

Fuzzy Adaptive PID for Flow Control System based on OPC

R. Manoj Manjunath

Department of EIE
Hindustan University
Chennai, TN, India

S. Janaki Raman

Assistant Professor, Department of EIE
Hindustan University
Chennai, TN, India

ABSTRACT

The fuzzy adaptive PID control algorithm based on OPC (Open Process Control) is designed for the flow process station to improve the control performance better than the conventional PID controller. PID controller works well only if the mathematical model of the system could be computed. Hence it is difficult to implement PID control for variable as well as complicated systems. But Fuzzy logic control doesn't require any precise mathematical model and works good for complex applications also. In this paper, a two input and three output self adapting fuzzy PID controller was designed to control the final control element of the flow process station. S7-300 PLC is connected with the process station. Real time data exchange between the PLC and MATLAB is realized by means of OPC server. Fuzzy logic is developed using fuzzy toolbox available in MATLAB and OPC toolbox helps in fetching data from the OPC server. The proposed method can be used to realize data process and advanced control to improve the quality of the control. New control algorithms created in MATLAB can be checked with real time systems using this method.

General Terms

Fuzzy systems, Algorithms

Keywords

Fuzzy Adaptive PID, MATLAB, OPC

1. INTRODUCTION

Industrial process control systems have many features such as non linear, inertial lag, time delay and time varying so on. Due to this, precise mathematical modeling is not possible. Traditional PID algorithm doesn't holds well for such systems which has disturbances. A new algorithm which can deal with these limitations should be considered. Fuzzy PID has more advantages as compared to PID. It has fast response, small overshoot and good anti-interference ability. The fuzzy controller is a non linear controller, the fuzzy control algorithm are based upon the intuition and experience about the plant to be controlled. Therefore it doesn't rely on the precise mathematical modeling. In this paper a novel method for such system is introduced, named Fuzzy Adaptive PID. This method has the merits of both Fuzzy and PID. The fuzzy PID control of these papers was simulated in simulation platform MATLAB. MATLAB is

important in simulation stages of advanced control system algorithms and process control. Complex models can be simulated easily using them. But the drawback is deploying control system quickly from simulink to real time system is quite complicated. To make them happen OPC servers are used. In this paper Matrikon OPC server was used to establish real time communication between flow process station and S7-300 PLC. Matrikon OPC is compatible with many industrial instruments and it is used here. This OPC server provides data exchange between SCADA, PLC or DCS. When the simulated algorithm is implemented using hardware through other means, it would be complex. The use of this method reduces the hardware complexity that is involved.

Open Process Control (OPC), also known as OLE for Process Control, is a series of seven specifications defined by the OPC Foundation for supporting open connectivity in industrial automation. OPC uses Microsoft® DCOM technology to provide a communication link between OPC servers and OPC clients. OPC has been designed to provide reliable communication of information in a process plant, such as a petrochemical refinery, an automobile assembly line, paper mill and so on.

A two input and three output fuzzy adaptive PID is designed in this paper. This controller is simulated in MATLAB. The simulation results show that the controller's adaptive ability and robustness are better than the conventional PID controller. The designed controller in MATLAB is made to communicate with the flow process station using Matrikon OPC and it acquires real time data.

2. CONTROL SYSTEM OF FLOW PROCESS STATION

The flow process station consists of a reservoir from which the liquid is transferred to the overhead tank by means of a motor. The process variable in this process is flow. The desired flow is set by the user say 500 lph. An orifice meter is employed to measure the flow rate of the liquid passing through the pipe. DPT (Differential pressure transmitter) senses the pressure difference and it is calibrated to provide the correct flow rate. DPT now sends the measured value to the process computer where controller is employed.

After the execution of the fuzzy PID algorithm in MATLAB the control variable is given to the final control element. The objective here is to maintain the desired flow rate. But due to

external disturbances in real time process industries, it's difficult to maintain constantly throughout.

3. FUZZY ADAPTIVE PID CONTROL ALGORITHM

The structure of fuzzy adaptive PID is shown in fig. 1. It consists of two parts, one is the conventional PID controller and the other is fuzzy controller.

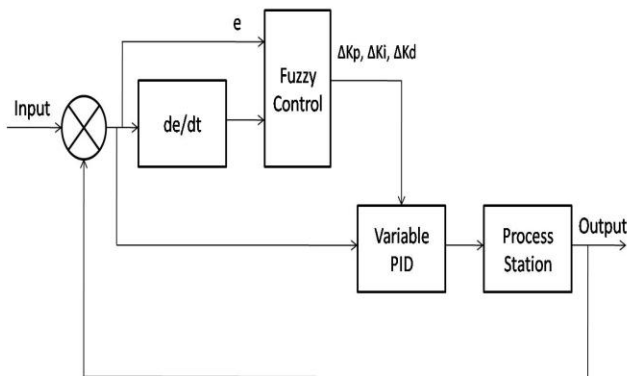


Fig.1 Structure of Fuzzy Adaptive PID

As said earlier, a two input and three output fuzzy adaptive PID is designed in this paper. The inputs are the error and the error rate. The outputs are the K_p , K_i and K_d values.

The objective is to find the fuzzy relations among K_p , K_i , K_d , error, and error rate. With continual testing, the three output parameters are adjusted online so as to meet different requirements and achieve good stability.

Variable PID controller adds the output value of the fuzzy controller and default PID values.

3.1 Design of Fuzzy Adaptive PID

Fuzzy controller is the hardcore of the system. It includes the fuzzification, knowledge base, inference engine and de-fuzzification. Fuzzy controller makes the input accurate quantity to fuzzy quantity. It maps the input to the corresponding discourse. The knowledge base contains the experienced knowledge of the flow process station. Data base contains the membership function of every linguistic variable. Control rules are described by the data base. De-fuzzification again transforms the fuzzy quantity into accurate quantity.

3.2 Membership Function

The membership function used by fuzzy controller is triangular membership function and Gaussian function. The input ranges from -6 to +6 and the fuzzy subset are Negative Big, Negative middle, Negative small, Zero, Positive small, Positive middle and Positive Big respectively termed as NB, NM, NS, ZO, PS, PM, PB.

The quantization factor and the scaling factor play an important role in the performance of the fuzzy controller.

3.3 Control Rules of the Fuzzy Controller

The control rules are framed to achieve the best performance of the fuzzy controller. In this paper 49 control rules are adopted. These rules are given in the table 1, 2 & 3.

Table 1

Kp Fuzzy Control Rule

EC \ E	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

Table 2

Kd Fuzzy Control Rule

EC \ E	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Table 3

Ki Fuzzy Control Rule

EC \ E	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

Using this control rules flow.fis is created. This control rules are framed using the fuzzy logic toolbox available in MATLAB. The above said membership function with the mentioned fuzzy subsets and the control rules form the fuzzy controller. This .fis file is called in the simulink environment and the connection is established between them. The inference engine used here is the Mamdani Inference engine.

3.4 Rule Surface Viewer of the Fuzzy Controller

Based upon the established fuzzy rules, the surface view of K_p , K_i and K_d are shown in figure 2, 3 and 4 respectively

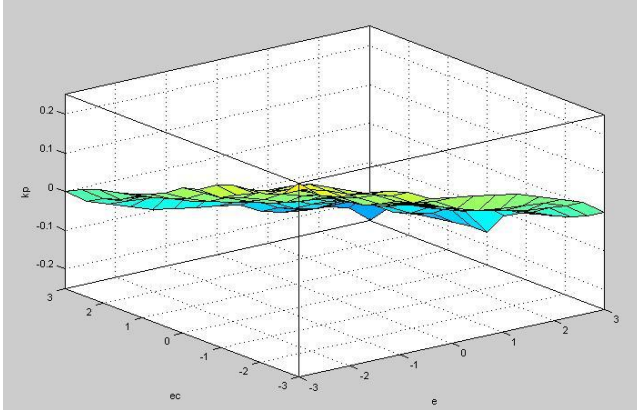


Fig. 2 Surface view of K_p

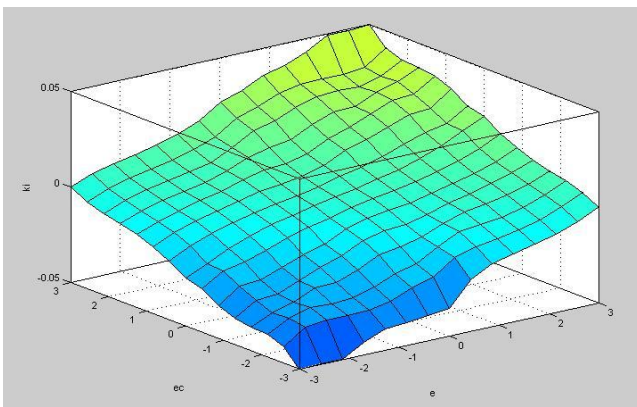


Fig. 3 Surface view of K_i

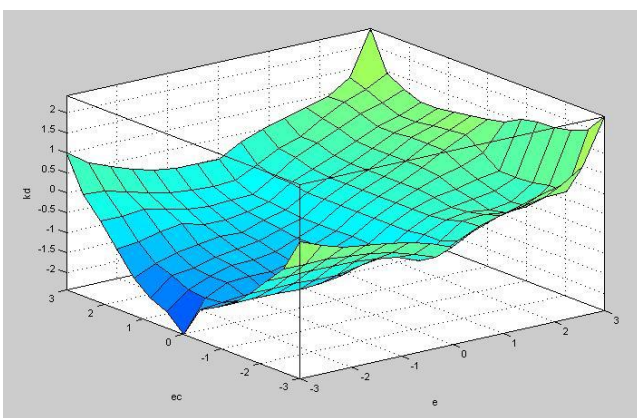


Fig. 4 Surface View of K_d

4. OPC COMMUNICATION

The communication between MATLAB and S7-300 PLC is realized by means of OPC server. OPC toolbox available in MATLAB helps in real time data exchange between OPC server and MATLAB.

Open Process Control (OPC), also known as OLE for Process Control, is a series of seven specifications defined by the OPC Foundation for supporting open connectivity in industrial automation. OPC uses Microsoft® DCOM technology to provide a communication link between OPC servers and OPC clients.

Matlab 7.0 and its above editions integrate the OPC Toolbox to facilitate interoperability with other softwares which is used as an OPC server. By utilizing the OPC Toolbox, we can create the OPC customer application programming easily to realize the communication between Matlab and OPC server and realize fast raw data analysis, measure and control. OPC object hierarchy is shown in the figure 5

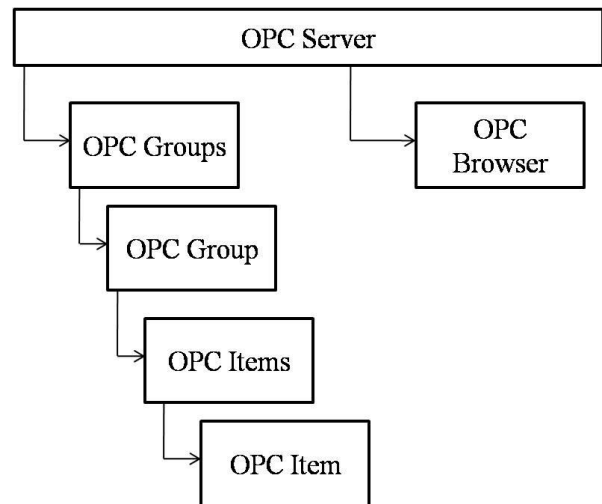


Fig. 5 OPC Object Hierarchy

The above figure is from the reference given by Data Access Interface Standard Version 2.02

4.1 OPC Communication Realization

The communication between MATLAB and S7-300 PLC is realized by means of Matrikon OPC. OPC toolbox available in MATLAB helps to fetch data from the Matrikon OPC.

The basic objects of the OPC Toolbox are OPC data access client objects, Data access group objects and Data access item objects.

The OPC data acquisition procedures are as follows

1. Determining server id
2. Connecting to opc servers
3. Creating a client object
4. Connecting a client to the server
5. Create an opc data access group object
6. Browse the server name space
7. Adding opc data access items to groups and configure
8. Log opc server data

The exchange of data between MATLAB and OPC Server can be performed either synchronously or asynchronously. In the synchronous mode, MATLAB will wait for the operation to complete before continuing with the next one. But asynchronous mode allows MATLAB to continue processing while the operation takes place in the background. The asynchronous mode is used to realize the data communication between MATLAB OPC toolbox and Matrikon OPC server in this paper.

5. ESTABLISH SYSTEM SIMULATION

The fuzzy logic toolbox, OPC toolbox and the MATLAB function are used by the fuzzy adaptive PID control method. Double click the Fuzzy Logic Controller and input the filename of flow.fis established above. To complete the simulation, communication between .fis file and the simulink environment should be established. The following command should be typed in the command window to establish this communication.

```
readfis;  
fismat =readfis ('flow')
```

The fis files are read in the workspace, and the connection is realized between Simulink and the fis files.

A step input is given to the system considered in the simulink environment. The simulation time is for 35s. Finally the system simulation is completed.

6. RESULTS

The simulation results of both conventional PID and Fuzzy Adaptive PID are shown in figure 6 and 7 respectively. The simulation results show that the Fuzzy Adaptive PID has small overshoot and fast response compared to the conventional PID. Moreover this method is feasible due to the effective communication realization between the process device and MATLAB with the help of OPC server.

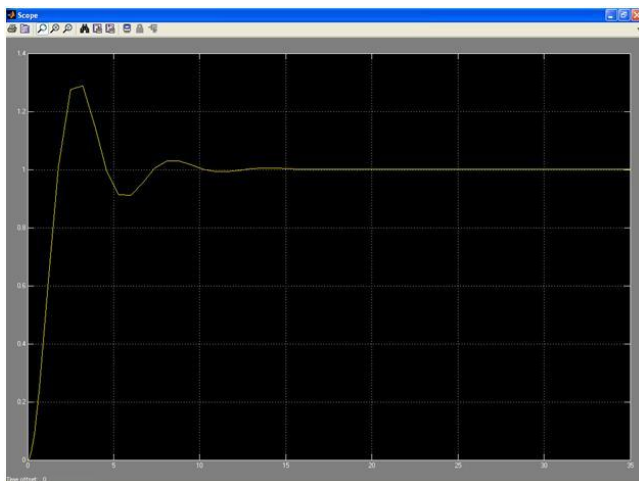


Fig. 6 Step response of Conventional PID

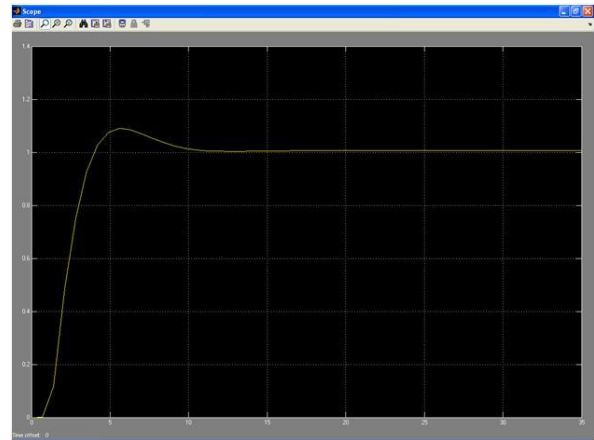


Fig. 7 Step response of Fuzzy Adaptive PID

7. CONCLUSION

A fuzzy self-adapting PID controller is designed for the flow process station. Fuzzy control combined with conventional PID controller constitutes an intelligent control, which adjusts the control parameters depending upon the error. A two input and three output fuzzy adaptive PID control is presented in this paper. The controller is simulated in MATLAB environment. The simulation results show that the fuzzy adaptive PID controller have better stability, small overshoot, fast response. Also communication realization between the process device and MATLAB through OPC server is presented. By this the simulated algorithm can be directly implemented in real time systems.

8. REFERENCES

- [1] OPC Foundation. OPC Data Access Custom Interface Standard Version 2.02[EB/OL](2001-12). <http://www.opcfoundation.org>.
- [2] Sun Xiang, MATLAB 7.0 Basic Tutorial [M]. Beijing, China, Tsinghua University Press, 2005.
- [3] The Math Works Inc. OPC Toolbox For Use with MATLAB [P/OL]. 2004-10, http://www.mathworks.com/access/helpdesk/help/pdf_doc/opc/opc.pdf.
- [4] Vue Wei-jie, Liu Yong-xin (2009), "Boiler Drum Level controlled by Fuzzy Self-adapting PID", Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications.
- [5] Zhang Lieping, Zeng Aiqun, Zhang Yunsheng (2007), "On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology", Proceedings of the 26th Chinese Control Conference.
- [6] Qingbao Huang, Qianzhong She, Xiaofeng Lin (2010), "Adaptive Fuzzy PID Temperature Control System Based on OPC and Modbus/TCP Protocol", 2nd International Asia Conference on Informatics in Control, Automation and Robotics.
- [7] Qingjin Meng, Baoling Xing, Hongliang Yu, Jingjian Wu (2009), "The Application of Intelligent Control to Combustion Control System of CFB Boiler", Ninth International Conference on Hybrid Intelligent Systems.