Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 3 (2015), pp. 1331-1340
© Research India Publications
http://www.ripublication.com

Design and Development of RTDA Controller for a SISO Pressure Process

K.J. Nidhil Wilfred and V.Bagyaveereswaran

AssistantProfessor (SG), SELECT, VIT University
Vellore, TamilNadu, India
nidhilwilfred@gmail.com
SELECT, VIT University, Vellore, TamilNadu, India
bagyaveereswaran@gmail.com

Abstract

In this paper RTDA controller is designed and implemented in real time for pressure process. RTDA controller is a recently developed control strategy for effective control of complex processes. A controller should have three performance features, set point tracking-ability to track changes in set point, Disturbance rejection- property that eliminates effects of sudden disturbances, and Robustness- ability to remain stable even when there is plant model mismatch. Even though the PID controller is simple, its intrinsic structure makes it difficult to tune for satisfying above performance parameters. To overcome the shortcomings of PID controller we have designed a controller with above mentioned performance parameters as tuning parameters.

Keywords: RTDA, robustness, tracking, disturbance rejection, PID.

I. INTRODUCTION

It is a well-known fact that PID controller is the most widely used controller in industrial applications [1]. Tuneable parameters in PID controllers are not directly related to set point tracking, disturbance rejection and robustness. So designing a PID controller to achieve desired performance is a tedious task. It is not possible to tune a PID controller which satisfy set point tracking and disturbance rejection simultaneously. According to a study conducted by Ender on control loops in industries, 30% of the control loops operate in manual mode while more than 60% are not tuned properly [2].The performance of PID controller is poor for dead time dominant and nonlinear processes. In these scenarios we have to use additional components like smith predictor for which implementation is complex[3].Fuzzy controller can be an alternative with better performance, but implementation is more complex. So we need to design a controller that is as simple as a PID controller with better performance[4].

RTDA controller eliminates all the drawbacks of the PID controller and its structure is as simple as that of a PID controller. The tuning parametersθ_R, θ_T, θ_D and θ_A are directly related to performance parameters such as set point tracking, disturbance rejection, robustness and an additional parameter aggressiveness respectively. The values of these parameters are fixed to lie between 0 and 1.

In thispaper section II explains the procedure todesign RTDA controller.Section III deals with the simulation. Section IV covers the hardware setup used for testing the performance of RTDA controller. Section V discusses the experimental procedure for controlling the pressure processfollowed by conclusion in section VI.

II. RTDA CONTROLLER DESIGN

The tuning parameters of RTDA controller is directly linked to desired performance parameters. The following are the steps in controller design [5].

1. Process model: Develop a First Order Plus Delay Time model for the process to be controlled. The continuous time model is given by

$$y(s) = \frac{Ke^{-\alpha s}}{\tau s+1}u(s) \qquad (1)$$

where K, $\alpha$and τrepresents gain, dead time and time constant respectively.

2. Discrete model: Convert the developed continuous model to discrete form and is given by

$$\hat{y}(k+1) = a\hat{y}(k) + bu(k-m)\ldots\ldots\ldots\ldots\ldots\ldots \qquad (2)$$

where m denotes dead time in discrete domain.

3. Uncorrected model prediction: A future prediction for the process output m time steps can be obtained from the following equation

$$\hat{y}(k+m+1) = a^{m+1}\hat{y}(k) + b\mu(k,m) + \ b\eta_i u(k)\ldots\ldots\ldots\ldots \qquad (3)$$

where $\mu(k,m) = \sum_{i=1}^{m} a^i u(k-i)$ and $\eta_i = \frac{1-a^i}{1-a}$for 1<$i$ <N

4. Model prediction update: There will be a mismatch between A FOPDT model and true process. The modelling error is given by

$$e(k) = y(k) - \hat{y}(k)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (4)$$

error can be decomposed into two parts first one due to model process mismatch and the other due to unmeasured disturbances.

$$e(k) = e_m(k) + e_D(k)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$$

$$e_D(k) = \theta_R e_D(k-1) + (1-\theta_R)e(k)\ldots\ldots\ldots\ldots \qquad (6)$$

5. Future disturbance prediction and updated model prediction: Future disturbance effect is given by the equation

$$\widehat{e_D}(k+j) = e_D(k) + \frac{(1-\theta_D)}{\theta_D}\left[1-(1-\theta_D)^j\right]\Delta e_D(k)\ \ldots\ldots\ldots\ldots\ldots\ldots \qquad (7)$$

where

$$\Delta e_D(k) = e_D(k) - e_D(k-1)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (8)$$

with this prediction updated model prediction is given by

$$\tilde{y}(k+m+i) = \hat{y}(k+m+i) + \widehat{e_D}(k+m+i)\ldots \qquad (9)$$

6. Control action formulation: At each point of time the control input is calculated such that the difference between predicted process output and desired trajectory is minimum. The desired trajectory is given by

$$y^*(k+j) = \theta_T^j y^*(k) + \left(1-\theta_T^j\right) y_d(k)\ldots\ldots\ldots\ldots \quad (10)$$

We will get the control input by solving the least square optimisation problem.

$$u(k) = \frac{\sum_{i=1}^{N} \eta_i \psi_i(k)}{b\sum_{i=1}^{N} \psi_i^2}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (11)$$

where

$$\psi_i(k) = y^*(k+i) - a^{m+i}\hat{y}(k) - a^{i-1}b\mu(k,m) - \widehat{e_D}(k+m+i) \quad (12)$$

7. Tuning Parameters: Generally we give $\theta_R = 0.5$.Other parameters depends on process.

$$\theta_D = 1-\theta_R \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (13)$$

$$\theta_T = 10^{\frac{-\Delta t}{\tau}}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (14)$$

$$\theta_A = 1-e^{-\frac{(N-1)\Delta t}{\tau}}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (15)$$

III. SIMULATION

We considered a system with transfer function $(s) = e^{-0.5s}\frac{1}{s^2+s+1}$. The first step in RTDA controller is FOPDT model development. The developed FOPDT model is given by $G_1(s) = e^{-0.5s}\frac{1}{s+1}$. The tuning values based on the model are given by θ_R =0.5, θ_D=0.5, θ_T=0.7943, θ_A=0.0555. The fig. 1 shows the closed loop response with RTDA controller for amodel$G_1(s)$.

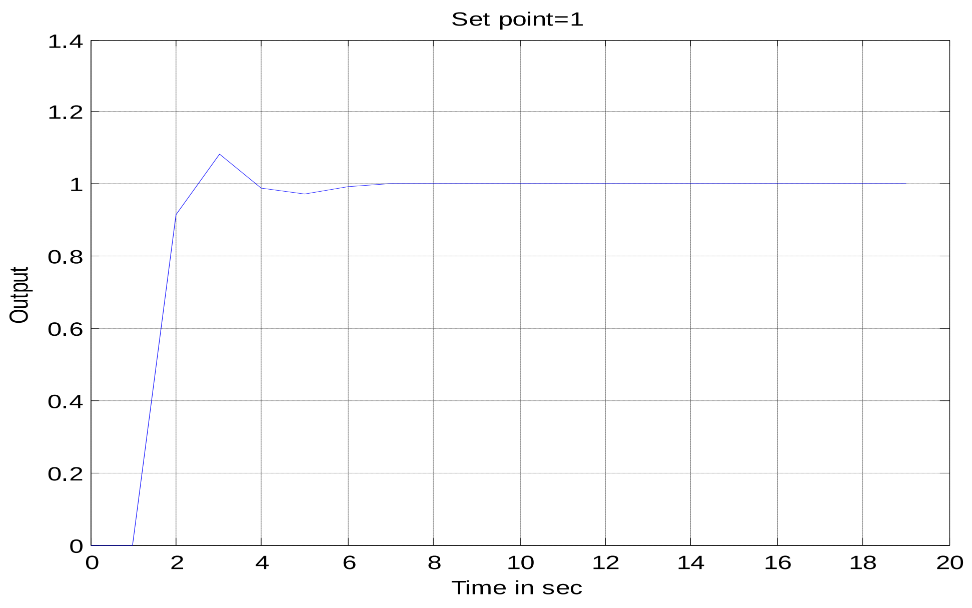

Fig.1Simulared process response with RTDA controller

IV. HARDWARE SETUP

Hardware setup consists of pressure process station, Arduino Uno DAQ and PC. The detailed description is given below

A. Pressure Process station

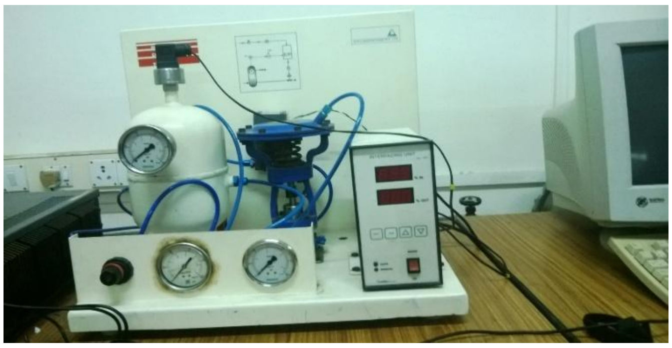

Fig.2Pressure process station.

The pressure system shown in fig.2 is used for the entire work.The process summary can be easily understood from fig.3.The main component is a metallic tank with inlet and outlet. We keep inlet hand valve at a fixed position and there is a pneumatic actuated control valve at the outlet side. Here pressure is the controlled variable and outlet flow rate is the manipulated variable. The pressure inside the tank is measured using a diaphragm based pressure transmitter. It converts the pressure in (0-75) psi range to (4-20) mA current signal.It consists of a primary sensing element that converts pressure to displacement and a secondary element that uses this displacement change for changing resistance. A fixed dc excitationof (12-24) V is needed for transmitter to work properly. The current signal generated by the transmitter is an input for the controller. The controller output would be of (4-20) mA current signal. TheI/P converter gives pneumatic signal output in 3-15 psi range corresponding to controller output.The I/P output operates the control valve at the outlet. Hence controls the pressure inside the tank.

Table 1:Pressure process station specifications

Part name	**Function details**
Process tank	MS
Pressure gauge	0-35 psi
I/P converter	4-20 mA input 3-15 psi output
Air regulator	Size ¼” BSP range 0-2 kg/cm2
Control valve	Size ¼ pneumatic actuated, Input 3-15 psi
Pressure transmitter	Two wire type, Input 0-75 psi, Output 4-20mA DC

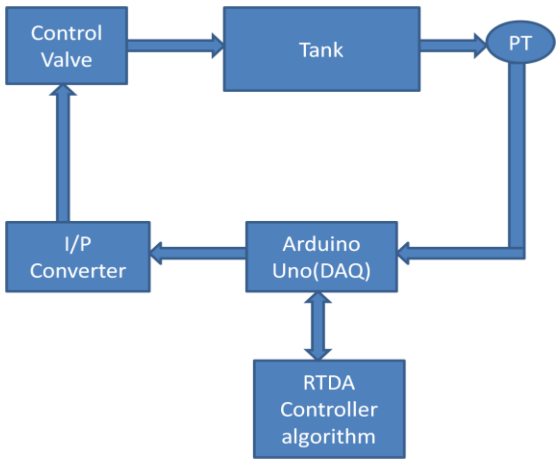

Fig.3Process summary

B. Arduino Uno

Arduino Uno is used as a data acquisition device (DAQ).Arduino Uno Input Port 2receives only voltage signal in the range 0-5 V.The 4-20 mA signal from Pressure Transmitter should be converted to voltage using a 250Ω resistor.ie, 4 x 250 =1V and 20 x250 = 5V.The DAQ Board receives pressure signal in the range (0-75) psi as (1-5)V.Arduino Uno Input Port 3 gives voltage signal in the range (0-5) V.This voltage signal should be converted back to 4-20 mA current signal, which is the input for I/P converter.I/P converter gives the input for the fail to open type pneumatic actuated control valve.

C. MATLAB-Arduino Uno Interface

By using MATLAB-Arduino Support package Arduino can communicate with MATLAB program.MATLAB can access both input & output ports of Arduino Uno using interface commands.

Fig.4 Arduino Uno

V. EXPERIMENTAL VALIDATION

The complete setup for the experimental validation is shown in fig.5.The first step is finding the FOPDT model of the pressure process. Integral equation method was used which requires the step response of the system [6]. Step input was applied to the system and the tank pressure was allowed to settle. The control input and pressure variable values were recorded. Using these values FOPDT model was developed and is given by

$$y(s) = \mathrm{e}^{-.5s}\frac{1.01}{18.42s+1}u(s)\ldots\ldots\ldots\ldots\ldots\ldots \qquad (16)$$

RTDA controller was developed for the pressure process given in eqn. (16).The tuning parameters are given byθ_R=0.5, θ_D=0.5, θ_T=0.955, θ_A=0.00006.

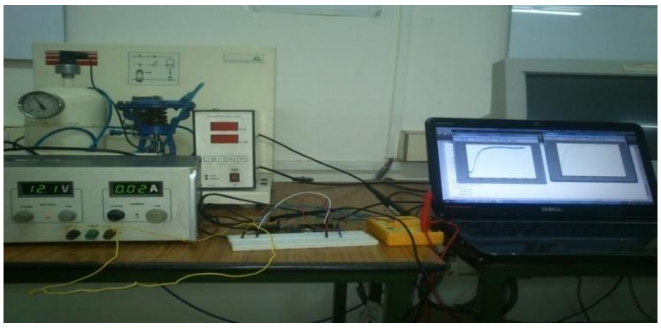

Fig.5Experimental setup

A. Set point tracking

Initial pressure inside the tank is setit as 25 psi. Then a set point of 35 psi was given to the developed RTDA algorithm. The response graph is shown in fig.6.The pressure settled at 35 psi without any overshoot.

B. Disturbance rejection

Similar to the first step pressure was set at 25 psi initially and 35 psi set point was given. The pressure inside the tank settled at 35 psi. Then a disturbance of 10% of current pressure value was introduced at around 270 secs in the vent side which caused pressure to fluctuate. But the controller eliminated its aftereffects and pressure regained its set point. The response graph is shown in fig.7.

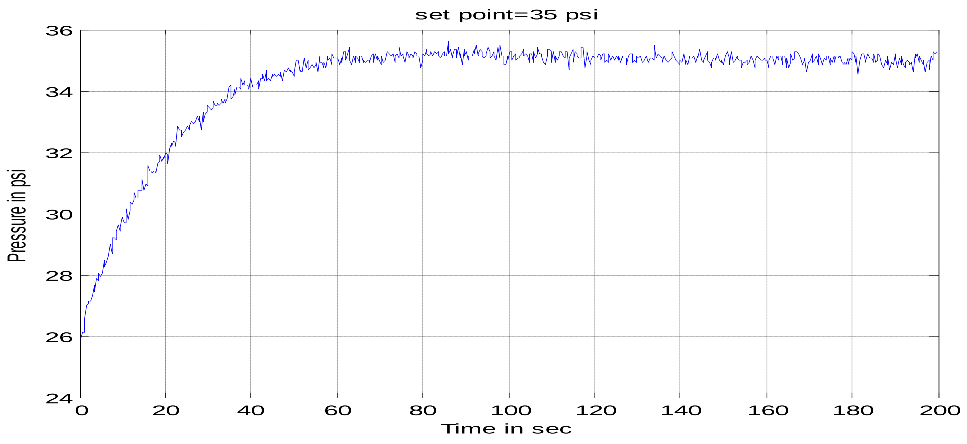

Fig.6set point tracking with RTDA

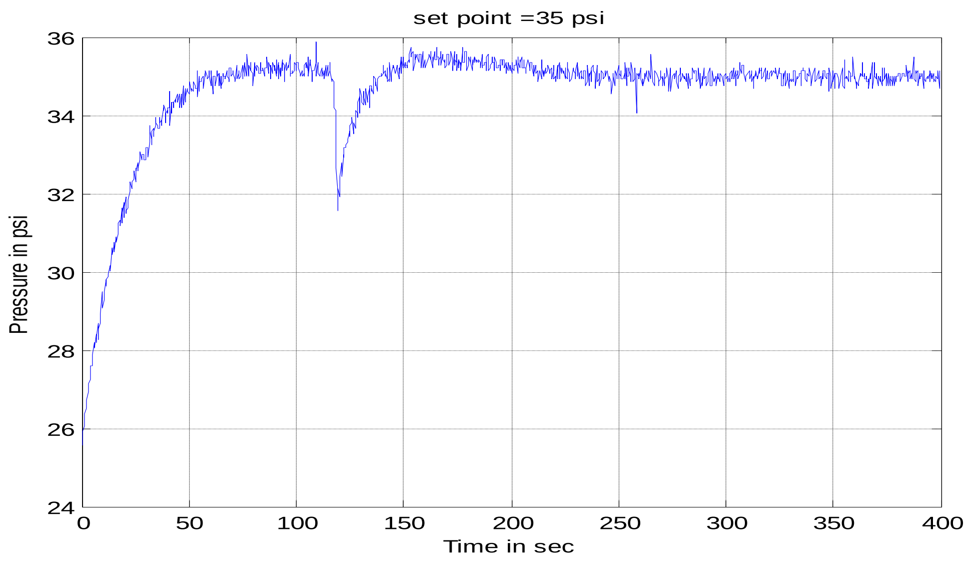

Fig.7 Disturbance rejection with RTDA

C. Robustness

An ideal controller should nullify any effects of model-process mismatch which shows its robustness property. For checking the robustness property instead of taking 18 as time constant, its value was modified to be 20 (ie, 10% process model mismatch).The process response didn't change drastically.It is shown in fig.8.It is seen that it is almost similar to previous response in fig.6.

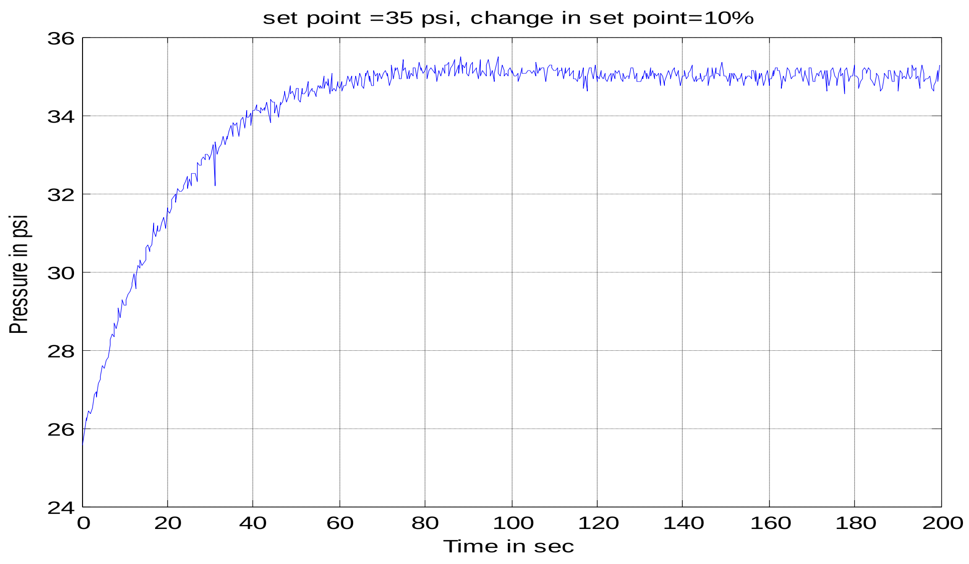

Fig.8 Robustness with RTDA

D. PID Implementation

PID controller is the most commonly used controller for pressure process[7].A SIMULINK model was developed for controlling the pressure process as shown in fig.9.Using the identified model PIDparameters were tuned.Tuning values are given by K_p=0.473191, K_i=0.02216, K_d=0. The pressure inside the tank was set at 23 psi and set point for the controller is 35 psi. Pressure response with PID is shown in fig10. It is also having good set pint tracking ability as we can see. But settling time is more when compared to the response with developed RTDA controller. In order to check the disturbance rejection property, a disturbance was introduced at the vent side at 330 sec and the response is shown in fig.11. It was taking more time to eliminate the effects of disturbance. Similarly for checking the robustness property, 10% change in model parameter was taken and corresponding response is shown in fig.12.

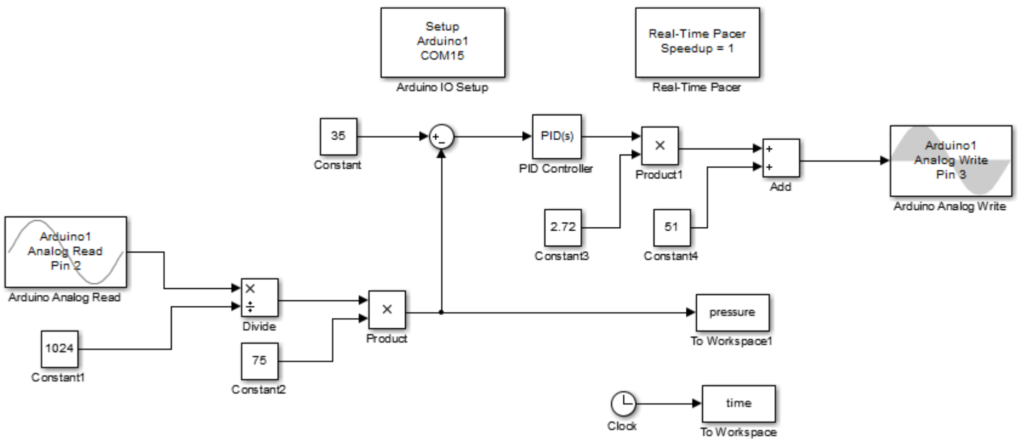

Fig.9 Simulink diagram for PID implementation

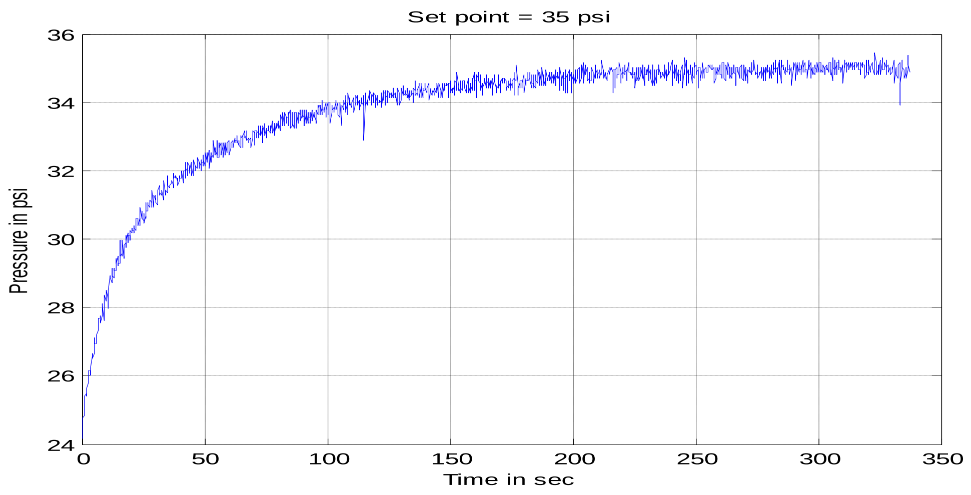

Fig.10Set point tracking with PID controller

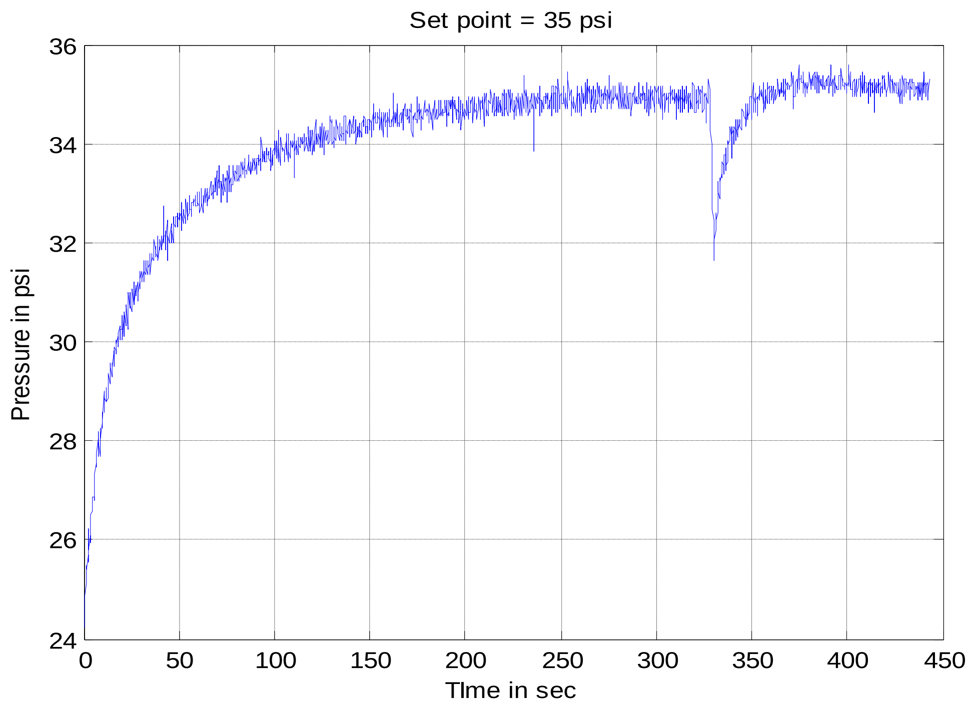

Fig.11Disturbance rejection with PID controller

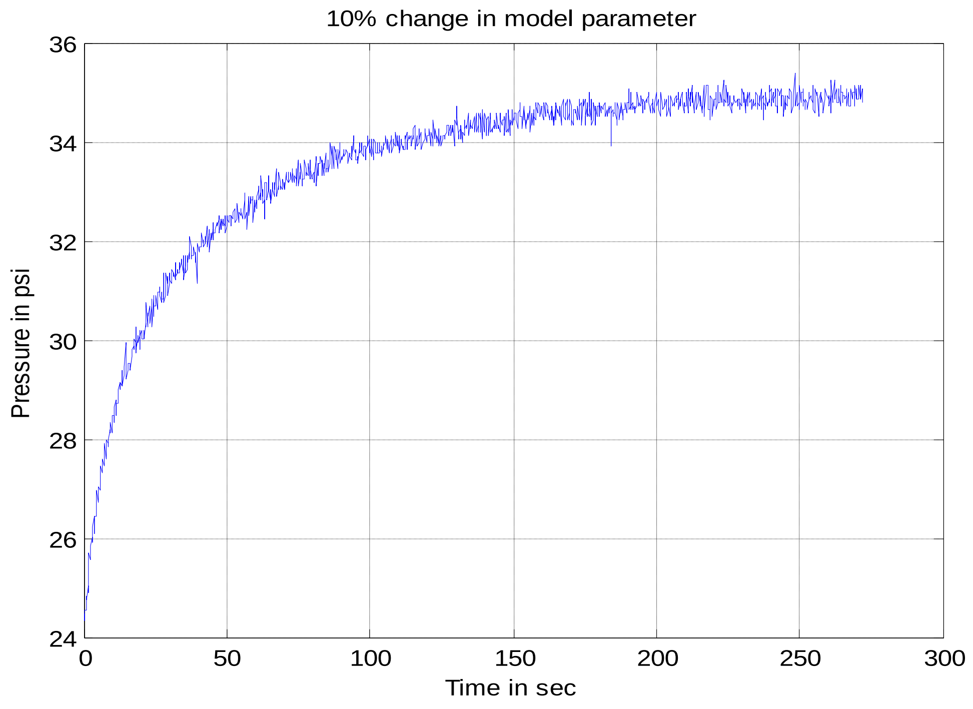

Fig.12Robustness with PID controller

VI. CONCLUSION

A new method of RTDA control strategy was used for controlling the pressure process. Its performance was tested for different properties like robustness, set point tracking and disturbance rejection in real time. The proposed algorithm provided good results for these tests. RTDA controller performance was compared with PID is shown in Table2.It provided better results compared to PID controller and can be easily selected as an alternative for PID controller.

Table 2: Performance Comparison

Specifications	**RTDA**	**PID**
Settling time (s)	70	250
Overshoot	0	0
Rise time (s)	60	200

REFERENCES

[1] Kapil Mukati, Babatunde Ogunnaike, “Stability analysis and tuning strategies for a novel next generation regulatory controller”, Proceeding ofthe 2004 American Control ConferenceBoston, Massachusetts June 30 -July 2, 2004

[2] Ender, D.B. (1993). “Control Engineering”, September, 180.

[3] Astrom, K.J., & Hagglund, T.(1984). Automatictuning of simple regulators with specifications onphase and amplitude margins. Automuticu, 20(5).

[4] Astrom, K.J., & Hagglund, T. (2001). The futureof PID control. Control Engineering Practice, 1163-1175..

[5] Ogunnaike, B.A. and K. Mukati, (2003)"Development, Design and Implementation of analternative structure for Next Generation regulatory controllers" Preprints of the Annual AIChE meeting, San Francisco, CA

[6] V.Bayaveereswaran, K J Nidhil Wilfred, S.Sreeraj, B.Vijay “System Identification from step input using Integral equation”, International Journal of Applied Engineering Research, ISSN 0973-4562, Vol. 8, No. 17 (2013), pp. 2165-2169

[7] Chen, D.; Seborg, D. E. “PI/PID controller design based on direct synthesis and disturbance rejection”. Ind. Eng. Chem. *Res.* 2002, 41, 4807–4822.