DESIGN OF PID CONTROLLER AND STATE OBSERVER FOR FLOW CONTROL SYSTEM USING MATLAB

ABITHA V S, VASUDEVAN JAYSANKAR, DIVYA A, JOHN JOSEPH, SUNIL NAG P V

Department of Electronics and Communication, Amrita Vishwa Vidyapeetham, Coimbatore, India

Abstract - In this paper a scheme for developing PID controller and state observer for a flow control system in MATLAB is proposed. The flow control system is interfaced with MATLAB using OPC toolbox. A digital PID controller is constructed based on position algorithm which enables the user to control the system performance. State space model for the system is developed using System Identification. State observer is constructed in MATLAB by employing the system parameters acquired. The estimation error of the observer should converge to zero to track the system properly.

Keywords- PID controller, System Identification, State space model, State observer

I. INTRODUCTION

An observer is used to estimate the state variables of a dynamic system [5]. It is assumed that when designing feedback control system, the input and output of the system is accessible to the user. These input and output is used for estimating states of the system. A discrete PID controller designed in MATLAB is used to control the system. The algorithm developed brings the process value to the set point given by the user. Process reaction curve is used to time the controller [3]. The mathematical representation of the system is constructed based on the input and output data collected under normal operating conditions. System identification toolbox is utilized in order to select the best model corresponding to the system using prediction-error minimization (PEM) techniques. The system parameters obtained from system identification is used to construct an observer.

The outline of this paper is as follows: Following this introduction is a brief overview about the interfacing procedure used to link the system and PC and PID controller tuning. System is modeled using state space representation and observer development is done.

II. SYSTEM DESCRIPTION

Flow process system consists of supply water tank which is fitted with a pump for water circulation. A DP transmitter is used for detecting the flow by measuring differential pressure across orifice meter. The process parameter (flow) is then provided to the PID controller which adjusts the control valve fitted along the water flow through I / P converter. The process is controlled by measuring the parameter to be controlled and the set point of the system. The difference between the measured signal and the set point is error. The controller performs on-line calculations based on error and other setting parameters and generates an output signal. The output signal drives the final control elements like pneumatic control valve to control the process to the desired set point.

III. INTERFACING

OPC which stands for OLE (Object Linking and Embedding) for Process Control is considered as one of the most popular standard data-connectivity method. It is used for the communication between devices, controllers and client applications. The OPC enables data exchange between two devices without prior knowledge regarding the communication protocol and specifications of OPC are generally non-proprietary in nature. OPC comprises of two components which is OPC client and OPC server. The OPC client is software used to communicate with OPC connectors and OPC server, which is the software application, is connected to a device for translating the data into a standard OPC format and this translation is based on server/client architecture [1]. MATLAB is a high level programming language which enables the user to compute the respective tasks. It is used for designing, analysing and simulating the advanced control techniques. OPC Toolbox in MATLAB is used to interact with OPC servers and to enable read, write and log OPC data from devices to control systems. OPC Read enables
user to choose items from the OPC server and to read online data and OPC Write enables us to write the data directly into the OPC server. Thus MATLAB with OPC Toolbox can be used in process industries for data analysing, visualizing, simulating and prototyping of algorithms on real processes [2].

Flow control system uses YOKOGAWA-UT32A controller. OPC Toolbox is used by the user to interface the PID controller with PC where PID controller overwrites the function of YOKOGAWA controller by using Open Serial Demo Configuration server by allocating a specific port and its conversions (depending on the system). A GUI is developed in MATLAB for the system.

IV. PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROLLER

The PID controller is one of the most widely employed feedback control mechanism used for industrial application. The method involved in finding the optimum mode gains for the system is based on the dynamics and complexity of the system, this is known as loop tuning. PID parameters such as proportional band, integral time and derivative time is selected in order to stabilize the system to the desired set point with minimum disturbance. Prior knowledge of the system model is not available in many cases therefore developing of mathematical model for these systems are difficult. A set of rules is proposed Cohen Coon for PID tuning which do not require mathematical model this is the open loop method. Non interacting position PID algorithm is used. The controller is considered to be non-interacting as the derivative and integral modes of operation is independent of each other. Integral and derivative modes work parallel to each other to provide the Non interactive action. The equation describing the PID controller is

\[ u(t) = K_c e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \]  

(1)

Where \( T_D \) is the derivative time [3]

A. PID CONTROLLER TUNING

Tuning of PID controller is done using Process Reaction Curve method

Fig. 2 Theoretical process reaction curve for estimation parameters using two point method [3]

Fig. 3 Process reaction curve for estimation of PID parameters

Fig 3 shows the Process reaction curve obtained for flow control system on using two point method the following

PID parameters are determined. [3]

\[ \tau = \frac{1}{3} (t_2 - t_1) \] 

(3)

Where \( \tau = \) effective process dead-time

\[ \tau = \text{effective process time constant} \] 

\[ t_1 = \text{time at which } \Delta y(t) = 0.283 \Delta y_{ss} \] 

\[ t_2 = \text{time at which } \Delta y(t) = 0.632 \Delta y_{ss} \] 

\[ \Delta u = \text{Step change in process output} \] 

\[ \Delta y_{ss} = \text{Steady-state change in process output} \] 

For the system under consideration we get the following position values

\[ \Delta y_{ss} = 18 \] 

\[ t_1 = 20.094; t_2 = 26.376 \] 

\[ \tau = 0.125; T_D = 2.11 \] (refer Eqn. (4))

Process gain \( K = \frac{\Delta y_{ss}}{\Delta u} = 1.8 \)

By using QDR tuning formulas based on process reaction curve the values of, \( K_c, T_i, T_D \) and \( T_D \) are as follows

Gain, \( K_c = \frac{1.2 \tau}{T_D} = 0.037 \)

Integral time, \( T_I = 2.0 \tau_D = 0.947s \)

Derivative time, \( T_D = 0.5 \tau_D = 1.055 \)

B. NON-INTERACTING POSITION ALGORITHM FOR PID CONTROL

The equation describing the idealized non-interacting PID controller is as follows (refer Eqn. (1))

\[ u(t) = K_c e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \]  

(5)

\[ K_c = \text{controller gain} \]

\[ T_i = \text{integral time} \]

\[ T_d = \text{derivative time.} \]

Equation (5) for small time interval \( T \) is converted into difference equation by discretization. On using the backward-difference approximation on the derivative mode and backward integration rule on the integral mode, we obtain

\[ u(k) - K_c e(k) + \frac{1}{T_i} S(k) + \frac{T_d}{T_s} (e(k)e(k-1)) \]

(6)

\[ S(k) = S(k-1) + T e(k) \]

(7)

\[ u(k) = \text{the controller output at sample k} \]

\[ S(k) = \text{the sum of errors} \]

\[ T = \text{the sampling interval.} \]
Equation (6) is known as the „absolute form” or „position form” of the PID algorithm, because the value of control variable obtained from the error sequence determines the value of the manipulated variable (valve position) [3].

V. GRAPHICAL USER INTERFACE

GUI is a graphical display which enables the user to perform some interactive tasks. Push button, edit text, static text, axes, Radio buttons and check boxes are some of the components that are in build in GUI. Push button generally possesses a call back function where the entire function has to be performed. Edit text displays a string of values with different measurement units. Static texts serves as a label for pop-up menu and modified static texts also helps us to read the selected data. These components can be aligned in a user prescribed way and the properties of the components can also be changed and hence GUI is user-friendly and it leads to an easier access and displaying of data [2]. The entire working of the flow control system can be shown in GUI by creating push buttons, static texts, edit texts and axes. Each component can be differentiated by its tag name. The tag name of each component is accessed in the MATLAB code and the values are showed in the corresponding component. Since input to the control system needs to be given by the user, we use a text box so as to enter set point values in the range of 0 to 100(depending upon the system’s compatibility). Depending upon the algorithm used for PID controller, the system’s performance can be controlled and it can be viewed graphically with the help of axes. Since we need to assure that the flow rate of the system equals the set point flow which in turn gives rise to the output of the controller, the response of set point, flow and controller output has to be plotted in an axes and thereby representing the corresponding values in the respective textboxes. Graph is plotted and from it we can exactly determine the time the system takes to reach a steady state value. We can also use MATLAB logging or workspace to determine the values and can be plotted but only GUI gives a real time graphical representation of system parameters and each graph can be represented separately in each axes or can be represented in one single graph and can be differentiated by using legend and hence graphical representation using time domain parameters can be shown using GUI.

VI. SYSTEM IDENTIFICATION

System identification tool is used to obtain the mathematical model of the system from an observed or measured input and output data procured from the system. System modeling can be performed through two techniques parametric or non –parametric modeling. Parametric modeling involves choosing a model structure and estimating the best fit of the model parameters. In non-parametric technique the model is not known prior to experimentation the model is deduced from the accumulated data [6].

System identification deals with selecting a linear model to represent the system dynamics through A, B, C and D matrices. Grey box modeling is performed on the acquired data to obtain the system structure data is then estimated using state space model by specifying the number of states that are essential to describe the flow control system.

State space modeling method is used for estimation as it considers all initial conditions and it provides provision to observer hidden or intermediate quantities provided the conditions are satisfied. General State space representation of a discrete system is

\[ x(k + 1) = Ax(k) + Bu(k) \]  
\[ y(k) = C x(k) + Du(k) \]

Eqn. (8) is the state space equation and Eqn. (9) is the output equation where \( x \) is a \( n \times 1 \) state vector, \( u \) is a \( m \times 1 \) control vector, \( y \) is a \( q \times 1 \) measurement vector, and \( k \) is the time parameter. The \( A \) matrix is the state matrix, \( B \) the control influence matrix, \( C \) the output influence matrix, and \( D \) the direct or feed through matrix [4].

The flow process is modelled using the dynamics of a first order linear flow control valve [4],

\[ \dot{W}(t) = \frac{W(t)}{\tau} + MU_I(t) \]

Where \( W(t) \) is the Valve Displacement, \( U_I(t) \) is Controller Output, \( \tau \) is Valve Time Constant and \( M \) is Valve flow Coefficient.

The state space representation of control valve is

\[ W(t) = \frac{W_0}{\tau} + \left( \frac{M}{\tau} \right) U_I(t) \]

The state space equation derived from the above state space model is

\[ \dot{W}(t) = \frac{1}{\tau} \left[ W(t) + \left( \frac{M}{\tau} \right) U_I(t) \right] \]

Its corresponding Output equation is [4]

\[ Y(t) = [10]W(t) \]

System identification is performed by imposing the derived structure on the data generated by the system when set at set point 50, the prediction error minimization technique produces a model with best fit of 84.19% and the matrices \( A, B, C \), and \( D \) are estimated to obtain the following values

\[ [A] = [0.9600] \]
\[ [B] = [-0.0357] \]
\[ [C] = [10] \]
\[ [D] = [0] \]

VII. OBSERVER DESIGN

Consider a linear, time invariant discrete-time dynamic system [5]
Design of PID Controller and State Observer for Flow Control System Using MATLAB

\[ X(k + 1) = AX(k) + Bu(k) \]  \hspace{1cm} (14)
\[ y(k) = Cx(k) \]  \hspace{1cm} (15)

Where A, B, and C are the parameters of state space model. The state, the input and output of the system is represented by \( X(k), u(k), y(k) \) respectively. The state observer for the above system \([5]\) is
\[ \hat{X}(k + 1) = A\hat{X}(k) + Bu(k) + L(y(k) - C\hat{X}(k)) \]  \hspace{1cm} (16)

In the above equation \( \hat{X}(k) \) and \( L \) represent the estimate of the state calculated by the observer and observer gain matrix respectively. Order of the system depends on number of states of the system. The aim of observer is to estimate the states of system. Value of states provided by observer and the system model should be approximately equal. If there is any difference between values of observer and system estimates then it means that observer accuracy is low. Another advantage of observer is that we can estimate unknown states which the system model is unable to identify.

The estimation error is
\[ e(k) = x(k) - \hat{x}(k) \]  \hspace{1cm} (17)

The error dynamic equation is
\[ \dot{e}(k) = (A - LC)e(k) \]  \hspace{1cm} (18)

If the observer tracks down the system accurately then \( e(k) \) tends to \( 0 \). Initial state of observer \((\hat{x}(0))\) is assumed to be zero. In the flow control system, except the control valve all other components are represented as a gain. So only the dynamics of control valve is involved. In this scenario, estimation of valve displacement in terms of percentage is done. So the only unknown state to be estimated is valve displacement. The inputs to the observer are system input and system output expressed in terms of deviation variables.

The estimation error comes to zero because Eigen values of \((A-LC)\) matrix is placed on left half of z-plane. The observer gain matrix \((L)\) is estimated using pole placement method. During estimation, input to the observer and system model is same. The input to observer is controller output produced during PID control. If the poles are on left half of z-plane then estimation error can be made zero. This means observer tracks the states of system perfectly.

**CONCLUSION**

The flow control system hardware model is interfaced with MATLAB which enables user direct access to system data. PID controller provides control over the working of flow control system.

The observer constructed tracks the flow value of the flow control system after certain samples for a set point of 50 LPH. This on further development can be used to detect, isolate and identify individual faults that could incur in the system by evaluating the residual generated between the observer and system output.

Fig. 3 shows that observer tracks the real process after certain fixed samples. This happens when the estimation error becomes zero after certain samples. So the observer acts as the real system when the estimation error Eqn. (17) is zero.

**REFERENCES**


