Self Tuned PID Controllers Theory and Design Through Implementation

Damoon Bazargan , University of Tehran, ECE Department
Damoon_2nd@ieee.org

Abstract—*In this article we will go through history and design of self tuning PID controllers and the methods used for their implementation . Among those we can name intelligent controllers and adaptive controllers being used by the usage of neural-fuzzy and adaptive algorithms.*

Keywords: self tune, PID controller.

1 Introduction

This document will provide a quick over view on PID controllers with a strong intention of discovering PID self tuning implementation. At the end we will digitalize the described problem and try to use the techniques we could obtain. In each step various comparisons will be made according to the simulated designs. We use MATLAB and simulink environment in order to gain and compare the desired results.

1.1 Why controllers?

Throughout history mankind has tried to control the world in which he lives. Fundamental to any control system is the ability to measure the output of the system and to take corrective action if its value deviates from some desired value.
In the other word we want the input signal of the system to be tracked exactly by the output or let say we want to have an overall system with unit transfer function(suppose you are in laplace plane).The question emerges here is that why we do not use a single simple wire in return of these methodologies.
The answer is that the plant should be used, suppose you are controlling a certain plant with a certain transfer function so you will have to use the controller unless you want to the input signal to be tracked, otherwise you will face some errors in output data in this regard. Figure 1 shows the simulated theory.

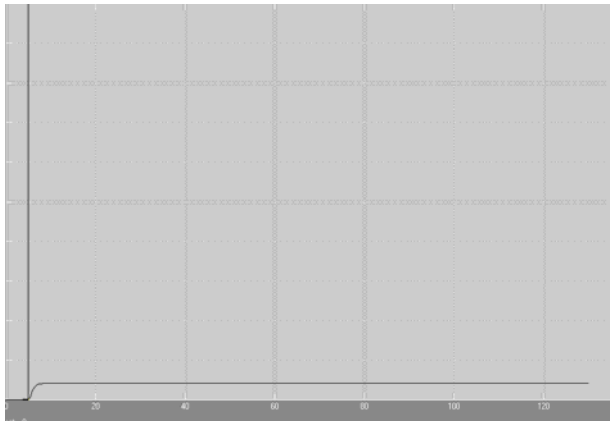

Fig 1 shows that the output could not track the input according to the plant used in fig 1

Using a simple controller in the described system even a proportional one can cause the output to change dramatically.

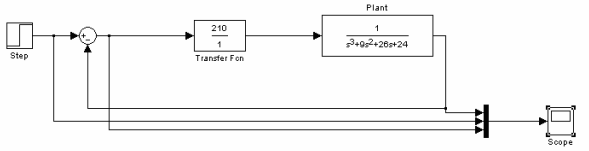

Fig 2 shows a close loop system with a proportional controller.

2 PID Controllers

A tracking problem should be protected against difficulties that might cause the system down such as steady state error, overshoot effect, rise time ,frequency response, disturbance , etc.
Using Proportional integral derivative controller (PID controller) the difficulties can be minimized. Here goes some of the key notes that could cause in a better design.
1-Increasing proportional parameter could decrease the steady state error while will cause system instability.
2- Increasing integration parameter can slower the system and will cause system instability.
3-increasing derivative parameter will faster the system and will improve overshoot effect and rise time.
The question remains is how to tune PID parameters in order to have the best result zigler-nicols has been on of the commonly used methods.

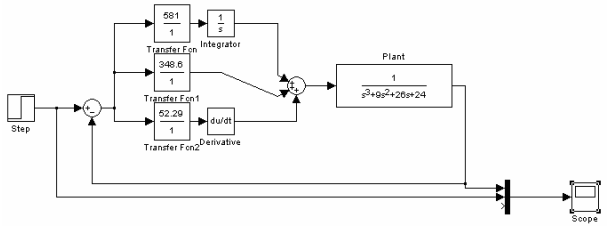

Fig 3 shows zigler tuned PID controller

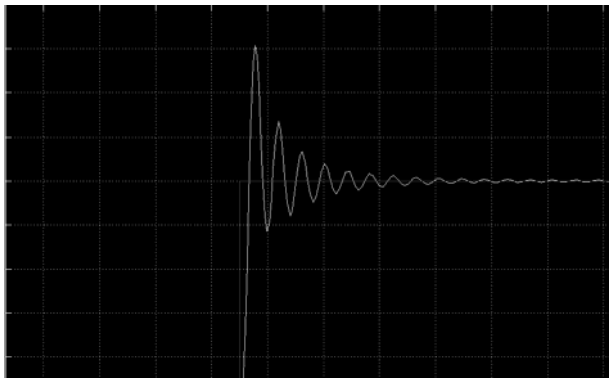

Fig 4 shows the simulated result output could track input

3 Self tuned PID Controllers

3.1 Introduction

Intelligent control has been studied actively in last ten years .These kinds of systems have been designed with capabilities of conditional decision making under unknown environment circumstances. Generally speaking we can divide these methods in to two groups. Those that have been based on adaptive and robust algorithms and neural network or fuzzy based ones.

3.2 Intelligent Control

Intelligent control using NN algorithms can be used for designing NNC (neural network controllers) These systems can be trained online or offline meaning through working or using training information. It is clear that online training based networks are much better than offline networks because no need of training data is felt. But even in online networks the required time for data generation and network training is a dismal property.

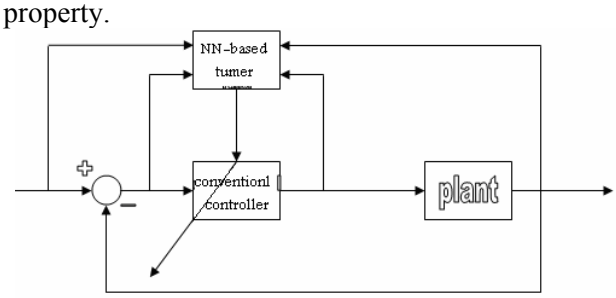

Fig 5 shows neural network based PID controller block diagram

3.3 Adaptive interaction

The most important factor that has distinced Adaptive interaction controllers from other predictive controllers is its dependence from plant specification this will be lead in high speed performance and robust properties.

3.3.1 Theory

Suppose one can divide the system into smaller subsystems with indexed integrable input and outputs through connective coefficients this yields these equations

$$x_n(t) = u_n(t) + \sum_{c \in I_n} \alpha_c y_{pre_c}(t), \quad n \in \mathcal{N}, \tag{1}$$

$$y_n(t) = \mathcal{F}_n[u_n(t) + \sum_{c \in I_n} \alpha_c y_{pre_c}(t)], \quad n \in \mathcal{N}. \tag{2}$$

By defining connection coefficients the performance function will be minimized

$$E(y_1, \ldots, y_n, u_1, \ldots, u_n)$$

3.3.2. Theorem 1

For the described system above we have:

$$\dot{\alpha}_c = \Big(\sum_{s \in O_{post_c}} \alpha_s \dot{\alpha}_s \frac{\frac{dE}{dy_{post_s}} \circ \mathcal{F}'_{post_s}[x_{post_s}]}{\frac{dE}{dy_{post_s}} \circ \mathcal{F}'_{post_s}[x_{post_s}] \circ y_{post_c}} - \gamma \frac{\partial E}{\partial y_{post_c}} \Big) \circ \mathcal{F}'_{post_c}[x_{post_c}] \circ y_{pre_c}, \quad c \in \mathcal{C}, \tag{3}$$

The above equation stands for coefficient parameters

Where this is its unique solotion

$$\dot{\alpha}_c = -\gamma \frac{dE}{d\alpha_c}, \quad c \in \mathcal{C}, \tag{4}$$

Where the constant is the adaptive coefficient.

3.3.3 Theorem 2

For a PID controller we devide the system into 4 devices proportional, integrative and derivative, and the plant so the equation will be

$$\dot{\alpha}_c = -\gamma \frac{\partial E}{\partial y_{post_c}} \circ \mathcal{F}'_{post_c}[x_{post_c}] \circ y_{pre_c}. \tag{5}$$

$$\begin{aligned} \dot{K}_P &= -2\gamma(y_4 - u)\mathcal{F}'_4[x_4] \circ y_1 \\ &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_1. \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{K}_I &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_2 \\ \dot{K}_D &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_3. \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{K}_P &= -\gamma e y_1 \\ \dot{K}_I &= -\gamma e y_2 \\ \dot{K}_D &= -\gamma e y_3. \end{aligned} \tag{8}$$

For future investigation see ref [2] C.4. Block diagram In this part fig 6 shows the used block diagram

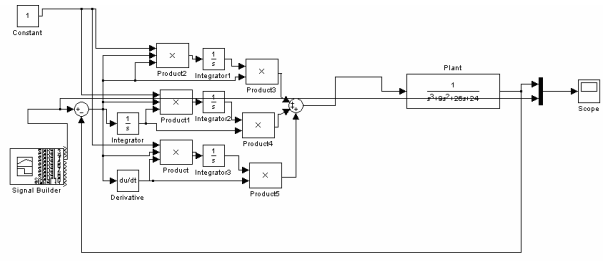

Fig 6 shows self tuning PID controller block diagram

4 Simulation

The question remain is how to tune the constant coefficient mentioned previously. We will go through this problem by simulation.

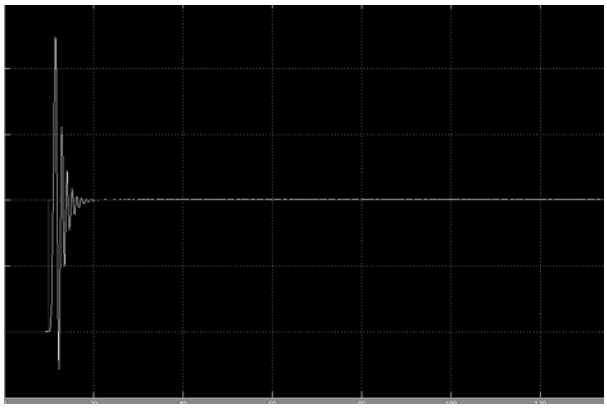

Fig 7 ,CTE=1 leads 150% overshoot with 10sec settle time

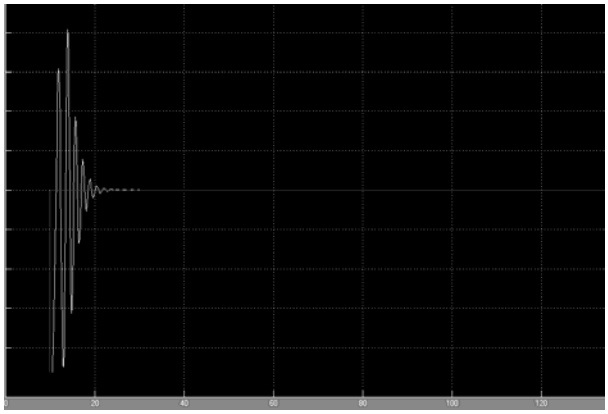

Fig 8 ,CTE=.5 leads 100 % overshoot with 20sec settle time

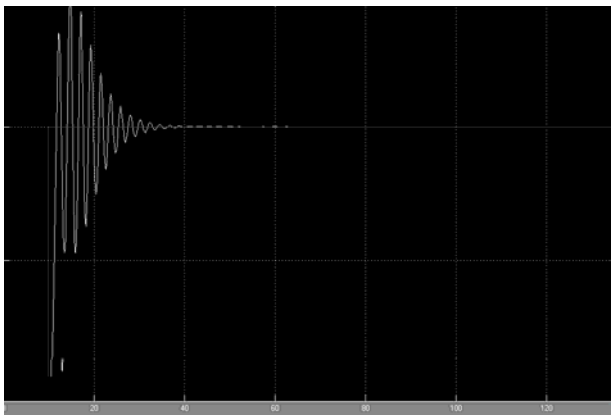

Fig 9 ,CTE=.3 leads 50 % overshoot with 40sec settle time

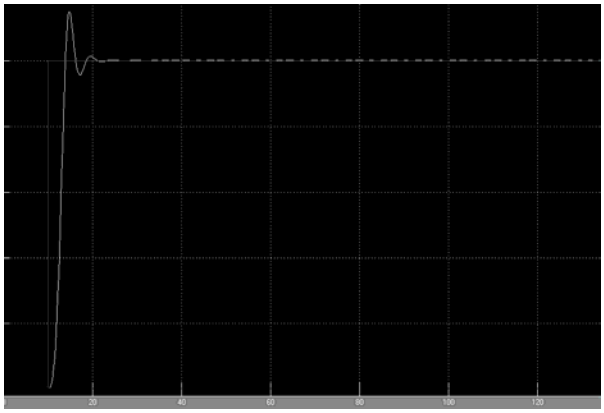

Fig 10 ,CTE=.03 leads less 15 % overshoot with 10sec settle time

As investigated above Decreasing constant value will lead in better performance.

5 Digitalization

Digitalizing a PID controllers is a commonly known procedure and it is assumed that the reader knows these basic concepts for further investigation see ref[3].

The question remains is how to digitalize the coefficient obtained from the discussed method above. According to Z transform proprieties the simulated results show that:

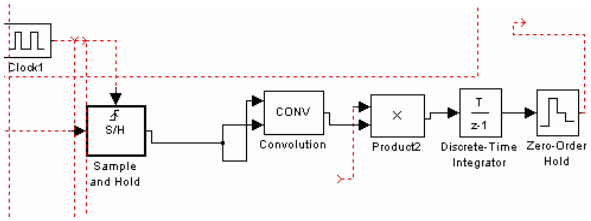

Fig 11 ,shows how the proportional coefficient can be get through error signal as its input

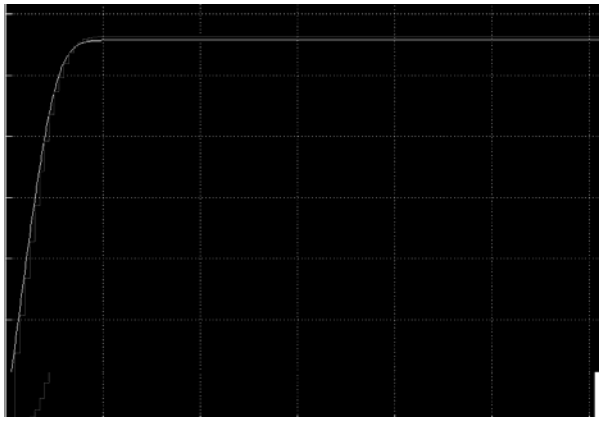

Fig 12,As you can see the gray signal could be obtained through digital processing with ignorable error

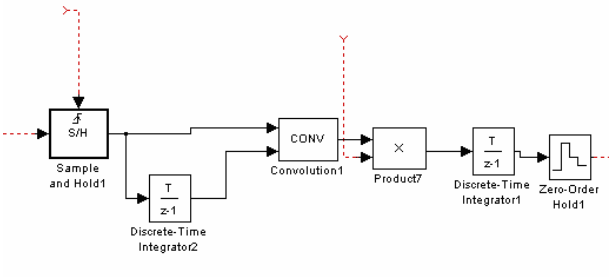

Fig 13 ,shows how the integrative coefficient can be get through error signal as its input

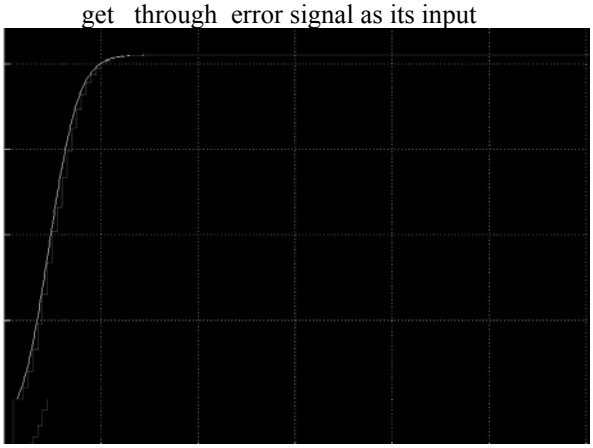

Fig 14 ,As you can see the gray signal could be obtained through digital processing with ignorable error

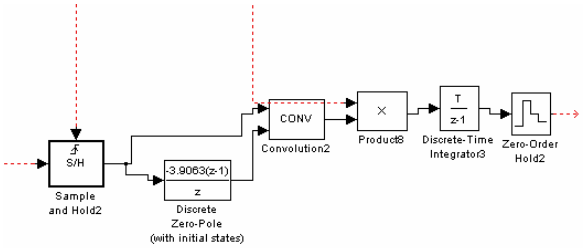

Fig 15 ,shows how the integrative coefficient can be get through error signal as its input

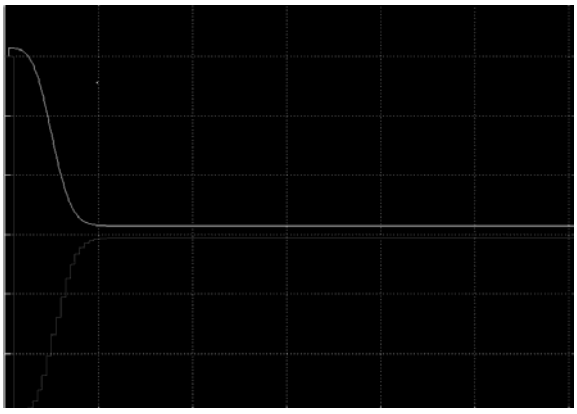

Fig 16 ,As you can see the gray signal could be obtained through digital processing with neglected error

6 Simulation Using S-Function

Insane improvement of technical computing led us to design a code based controller as it was our first ambition. Using s-function in simulink environment we could check the results of desired answers. For instance the code bellow can completely product kp coefficient..

```
temp=u0[0]*u0[0];
sum=sum+temp*srate[0]*gama[0];
y0[0]=sum;
```

Where U0[0] is the input and Y[0]0 is the output port of our hardware device.

7 Suitable or Unsuitable?

The question remains is how to tune self tuned controllers in order to be used in more projects. Actually the symbolic question is how a fox can track rabbits path with the least error involved. Fig 17 shows tracking of a Gaussian noise with 1 as its variance and 1 as its frequency and Fig 18 shows same noise with 4hz as its frequency.

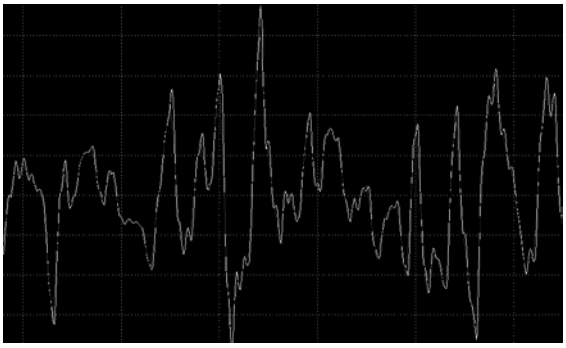

Fig 17 ,Noise traction with none self tuned method

Fig 18, increasing frequency caused some errors

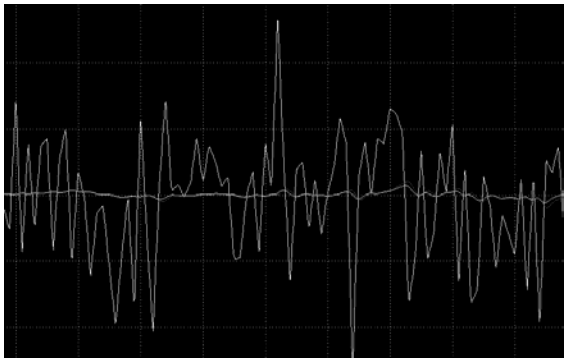

Fig 19 ,Not considering the tuning time in pid self tuned controllers caused ridiculous results

8 Conclusions

Developing a code based strategy we could design a digital self tuned controller with no need of plant recognition which means a more user friendly device and technique.The tuning period varies based on plants pole situations and its natural speed.

References

[1] Keigo watanabe, “Synthetic Intelligent control based of flexible neural network” Boston: Kluwer Academic publisher, 1999 2nd ed. vol. 19, Ed.

[2] karl j.Astrom "computer controlled systems theory and design "

[3] Fang lin, Robert D.Brandt, "Self tuning of PID controllers by adaptive interaction" *IEEE Trans.*