PWM motor drive. With the help of SIMULINK and Embedded MATLAB

Figure 9. Comparison of responses of PID and AZNPID for Fourth order system

Figure 10. Comparison of responses of PID and AZNPID in presence of a) noise and b) disturbance for Fourth order system

Figure 11. Nyquist Diagram for Fourth Order System
Figure 12. Comparison of responses of AZNPID for different values of performance indices for Third Order system function, under MATLAB 7.3 (2009b) control algorithm was implemented.

A. Mathematical Model

In the Quanser’s servo plant module, servo motor is connected to optical encoder through gearbox. Gear ratio can be changed according to requirement of user [15, 16]. Transfer function of servo plant can be written as,

\[ \frac{\theta(s)}{V_m(s)} = \frac{K}{s(s\tau + 1)} \]  

(14)

Where, \( \theta(s) \) is Laplace transform of position of the load shaft, \( V_m(s) \) is Laplace transform of input voltage, \( K \) is the steady-state gain, \( \tau \) is the time constant, \( s \) is the Laplace operator.

State Space representation of Servo Plant was obtained with position and velocity as states as,

\[
\begin{bmatrix}
    \dot{\theta}(s) \\
    \dot{\omega}_m(s)
\end{bmatrix} =
\begin{bmatrix}
    0 & \tau \\
    1 & -1/\tau
\end{bmatrix}
\begin{bmatrix}
    \theta(s) \\
    \omega_m(s)
\end{bmatrix} +
\begin{bmatrix}
    0 \\
    \frac{K}{\tau}
\end{bmatrix} V_m(s)
\]

(15)

\[
y(s) =
\begin{bmatrix}
    1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    \theta(s) \\
    \omega_m(s)
\end{bmatrix}
\]

(16)

Where, \( \omega_m(s) \) is Laplace transform of speed of load shaft. By substituting specifications of the servo plant [15, 16] mathematical model was obtained as,

\[ G_p = \frac{210}{s(s + 10)} \]

B. Results

For experimentation, reference square wave was taken from SIMULINK. The experimental setup is shown in Fig. 13. Here, the proposed controller was implemented using SIMULINK programming. With the help of Real Time Workshop (RTW) and Quarc4 Windows Target (RTWT) environment the performance of the proposed auto-tuning scheme has been implemented for achieving the desired position (given by the reference signal) by the dc servo motor.

RTWT communicates with the control program, and interfaces the mechanical unit through the digital board. RTW generates C code using Microsoft C++ Professional from the SIMULNK block diagram and acts as the intermediary for two way data flow from the physical servo system to and from the SIMULNK model. The tuning of the PID controller is done according to ZN relation (proportional, integral, and derivative parameters are given by the manufacturer feedback).

Fig.14, Fig.15 and Fig.16, show the responses and variation of control signals for ZN PID and AZN PID, respectively for both constant set-point and variable set-point (square wave).

One can observe that the proposed scheme supposed to provide significantly improved performances for those systems which produce large overshoots under ZN PID. However, the real servo system does not exhibit such behavior; as a result, we find marginal improvement in the performance of AZNPID over ZN PID as shown in Fig.14, Fig.15 and Fig.16.

Fig. 17 represents output of AZNPID in presence of noise and disturbance. In sub-plot 2, one can observe chattering state due to noise. In sub-plot 3, sine disturbances are applied to AZNPID. In general, PID doesn’t withstand in presence of sinusoidal disturbances, although the plot is showing same property but the amplitude of diversion from reference is very less.
Table II. Comparison Of Responses For PID And AZNPID In Matlab

<table>
<thead>
<tr>
<th>System</th>
<th>Performance Parameters</th>
<th>ZNPID</th>
<th>AZNPID</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_P = \frac{e^{-s\tau}}{s(1+s)}$</td>
<td>% overshoot</td>
<td>50.8</td>
<td>34.44</td>
</tr>
<tr>
<td></td>
<td>Rise Time (sec)</td>
<td>0.25</td>
<td>0.472</td>
</tr>
<tr>
<td></td>
<td>Settling Time (sec)</td>
<td>6.5</td>
<td>5.2</td>
</tr>
<tr>
<td>$G_P = \frac{1-0.1s}{(1+s)^2}$</td>
<td>% overshoot</td>
<td>36.5</td>
<td>24.9</td>
</tr>
<tr>
<td></td>
<td>Rise Time (sec)</td>
<td>1.416</td>
<td>1.221</td>
</tr>
<tr>
<td></td>
<td>Settling Time (sec)</td>
<td>13.52</td>
<td>7.053</td>
</tr>
<tr>
<td>$G_P = \frac{(7s + 1) + 10s^2 + 35s^3 + 50s^4 + 24}{s}$</td>
<td>% overshoot</td>
<td>56.2</td>
<td>11.09</td>
</tr>
<tr>
<td></td>
<td>Rise Time (sec)</td>
<td>0.327</td>
<td>0.346</td>
</tr>
<tr>
<td></td>
<td>Settling Time (sec)</td>
<td>15</td>
<td>6.549</td>
</tr>
</tbody>
</table>


Figure 15. Responses of PID and AZNPID for square wave as reference source: process output

Figure 16. Responses of PID and AZNPID for square wave as reference source: control input

Figure 17. Comparison responses of AZNPID a) response of AZNPID b) in case of noise and c) in case of sine disturbance

Conclusions

In this paper, a simple model-independent, discrete auto-tuned scheme of PID controller is presented. Potency of the algorithm is verified by applying on second order process with dead time, third order non-minimum process and fourth order linear process. Using proposed AZNPID algorithm, it is observed that, the transient performance of the system improved significantly, even if, there is change in the order of the system. The scheme adjusts gain of P, I and D after every sample instant based on gain updating factor beta ($\beta$). Although variation of beta is non-linear, original structure of PID remains intact. Robustness of AZNPID and ZNPID was tested by adding noise and external disturbances. Though AZNPID is able to improve the transient performance, steady state performance remains unaffected. Thus, AZNPID controller can be used where the system has poor transient response (non-minimum phase system, higher order systems).

To test the efficacy of hardware implementation, the algorithm is applied to Quanser’s DC-servomotor position control system (SEV-02). AZNPID intensifies the results in different circumstances. Moreover, the scheme can be apt for embedding on dedicated hardware.

References

Conference on Mechatronics and Automation, August 7 - 10, Beijing, China.


[16] Introduction to QuaRec 2.0 & SRV02 and Instructor manual, Quanser Inc., Markham, ON, Canada, 2010.