ASSIGNMENT REPORT

ON

"Automatic Tuning & Adaptation of PID Controllers"

BY

GROUP NO. – 17

NITIN PANT (2018A8PS0381P)

AYUSH AGRAWAL (2018A8PS0469P)

PRANAMYA JAIN (2018A8PS0769P)

APRIL, 2021

Table of Contents:

1. Introduction (Page 3)
2. Self-Tuning Adaptive Regulation of PID Controller (Page 3)
3. Techniques & Solution Proposed: (Page 4-15)
   3.1 Basic Block Diagram & Description of the System
   3.2 PID Tuning Approach & Controller Structure in PM Tuner
   3.3 Design Approach of the PM Tuner (Open Loop)
   3.4 Tuner Operation
   3.5 Tuner Initialization
   3.6 Adaptive Algorithm and PFE Block (for PI controller)
      3.6.1 Observation from the simulations
      3.6.2 Choosing the Adaptive Gain Parameters.
4. Simulation Results (Page 15-18)
   4.1 Comparison With some benchmarks tuning rules
5. Learnings, Observation & Challenges
6. References
7. Weightage of Efforts.

1. Introduction:

Tuning of PID has extensively been researched since the last 4 decades. Due to rapid increase in innovative solutions for control and instrumentation auto tuning of PID controllers has seen a boom. Although there have been many advances in this field, the desire to improve the applicability and robustness of the existing auto tuners still fuels much scope for research. Most of the time the exact dynamics of the plant are unknown hence it is required for an auto tuner to estimate the gains and dynamics of the plant and adjust itself automatically. There are three main classifications of auto tuning of PID controllers namely Step response, Periodic excitation and Relay based excitation schemes. The **step response** relies on a simple approach of open loop step testing to characterize the plant dynamics. In a **periodic excitation approach** the open-loop plant may be excited using more complex (binary or multi-level) pseudo-random or multi-sine signals, with the frequency spectrum tailored to the particular process in question. These types of signals, combined with appropriate signal processing tools, can yield accurate process models or frequency response estimates in which the effects of noise and process nonlinearities are suppressed. The **relay-based** tuning is based upon closed loop analysis which involves replacement of PID controller by a relay having hysteresis. The method is simple and requires no prior information and is also fast but not without limitations. We have worked on a **different autotuning method** which retains the PID controller in the loop as well and uses the information obtained by injecting a variable-frequency sine wave into the loop. Normally this sine wave is undesirable but in this case its amplitude can be reduced to match that of noise so output quality is maintained.

2. Self Tuning Adaptive Regulation of PID Controller:

There have been few serious drawbacks linked with the auto tuning of controllers using various approaches for auto tuning PID controllers as mentioned in the introductory paragraph. For example, sensitivity to disturbances in step-based auto tuners as the experiment is being performed in an open loop, also the use of a parameterized process model (e.g., first-order/dead-time), which in turn results in a lack of generality of the system. In a relay-based approach the approximations inherent in describing-function analysis can lead to errors in the estimate of the frequency point which in turn can result in less effective PID parameters. Also, in the same approach the load disturbances also influence the estimation accuracy. Another serious issue with the relay-based auto tuner as well as the above-mentioned open-loop schemes is that they are only suitable for off-line tuning i.e., no control is provided during tuning. But there may be a requirement where we need to control the tuning in between the tuning process depending upon if there are some conditions which change during the tuning process which might affect the functioning of the system. The aim of the controller (discussed in the paper) is to get rid of drawbacks mentioned above. The controller proposed is supposed to have positive points of relay-based tuning which include its **ease of use** and **speed**. It is also expected to have **greater flexibility**, giving superior PID tuning. It must also have **Robustness to: (1) noise, (2) set-point changes and (3) disturbances**. Being an in-loop auto-tuner, it can provide **continuous tuning** i.e., we can also provide control to the auto tuner while the tuning process is still going on of the parameters but at the expense of a low-amplitude perturbing sine wave signal being added to the loop’s set-point. In the paper they have also proposed a PID autotuning method which uses phase frequency estimation to generate suitable P I & D parameters. The method is different from most existing methods in the sense that the tuning is performed on-line, that is whilst the controller is undertaking CL control or in other words while the tuning of the controller is taking place.

3. Techniques & Solution Proposed:

3.1 Basic Block Diagram & Description of the System:

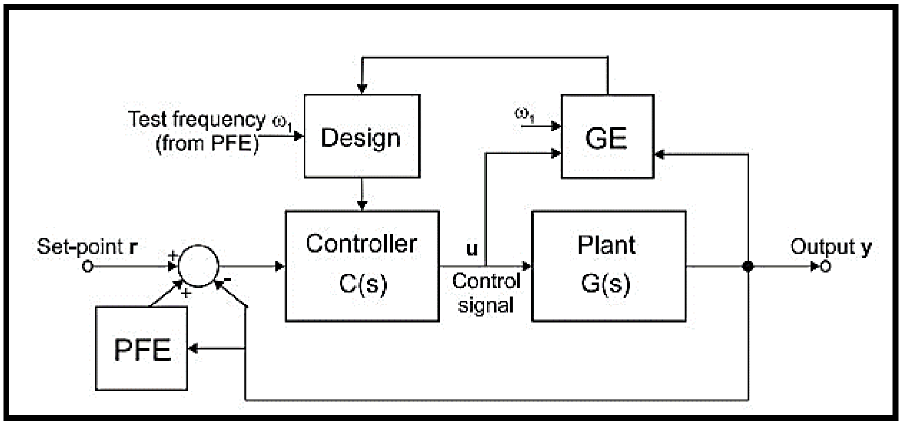

Fig1. Schematic diagram of system

Our primary objective is to adapt the controller so as to achieve a carefully chosen design point on the Nyquist Diagram. The process of choosing the appropriate design point is mentioned in the coming section. The main components used in the system involve the PFE (Phase Frequency Estimator) block, GE (Gain Estimator) block, Design Block, the controller and the Plant. The PFE (Phase Frequency Estimator) block injects a test sine wave into a system and continuously adapts its frequency (ω_1) until its phase shift attains a desired value (ϕ_{m}) i.e., the design point chosen on Nyquist Plot.

$$\frac{\mathrm{d}\omega_1(t)}{\mathrm{d}t} = K_a(\mathrm{Arg}\,\{G_c(\mathrm{j}\omega_1)\} - \phi_m) \quad \text{-(1)}$$

Where G_a is the adaptive gain of PFE and $G_c = \frac{GC}{1+GC}$ is the closed loop transfer function. C(s) is the transfer function of the controller and G(s) is the transfer function of the plant. In the above equation a frozen parameter assumption is taken which tells that the frequency (ω_1) and the controller gain C(s) are taken to vary slowly enough to allow transients to be neglected so that the response of $G_c(s)$ to the test signal is a Quasi-Steady State Sinusoidal meaning that the output from closed loop is sinusoidal signal as well. The GE (Gain Estimator) block works parallel with the PFE block. Inputs to this block are the instantaneous adapted frequency (ω_1) from the PFE block and the signal input to the plant and output from the plant so as to calculate the Plant Gain $|G(\mathrm{j}\omega_1)|$. GE actually monitors the combination plant/filter. Although the effect of the filter is generally small, accounting for it in the adaptation of the controller allows for more accurate convergence to the design-point. Other important parts of the tuner include VBPF (Variable Band Pass Filter) Block which is placed at the inputs of PFE and GE block. VBPF is a second order filter centered at the current value of test frequency (ω_1). Basically, these filters are used to isolate the probing signal from other signals circulating in the loop (such as noise, set point changes, load disturbances, etc.). The algorithm is initialized using a FODT (First Order Dead Time) approximation Ga(s) for the plant, obtained through a simple step test. The initialization involves computation of Ka (adaptive gain of PFE), ω_1 (t=0) and some other parameters associated with the PFE block, GE block and the controller. Basic form of this method requires amplitude and frequency of waveform to be accurately determined. Two amplitude estimators are being used in the GE block to determine the plant gain and a phase estimator is used in the PFE block to detect the phase.

3.2 PID Tuning Approach & Controller Structure in PM Tuner:

The design approach being used uses Nyquist ideas, with an aim to choose a design-point D_1 so that the compensated Nyquist plot exhibits certain characteristics and as a result favorable Closed Loop behavior is obtained. If a typical Type 0 plant i.e., a system having no pole at the origin, Ideally the Nyquist curve (Fig. 2) should be the vertical line with Real value of -0.5 parallel to the imaginary axis meaning unity CL gain at all frequencies and an infinitely fast response. However practically this is not achievable as practical systems exhibit a finite bandwidth and slew rate, so CG(jω) tends to 0 as frequency approaches infinity. Hence the best possible shape is something alike to curve B, with a high wd to ensure a fast CL response. A compensated Nyquist contour similar to A, where the curve 'bulges out' towards the critical point $(-1 + j0)$ is highly undesirable, as this indicates an oscillatory response. Also, the design rules which give a curve such as C (having a low frequency asymptote away from Re=-0.5) should be avoided, as they result in a slow CL pole. A natural choice is to make the design-point D1 a phase margin (PM) point and to use an extra constraint on the controller parameters to ensure that the Nyquist contour does not protrude towards the critical point $(-1 + 0j)$. **This is the design rule implemented in the PID tuner in the paper. We can also refer to the tuner as PM (Phase Margin) tuner.** The controller structure used in the PM tuner is as depicted in Fig. 3.

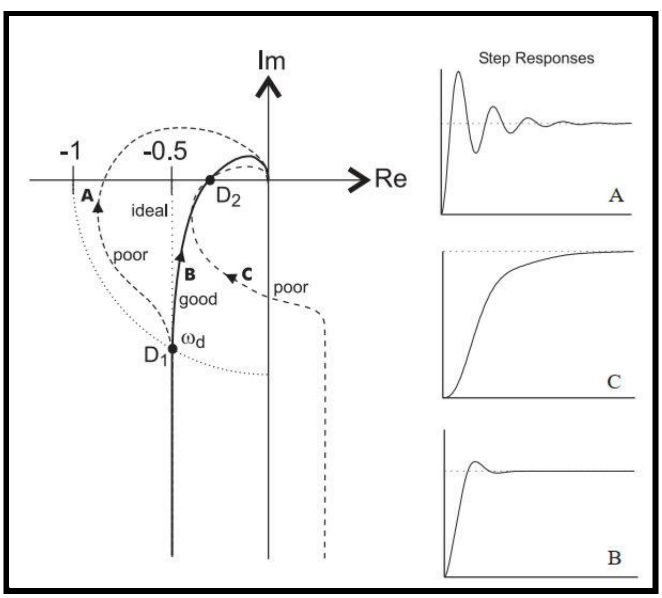

Fig2. Desired shape for compensated Nyquist plot

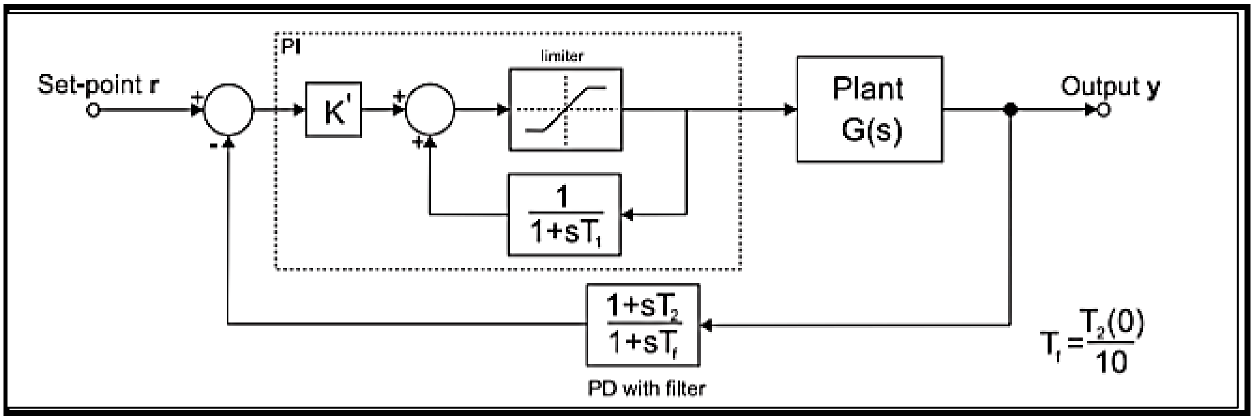

Fig3. Desired shape for compensated Nyquist plot

The controller is implemented in serial form and is split into two components: a PI and a PD unit, with the plant in between. Hence derivative action is applied on the output of the plant and 'derivative kicks' resulting from set-point changes are avoided. The PD module is filtered to prevent excessive amplification of measurement noise and the configuration used for the PI module protects the controller from integral wind-up. The effective transfer function of the overall controller (ignoring the filter) takes on the standard form for a PID compensator:

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_D\right) - (2)$$
$$= K\frac{(s^2T_iT_D + sT_i + 1)}{sT_i} \quad - (3)$$

where $K = \frac{K'(T_1+T_2)}{T_1}, T_i = (T_1 + T_2)$ and $T_D = \frac{T_1T_2}{(T_1+T_2)}$

3.3 Design Approach of the PM Tuner (Open Loop):

The aim of the tuner is to achieve a desired phase margin (phi m). Denoting the compensated open-loop by G_0 (jω). Hence the design problem becomes:

$$\begin{aligned} G_o(\mathrm{j}\omega_\phi) &= K\left[1 + \mathrm{j}\left(\omega_\phi T_D - \frac{1}{\omega_\phi T_i}\right)\right] G(\mathrm{j}\omega_\phi) \quad -(4) \\ &= -\exp(\mathrm{j}\phi_m) \end{aligned}$$

From the above equation by equating the real part with the real part and the imaginary with the imaginary we obtain two equations. We have 4 unknowns to be found out namely, $\boldsymbol{K}, T_i, T_d, \omega_\phi$.

But we only have 2 equations hence to uniquely determine the values of all the 4 parameters we need more equations. These are provided two additional constraints.

Constraint 1: Ratio $r = \frac{T_d}{T_i}$ is fixed at 0.25 which means that the controller has a double zero at $s = -\frac{2}{T_i}$. This constraint is found to be suitable for suitable typical plant dynamics.

Constraint 2: Based on the graphical method for design of PID controllers mentioned in paper (Hind 1980). From here we get the equation of Ti in the terms of a.

$$T_i = \frac{a+\sqrt{a^2+4r}}{2r\omega_\phi} \text{ - (5)}$$

Now we need to fix parameter 'a'. Once we fix the parameter a we get 4 equations hence we can uniquely determine the value of 4 variables. **Finding the value of 'a'**: Fortunately, the FODT model (see Eq. (6)) obtained from the prior step test allows the parameter a to be effectively chosen.

$G_a(j\omega)$ for the plant, obtained through a prior step test:

$$G_a(s) = \exp(-sT_d)\frac{K_p}{1+sT} \text{ - (6)}$$

FODT models of higher-order plants (and plants with a dead-time) are characterized by a longer relative dead-time = $\lambda = \frac{T_d}{T}$. By selecting '**a**' according to λ , good tuning can be obtained. In practice the exact value of λ obtained for a given plant will depend on the duration of the step test (T). Hence, we look to cover the likely range of λ by just a few values of 'a', each being suitable for a wide range of λ. Taking FODT plants with λ ranging from 0.1 to 4 and using ϕ_m=60, we can obtain a fairly consistent step response (similar overshoot and no oscillatory behavior or slow pole) and robustness (gain margin >2) by choosing between just three different values for a; as shown in Fig. 5.

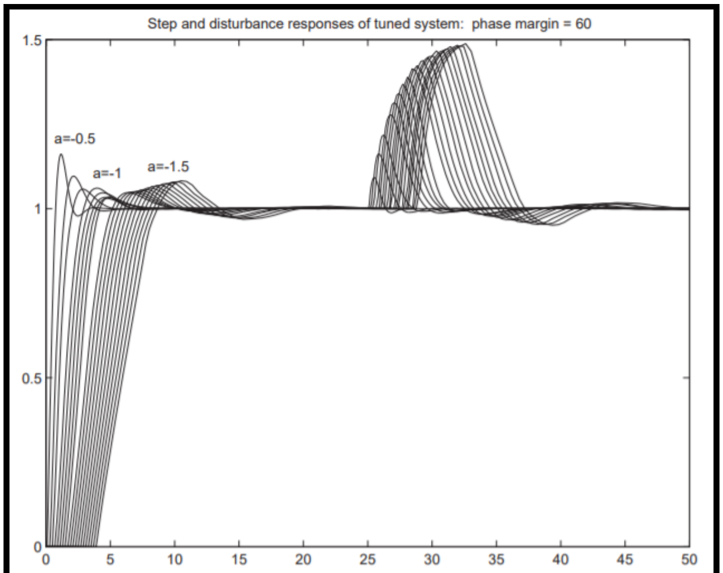

Fig. 5. Choosing *a* for FODT systems.

Fig4. Choosing a for FODT systems.

Moreover, these values include some leeway for inaccuracies in l and are also appropriate for ϕ_m>60. For ϕ_m<60 the slightly altered values given in Table 1 must be used.

ϕ_m	$\lambda \leq 0.5$	$0.5 < \lambda \leq 1.5$	$\lambda > 1.5$
$\leq 60°$	-0.5	-1	-1.5
$> 60°$	-1	-1.5	-2.5

Table 1 – Choosing a depending upon λ and ϕ_m

The values obtained this way also turn out to be suitable for a wide range of non-FODT plants with a few exceptions. Provided there is negligible deadtime, a value of around a = 1.5 (so that P is roughly 3/4 the way around the design circle) is found to be appropriate for both types of processes (from the perspective of achieving the fastest response without compromising on robustness). For Type 1 plants having considerable dead-time 'a' should be lowered to 0. Unfortunately, in the case of underdamped systems with dead-time the appropriate value of a is not well defined. However, it is found that with high λ processes ($>$ 2.5), whether or not they involve oscillatory dynamics, it is often useful to use $a < -3$ if good robustness is to be ensured. With a fixed the resulting design problem is guaranteed to yield a solution provided that the open loop phase shift of the plant reaches that corresponding to P as the frequency is increased from zero. We can express this condition as

$$\text{Arg}\,\{G(j\omega)\} = -\pi + \phi_m - \tan^{-1}\, a \quad \text{for some } \omega = \omega_\phi \quad \text{- (7)}$$

where ϕ_m is the desired phase margin.

3.4 Tuner Operation

The CL frequency response associated with the design-point D may be expressed using the triangle QDO (Hind 1980)

$$Gain: K_d = \frac{OD}{QD} = \frac{\sin\,(\pi-\phi_m)/2}{\sin\,\phi_m} \text{ – (8)}$$

$$\text{Phase shift: } \theta_d = -P\hat{O}D = -\frac{(\pi-\phi_m)}{2} \text{ -(9)}$$

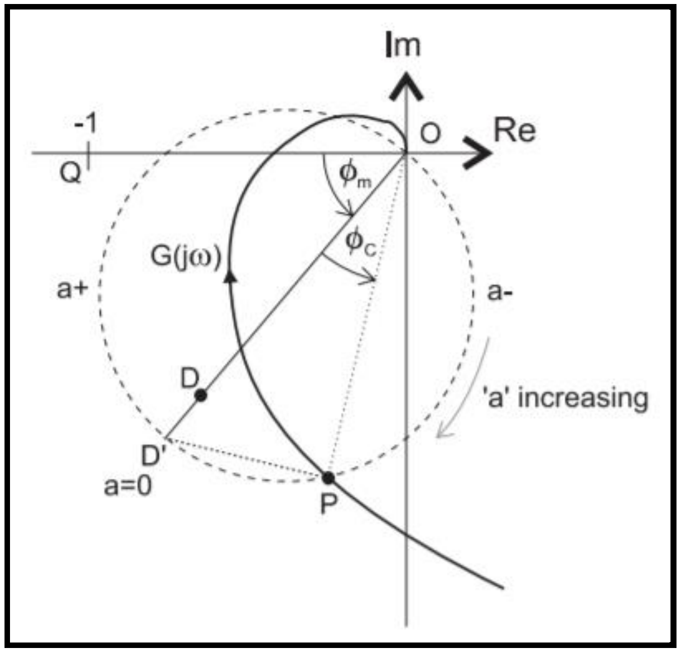

Fig5. Graphical PID Design (Hind 1980)

To implement the design rules in the tuner, ϕ_m is specified in the PFE as the desired phase shift for the test frequency ω_1. The PFE will vary the frequency ω_1 of the probing signal such that its phase shift approaches that of the design-point we want the phase shift to approach ϕ_m. At the same time, we seek to adapt the controller parameters so that the open-loop gain at ω_1 corresponds to that dictated by the design point (OD = 1), whilst ensuring that the design parameter a is met, and so convergence is obtained to the specified PM ($\omega_1 = \omega_\phi$). Thus, from Eqs. (3) and (4) the required adaptation is:

$$K = \frac{1}{\sqrt{1+a^2}|G(\mathrm{j}\omega_1)|},\ T_i = \frac{2}{\omega_1}\left(a + \sqrt{a^2+1}\right) \quad \text{-(10)}$$

$$T_D = 0.25T_i - (11)$$

3.5 Tuner Initialization:

The tuner consists of numerous parameters which need to be well-chosen if effective operation is to be ensured. Few things are to be kept in mind during initializing. First, that the initial value for the PFE frequency ($\omega_1(0)$) must be reasonably close to the actual design frequency of ω_ϕ. Secondly, The adaptive gain K_a of the PFE needs to be large enough to ensure a reasonably fast rate of convergence, yet excessively high values lead to instability. Hence a procedure must be devised that initializes the algorithm in a semi-automatic manner. The initialization being based on a FODT model by taking the $G_p(s) = G_a(s)$ where $G_p(s)$ is the Transfer Function of the plant:

$$G_a(s) = \exp\,(-sT_d)\frac{K_p}{1+sT} \quad - (12)$$

In most cases the step test will only need to be carried out the first time the algorithm is run on a specific plant.

1. Selection of the PID design parameter **'a'** according to the FODT model as dictated by the table 1 based upon lambda and phase margin.
2. Estimation of the frequency corresponding to the design-point by carrying out the PID design on the FODT model $G_a(s)$. The estimated frequency is denoted by ω_ϕ^e.

   ω_ϕ^e can be solved using:

$$-\omega_\phi^e T_d - \tan^{-1}(\omega_\phi^e T) + \pi - \phi_m + \tan^{-1} a = 0 \quad - (13)$$

3. Next, the initial frequency $\omega_1(\mathbf{0})$ of the PFE is set to 'ω_ϕ^e' while the initial output of the GE to $G_a\,(j\omega_\phi^e)$ allows start-up controller parameters to be generated.

4. The time constant of the controller filter is fixed at $T_f = \frac{T_2(\mathbf{0})}{10} = \frac{T_i(\mathbf{0})}{20}$ .Where $T_i(\mathbf{0})$ is calculated using:

$$T_i = \frac{2}{\omega_1}\left(a + \sqrt{a^2 + 1}\right) \quad - (14)$$

   (Since we have $\omega_1(\mathbf{0}) = \omega_\phi^e$ and we already selected 'a' implies we can find $T_i(\mathbf{0})$)

5. The adaptive gain K_a for the PFE is then calculated from the design-point frequency estimate and the FODT model, All the remaining parameters of the PFE and GE are calculated as specified in the previous paper (not available online).

6. The VBPF bandwidths $\beta = \frac{\Delta\,\omega}{\omega_1}$ are given values of 1.5, 0.15 (PFE, GE), and the time constant of the GE post filter is set to $T_f = \frac{10}{\omega_1(0)}$ - (15)

The initialization procedure is finished when the PFE is being switched on and starts to inject the sine wave into the CL system. These sinewaves then excite the plant (after a delay T_{d} and its output then passes through several VPBFs and Hilbert transformers. To allow their responses to settle, the updating of ω_0 and $|\widehat{G_p}|$ was inhibited for an arbitrary period $2(T_{\mathrm{d}} + \tau)$.

NOTE: To allow the responses of the estimators to settle before they embark on adjusting the controller parameters, the adaptation of the test frequency ω_1 and the plant gain estimate G_{p} ($\mathrm{j}\omega_1$) is halted for a $period = 2(T_{\mathrm{d}} + \tau)$. Where τ relates to the dead-time in the PFE and GE resulting from the use of Hilbert transformers.

3.6 Adaptive Algorithm and PFE Block (for PI controller)

An algorithm particularly suited for this application is a phase–frequency estimator (PFE), whose structure resembles a phase-locked loop .In essence the PFE generates a sinewave A sinw0t of amplitude A and, using an appropriate phase-sensitive detector (PSD), adjusts the frequency w0 until the phase difference between the generated and measured sinewaves is the desired value ϕ_m:

$$\frac{\mathrm{d}\omega_1(t)}{\mathrm{d}t} = K_a(\mathrm{Arg}\,\{G_c(\mathrm{j}\omega_1)\} - \phi_m) - (16)$$

A diagram showing the approach executed over a PI tuner is added below

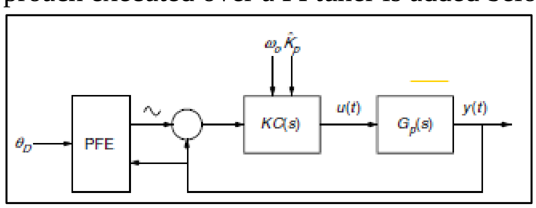

Fig6. PI Auto Tuner Using PFE

$$K(t) = \frac{p\cos\,\phi_D}{|\hat{G}_p(j\omega_o(t))|} \Rightarrow \frac{1}{2\hat{K}_p(t)}$$

$$T_i(t) = \frac{1}{a\omega_o(t)} \Rightarrow z_C = \omega_o(t)$$

The sinewave is applied to the setpoint of the loop and ω_1 adapted until the closed-loop phase shift attains the required ϕ_D.

The closed loop is itself time varying as we need to tune the controller based on the current values of ω_1, G_p^e(estimated G_p), we require an amplitude estimator for the plant gain, the paper introduced a Hilbert-transform PSD. A Hilbert transformer (designed using the same rules as for the PFE) and its associated delay operator of m=2 samples provide two signals that have a relative phase shift of $\frac{\pi}{2}$.

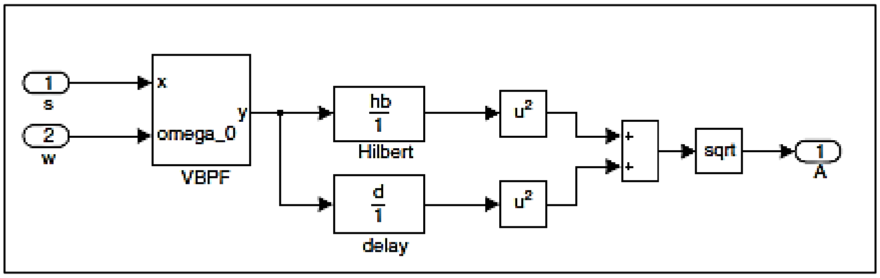

Fig7. Amplitude Estimator

Before the input is given to the Amplitude estimator we need to pass it through a VBPF

$$F_v(s) = \frac{\beta\omega_o s}{s^2 + \beta\omega_o s + \omega_o^2} \; (frequency\ response\ of\ VBPF)$$

Whose center frequency is the current value of the ω_o,(As given by the PFE) and whose bandwidth is the design parameter beta. The VBPF is used for the following reasons:
(i) elimination of DC values;
(ii) attenuation of harmonics, arising for example if there
are nonlinearities in the plant or actuator;
(iii) reduction in the effect of additive noise.

Testing this method:
1 Hz sinewave with unit amplitude was added to the estimator, using a sample rate of fs = 5 Hz (sample interval = 0.2 s) and m = 20;(the number of Hilbert transform parameters) Hence the delay implies $\frac{mh}{2} = 2sec$. Therefore the time by which a change ripples in all m terms of Hilbert Transform is 4 sec = mh

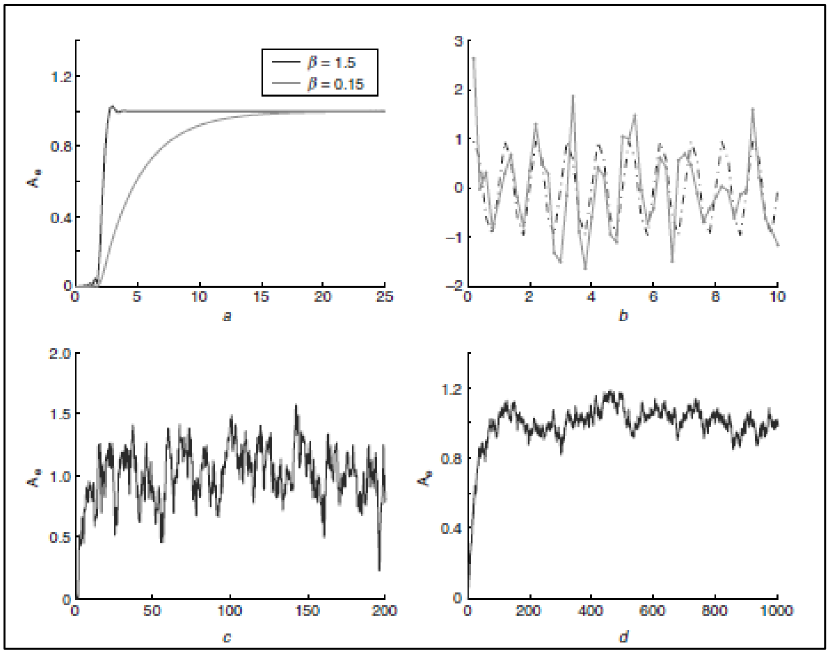

Fig 8.

3.6.1 Observation from the simulations:

The top-left plots show the noise free output, verifying that for $\beta = 1.5$ the output is correct and ripple-free after the corresponding delay. With the smaller value of $\beta = 0.15$ there is an additional effect: an exponential rise with time constant $\frac{1}{b\omega_o}$ .Thus increasing the settling-time. The power in the sinewave is 0.5, so to generate noise of the same power (giving unity signal: noise ratio) the bandlimited noise had spectral density $S_n = \frac{h}{2}$ for the sampling interval h.

A section of the noise-free and noisy sinewave is shown in the top-right of the Figure. The lower plots show the estimator's output for b=0.15; 0.015 respectively. As expected, the reduction of b by 10 reduces
sVar$\{\hat{A}\}$ also, by 10 (SD by $\sqrt{10}$) - the usual time/variance trade-off. Note. The estimator is biased, clearly apparent for $b = 1.5$; but for the values of b in the Figure the bias is small. As the SNR is unlikely to be much less than 1 in applications, it is clear that the estimate's variance is more important: indeed it is useful to smooth the estimate further (say by a simple lag with time constant T_f): Hence in the simulations reported below

$$\left|\hat{G}_p(j\omega_o)\right| = \frac{\hat{A}_y^f}{\hat{A}_u^f}$$

is used, where the superscript f denotes a posterior filtering action: typically $T_f = \frac{10}{\omega_0(0)}$

3.6.2 Choosing the Adaptive Gain Parameter:

We need to select adaptive gain K_a in a semi-automatic way. The idea is to use a procedure that initializes the algorithm without the need for user intervention. We initialise all the parameters of the PFE from a step-test and by fitting the experimental response to that of a first-order/dead-time (FODT) model

$$G_a(s) = \exp\left(-sT_d\right)\frac{K_p}{1+sT}$$

The model is time-normalized using $t' = \frac{t}{T} \rightarrow p = sT$ to give the NFODT model

$$G_A(p) = \exp\left(-p\lambda\right)\frac{K_p}{1+p}$$

The model is time-normalized using give the NFODT model and initialization depends on the dimensionless parameter $\lambda = \frac{Tdelay}{T}$ Through an approximate theory and simulation over a range of test cases, a value of adaptive gain K_a was deduced in an earlier paper that is near-optimal for a fixed transfer function. However, in this application the closed-loop is changing during adaptation, as the PI parameters are continually being updated. Hence the adaptive gain for the results shown subsequently was 'detuned' by (say) $K_a = \frac{K_{po}}{4}$; where K_a is the 'optimal' value suggested by the earlier paper. While this arbitrary detuning was effective for the reported simulations, inspection of the results shows that a more comprehensive theory is required. The prior step-test provides a FODT model for G_p; not G_o; G_c: Hence full initialisation requires an estimate for the starting PFE frequency $\omega_0(0)$ and loop-controller KC(s) at t =0 .To achieve this we proceed from the design point D and back-calculate the corresponding point P on the FODT's Nyquist Diagram.

From the design point we already are aware of the various angles(ϕ_d, ϕ_c) hence :

arg $G_A(j\Omega_o) = -\pi + \phi_D + \phi_C = -\lambda\Omega_o - \arctan\,\Omega_o$ where $\Omega_o = \omega_o T$ is the normalized frequency can be easily calculated.

Therefore finally;

$$K(0) = \frac{p\cos\,\phi_D\sqrt{1+\Omega_o^2(0)}}{K_p},\; T_i(0) = \frac{T}{a\Omega_o(0)}$$

The controller is initialized with these values.

From the local slope of NFODT's phase frequency response is:

$$G_A^*(\Omega_o) = \frac{d \arg\, G_A}{d\Omega}\bigg|_{\Omega-\Omega_0} \Rightarrow G_A^*(\Omega_o(0)) = -\lambda - \frac{1}{1+\Omega_o^2(0)}$$

Therefore, the final Adaptive Gain is:

$$K_a = \frac{0.75\pi}{T^2 G_A^*(0)\lambda(3\pi + 2m)}$$

M: number of Hilbert Transform parameters
As mentioned earlier, this gain needs to be detuned by a factor of 4[a value that works well for type 0 plants]
But for integrating type 1 plants, the NFODT is a very poor predictor of plants Nyquist diagram. As the estimated Kp, T are large and the dead time is very small. Hence for type 1 plants we detune it by a factor of 8. If this also underperforms then detune it by a factor of 16

The parameters of the passband prefilters and simple-lag postfilters were chosen by simple (and perhaps unjustifiable) rules-of-thumb.

4. Simulation Results:

Three series of simulation were performed by the researcher using MATLAB and Simulink. First set of Simulink involved noise free plant and was used to initialize the parameters and analyze the convergence of the algorithm. Second set of simulation was done to compare the tuning performance of the designed controller with then available autotuning methods. Finally, the third set of simulations was done for evaluation of real-world performance of the design.

Convergence to Design point:
Initially an open loop test was carried out to formulate a First Order Dead Time model of the plant.
Next step was to initialize the algorithm and then leave to self-operate to achieve convergence. This was followed by a step test on both initial and final, tuned controller. In between the initial and final states, a distribution of certain magnitude was given to the loop.
Further the author compares the results obtained with the benchmarks of tuning performances by the optimal PID design method of Panagopoulos, Astrom, and Hagglund (2002).

Firs test was performed on IG_3 i.e.

$$IG_3(s) = \frac{1}{s(1+s)^3}$$

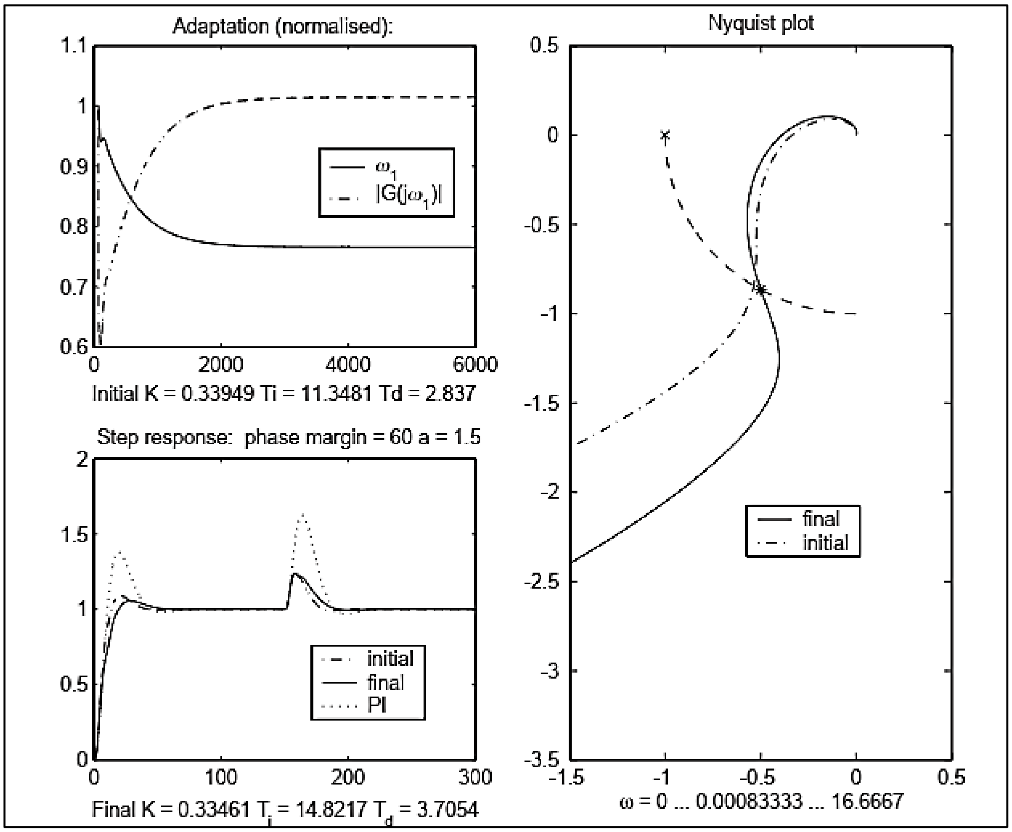

Fig9. Simulation for IG3

Above figure shows the results for G3. It shows the reaction of PID tuner before and after tuning. The right most figure shows Nyquist plots for the open-loop and final tuned system. Star in the Nyquist plot denotes the desired design point and cross denotes the critical point which is extensively discussed previously in this text. Although the FODT model does not represent the integrating plants very clearly still the initialization is adequate and the tuner successfully converges, but the process is a bit slow and can be considered a caveat of using this method. Also as discussed in the text above the tuner is expected to have a contour similar to that of desired contour in Fig 9 .but for lower frequencies the curve bulges and crosses the ideal $R_e = -0.5$ contour.

A similar test was done on G_7 i.e.

$$G_7(s) = \frac{1}{(1+s)^7}$$

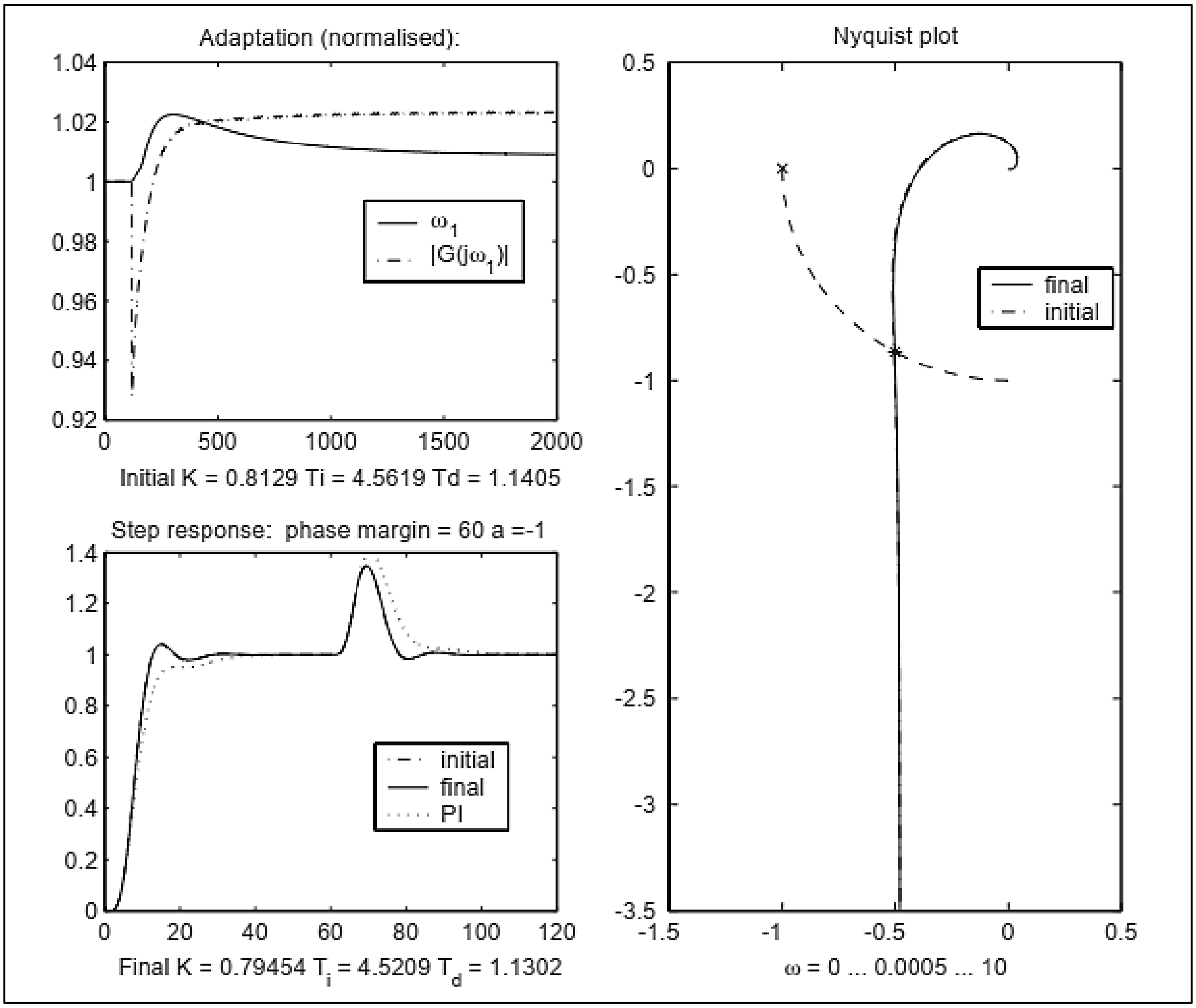

Fig10. Simulation for G7

For this plant the initial PID deduced from FODT did not require much tuning as we can see the initial and tuned run for a step input show very similar output, reason being that FODT model gives an excellent fit for high order plants and for plants with significant dead time. This system follows the desired Nyquist contour quite accurately.

4.1 Comparison With some benchmarks tuning rules:

Ziegler & Nichols method (including Astrom & Hagglund 1984) gives simple rules to find out the tuner settings. It works by calculating the plant gain K_u (Ultimate gain) and frequency f_u corresponding to $Arg\{G(j\omega)\} = -\pi$. Generally, the tuning obtained by this method is considered primitive and less effective as the rules are derived solely for a disturbance rejection criterion. Some more recent schemes ensure good robustness by using two frequency points (or a FODT model) to achieve (in some cases approximately) a prescribed phase margin and gain margin (Tan, Lee, & Wang,1996; Ho, Hang, & Cao, 1995). However, the tuning is often less than favorable in other respects (e.g., speed of response) and a certain degree of inconsistency exists regarding the choices of PM and GM (Tan et al., 1999).

The author has mainly compared three methods that are Internal model control, Kappa-Tau ($\kappa - \tau$) method and the PFE tuner being discussed above. In $\kappa - \tau$ method M_s was chosen to be 1.4 which corresponds to the sensitivity resulting from PFE tuning with phase margin set to be equal to 60). Design parameter b for IMC method was chosen to be lowest possible so as to give fastest Closed loop response.
The comparative tests were carried out as follows. For each test plant, the PFE tuning algorithm was run and the final controller settings recorded. The FODT mode obtained by the initialization procedure was then used to compute PID parameters by the IMC method. Next a relay experiment was performed to provide the Kappa Tau PID design. The CL (set-point and disturbance) responses obtained by the three methods were subsequently examined.

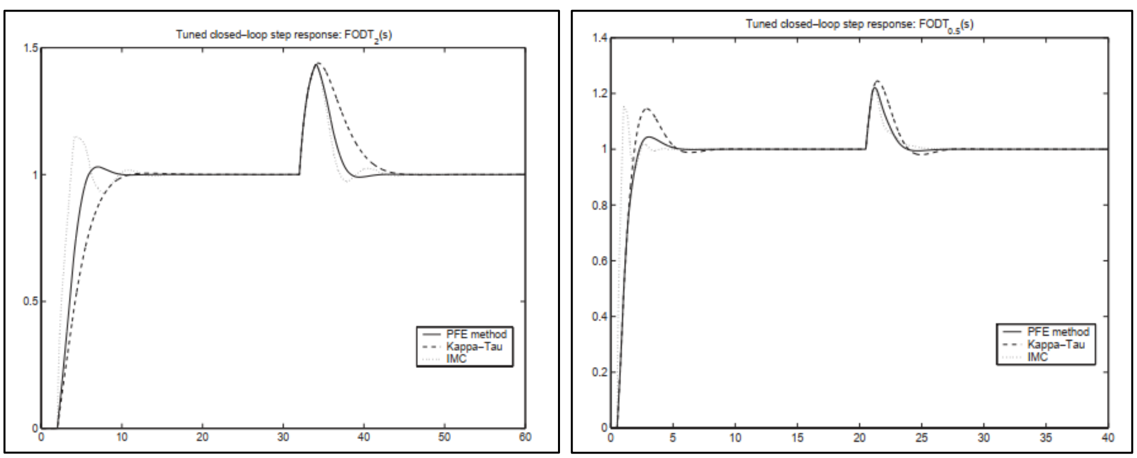

Fig 11. Comparative Results for $\lambda = 2$ & 0.5 respectively.

5. Learning, Observations & Challenges Faced:

We were able to implement only the individual blocks and verify their results. When run together there were some errors and difficulties that we mailed Dr. Puneet Mishra. We tried our best and worked diligently towards this project's implementation.

This was a great learning experience for our group. Tuning of PID was covered in depth by Dr. Puneet Mishra but this aspect of automatic tuning opened a whole new arena of learning for each one of us We were very recently taught the concept of adaptive control in the lecture classes taken by Dr. Surekha Bhanot. This project turned to to be a great practice and, an implementation of this paper increased our understanding of "adaptive control " We were throughout the duration of assignment enthusiastic and willing to learn new concepts.

We are now comfortable to approach any further applications of this concept with full confidence and conceptual clarity.

We even got the support and guidance of Dr. Valencia G. (the author of one of the review papers on this topic) He was kind enough to guide us in this aspect and help us with the simulations. Needless to say we got an extensive exposure to MATLAB and SIMULINK.

Overall, we would say this was one of the most learning oriented and well-structured assignment in our college life.

<u>Challenges Faced:</u>

- Inconsistencies in the paper and the paper told to refer
- There were 4 papers to be referred_and thus it was very difficult for us to keep up with the inconsistencies of each paper
- Moreover, one paper wasn't available online thus our simulation came to a dead end. We then too took some approximations and went ahead to simulate as much as we can.
- We are thorough with the concepts of the paper but got stuck in the implementation due to these errors in the papers

6. References:

1. Gyöngy, I. J., & Clarke, D. W. (2006). On the automatic tuning and adaptation of PID controllers. Control engineering practice, 14(2), 149-163.
2. Clarke, D. W. (2003). Pretuning and adaptation of PI controllers. IEE Proceedings-Control Theory and Applications, 150(6), 585-598.
3. Clarke, D. W., & Park, J. W. (2003). Phase-locked loops for plant tuning and monitoring. IEE Proceedings-Control Theory and Applications, 150(2), 155-169.
4. Crowe, J., & Johnson, M. A. (2002). Towards autonomous PI control satisfying classical robustness specifications. IEE Proceedings-Control Theory and Applications, 149(1), 26-31.
5. Valencia-Palomo, G., & Rossiter, J. A. (2009). Auto-tuned predictive control based on minimal plant information. IFAC Proceedings Volumes, 42(11), 554-559.
6. Wang, L. (2020). PID control system design and automatic tuning using MATLAB/Simulink. John Wiley & Sons.

7. Weightage Off Efforts:

All the group members had same efforts on basis of percentage.

Nitin Pant (33.33%)
Ayush Agrawal (33.33%)
Pranamya Jain (33.33%)