Advanced Control Systems

Prof. Somanath Majhi

Department of Electronics and Electrical Engineering

Indian Institute of Technology, Guwahati

Module No. # 04

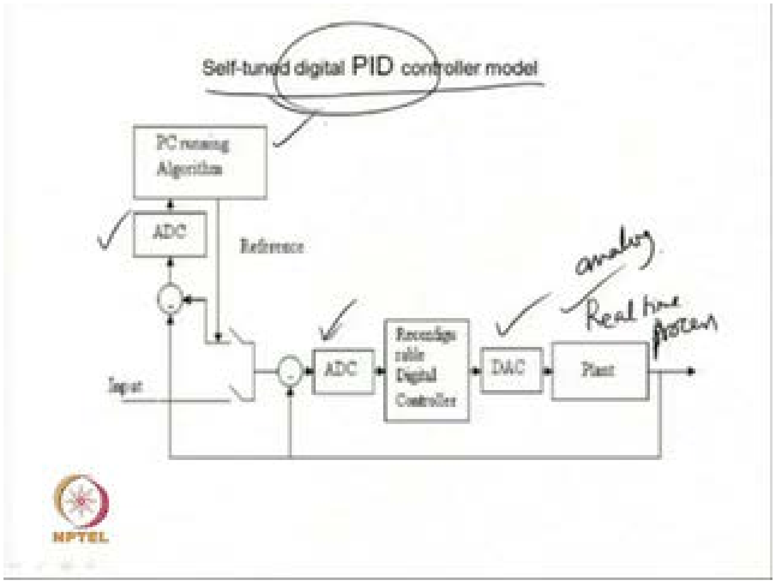
Design of Controllers

Lecture No. # 06

Tuning of Reconfigurable PID Controllers

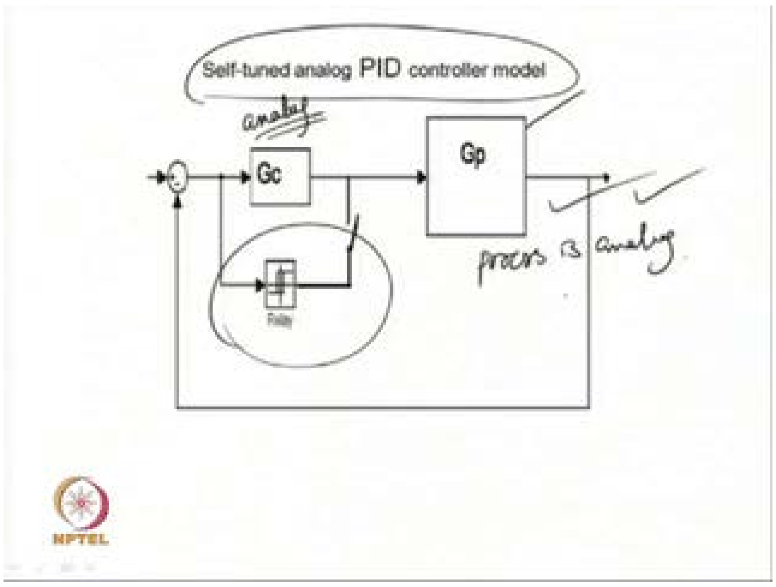
Welcome to the lecture titled the Tuning of Reconfigurable PID Controllers. PID controllers are widely used in process control industries. Digital PID controllers are used in wide spread particularly. But evolution of reconfigurable PID controllers has made a breakthrough in analog PID controller as it eliminates ADC and DAC requirements, and reduces power consumption. What is the ADC and DAC requirements? Since the process plants are analog in nature, you need to convert the analog signal to digital and digital signal to analog to operate the real time system, which is analog in nature.

(Refer Slide Time: 01:13)



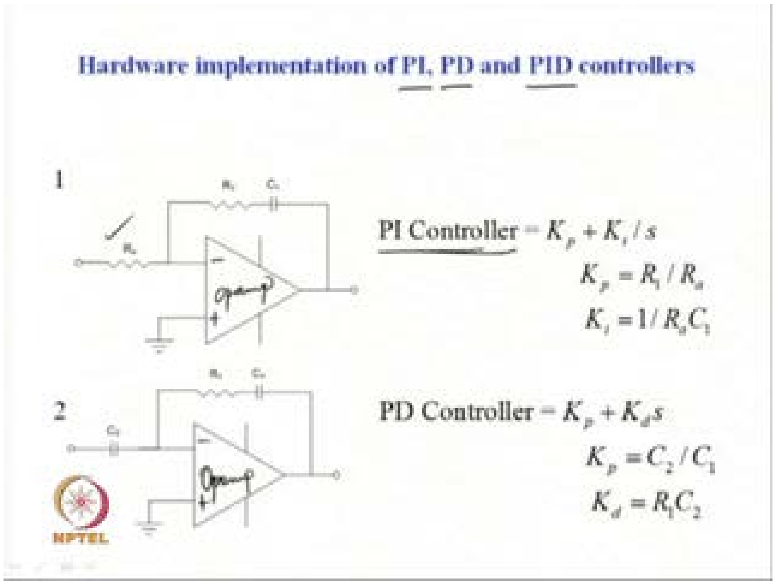
Coming to the self-tuned digital PID controller model, we see that for controlling the real time process, which is analog in nature, we need to employ analog to digital converter, digital to analog converter, analog to digital converter and personal computer for controlling the analog real time plant. So, the digital PID controller or control technique requires 1, 2, employ a number of ADCs and DACs apart from a personal computer. Is it so, when you are going to control the analog plant or analog system with the help of some analog PID controller? No, that is not so. Since the processor plant is analog, you do not require any ADC or DAC rather an analog controller can be employed to control the dynamics of the process in a straightforward manner.

(Refer Slide Time: 02:03)



Now the relay is used here to identify the dynamics of the process. So, the model for or the structure for a self tuned analog PID controller can be given like this, which requires less gadgets or equipments or components compared to that of a digital PID controller.

(Refer Slide Time: 02:51)



Hardware implementation of PI, PD and PID controllers

1

PI Controller $= K_p + K_i / s$

$K_p = R_1 / R_a$

$K_i = 1 / R_a C_1$

2

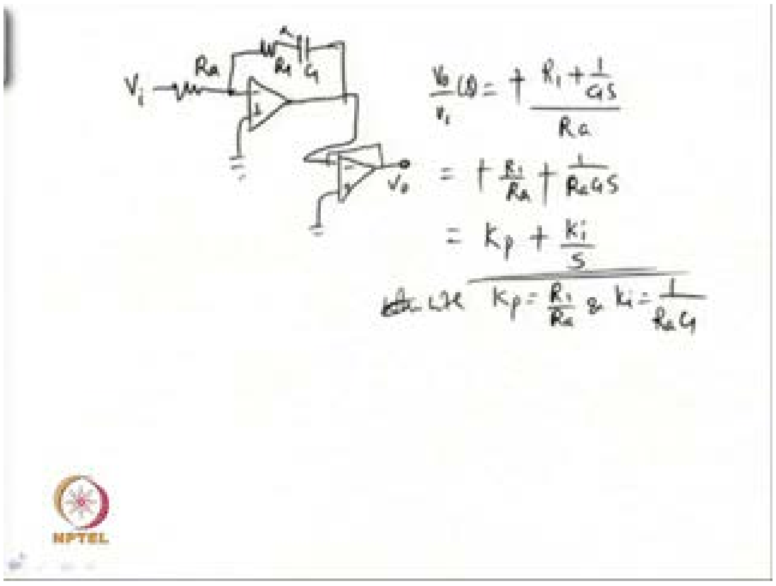
PD Controller $= K_p + K_d s$

$K_p = C_2 / C_1$

$K_d = R_1 C_2$

Now, we will see the hardware implementation of some real time PI, PD and PID controllers. How PI, PD and PID controllers can be employed or can be designed using simple electronic devices? Now, this is an Op-amp operational operational amplifier, so we have got Op-amps. Now, when an Op-amp is connected in this fashion with a resistor resistor in the input and resistance and capacitance or a resistor and capacitor connected in the feedback path, then the dynamics gives us a PI controller. How do you get that one?

(Refer Slide Time: 03:50)



$$\frac{V_o}{V_i}(s) = + \frac{R_1 + \frac{1}{C_1 s}}{R_a}$$

$$= + \frac{R_1}{R_a} + \frac{1}{R_a C_1 s}$$

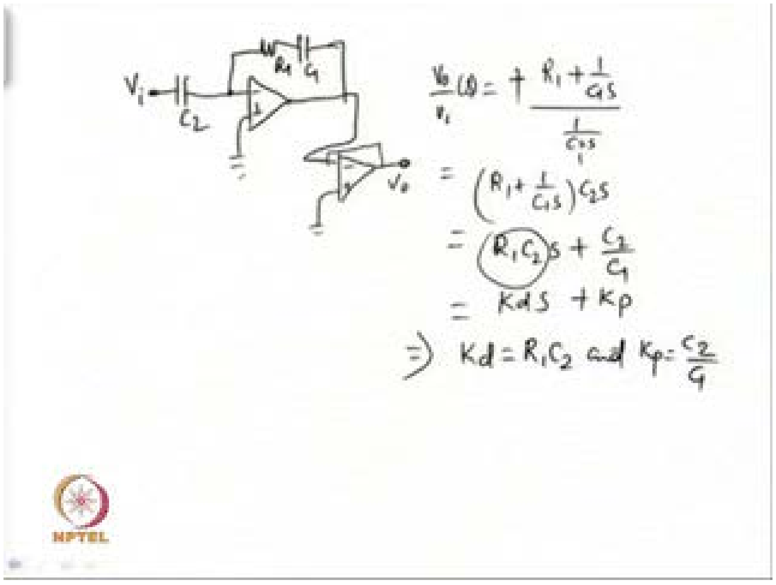
$$= K_p + \frac{K_i}{s}$$

[illegible] we $K_p = \frac{R_1}{R_a}$ & $K_i = \frac{1}{R_a C_1}$

Given an Op-amp, if the input resistance is R a, and you have got R 1 and C 1 in the feedback path, and let the input be V i and the output be V o. Then the ratio of the two in frequency domain will be minus of the impedance in the feedback path divided by the impedance in the forward path. So, minus of R 1 plus 1 upon C 1 S divided by R a; giving us minus R 1 upon R a minus 1 upon R a C 1 S. Now, how to get a PI controller? If I employ further at the output, another amplifier, which will give you unity gain, with gain of minus 1; like this, then the output you get will be plus R 1 plus 1 upon C 1 S divided by R a.

Then you will get V o upon V I (s) as R 1 by R a plus 1 upon R a C 1 S; giving us K p plus K i by S as the parameters of the PI controllers. So, this becomes a PI controller with with K p equal to R 1 by R a, and K i is equal to 1 upon R a C 1. So K p is equal to R 1 by R a and K i is equal to 1 upon R a C 1. So, this is how you design a PI controller using Op-amp resistors and capacitor. Now, similarly a PD controller can be designed using Op-amp two capacitors and one resistor.

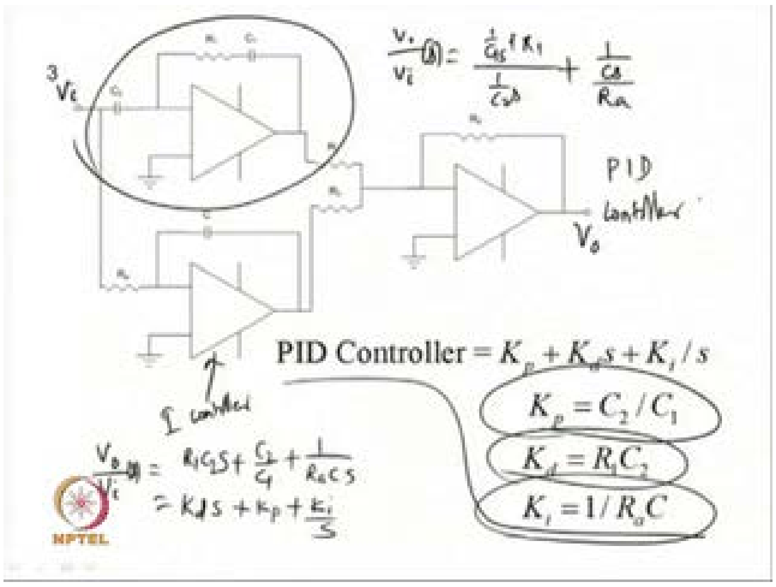
(Refer Slide Time: 06:10)



$$\frac{V_o}{V_i}(s) = + \frac{R_1 + \frac{1}{C_1 s}}{\frac{1}{C_2 s}}$$
$$= \left(R_1 + \frac{1}{C_1 s}\right) C_2 s$$
$$= \boxed{R_1 C_2} s + \frac{C_2}{C_1}$$
$$= K_d s + K_p$$
$$\Rightarrow K_d = R_1 C_2 \text{ and } K_p = \frac{C_2}{C_1}$$

So, how you will get the transfer function for that one? The transfer function for that one, when this R a is substituted by a capacitor C 2 and of course, we have got R 1 C 1 over here. Now the V o by V i (s) will be, V o is here, please keep in mind V o is here, not here V o our V i will be equal to plus R 1 plus 1 upon C 1 S divided by 1 upon C 2 S, because the impedance of the capacitor is now 1 upon C 2 S in frequency domain. Then the subsequent expression will be R 1 plus 1 upon C 1 S times C 2 s is equal to R 1 C 2 S plus C 2 by C 1. And you get this as the PD parameters given as K p plus K d S, K p S plus K d, where K d the parameter of the PD controller K d is found from the ratio of the two capacitances C 2 upon C 1; and K p is equal to R 1 C 2. So, K p no, R 1 C 2 S, K p is equal to first we will get the K d S plus K p sorry. So, K p will be equal to now, K d is equal to R 1 C 2, and K p is equal to C 2 by C 1. So, this is what you have obtained, K d is equal to R 1 C 2 and K p is equal to C 2 upon C 1. So, it is very easy to implement in hardware the PI and PD controllers.

(Refer Slide Time: 08:24)


V_i
$\frac{V_o}{V_i}(s) = \frac{\frac{1}{C_1 s} + R_1}{\frac{1}{C_2 s}} + \frac{\frac{1}{Cs}}{R_a}$
PID controller
V_o
I controller
$\frac{V_o}{V_i}(s) = R_1C_2 s + \frac{C_2}{C_1} + \frac{1}{R_a C s}$
$= K_d s + K_p + \frac{K_i}{s}$
NPTEL

PID Controller $= K_p + K_d s + K_i / s$

$K_p = C_2 / C_1$

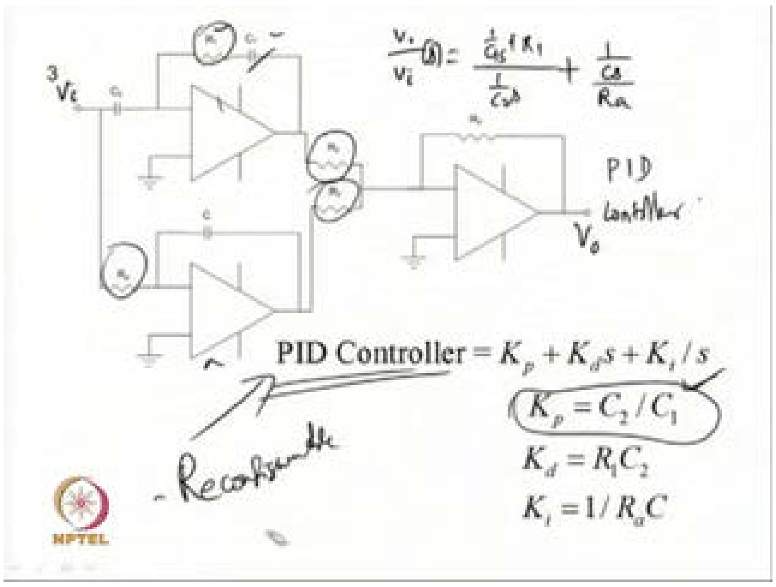
$K_d = R_1 C_2$

$K_i = 1 / R_a C$

Now, let us see how we can design a PID controller. It requires three Op-amps; this is the Op-amp you have got for summing the action of PI and PD controllers. So, where from you are getting the PID control action? This gives you the PD controller, and this gives you a p a an integral controller, so this is an I controller. So, when you add the two, you get the PID controller. So, this is the summing; this is for summing the two signals and getting you the PID control action. How do you find again this transfer function for the PID controller, is the same thing, you take the ratio of the two for PI; for this part, it will be simply your your… If the output is V o, if the input is V i, then V o upon V i (s) will be found as 1 upon C 1 S plus R 1 divided by 1 upon C 2 S plus for this, it will be 1 upon C S divided by R a. This is what you get.

So, I will collect the terms now, and write as V o by V i (s) is equal to R 1 C 2 S plus C 2 by C 1 plus 1 upon R a C S. So, this is what you get; from where, when you compare and put it in the standard form, this is your derivative controller. So, K d S plus K p plus K i by S; giving us k d as R 1 C 2, k d as R 1 C 2, K p as C 2 by C 1, and K i as K i as 1 upon R a C. So, this is how a PID controller is implemented in hardware using three Op-amps, and a number of capacitors and resistors.

(Refer Slide Time: 10:53)

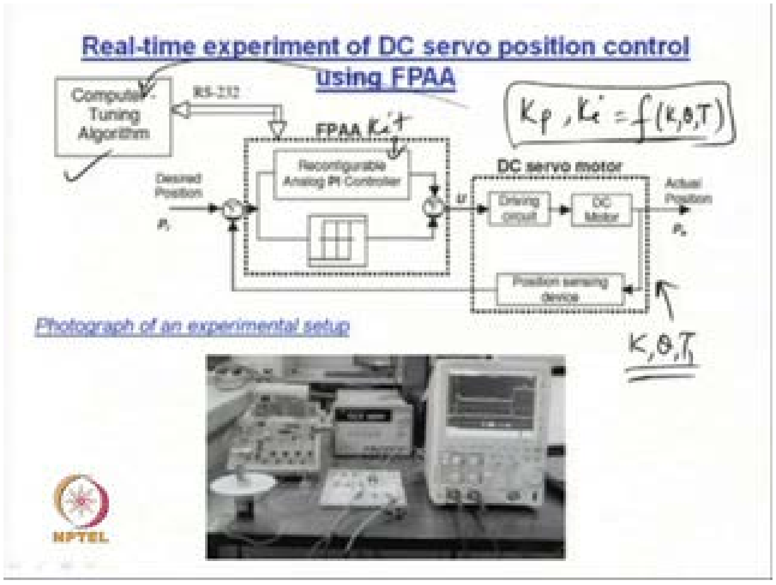


Now, interestingly you see that any gain can be obtained from the ratio of two capacitors. Is it possible to replace the resistors by capacitors? What are problem with the resistors for us? When you are going to change the values of a PID controller, you need to change the values for resistors. How can you change the values for resistor without replacing them? Is it possible to change the values of the resistor by some command or without physically replacing the resistors particularly? The capacitors you can implement very easily, if you have got variable capacitor easily, you can or if you have got a bank of capacitors, you can find various values of C 1; how can either by the help of parallel or series combinations, it is possible to get various values for C 1. But it is very difficult to get exact values for all the resistors; you have been the circuit so easily.

Is it possible to replace the resistors by some capacitors? If that is so, then our life will be very easy. And we can reconfigure the PID controller or change the PID controller parameters very easily using software or using a computer. Now a computer cannot be employed to change the R 1 values particularly, R 2 values particularly, R a values particularly; the computer can be employed to change the capacitances, capacitor values, whereas a computer cannot change the resistance values. So, if the resistors can be substituted by some capacitors, bank of capacitors or by some analog capacitor bank, then it is possible to have reconfigurable PID controller.

What we mean by reconfigurable; which parameters can be controlled by software or by some computer, you need not again assemble or bring the resistor value input, because it will take time, it will take much time to find accurate value for the resistors and substitute over here or put over connect over here. So, if it can be dynamically programmed by a computer, and if the capacitors can be used to substitute the resistors, then it is possible to design a reconfigurable PID controller. So, the focus of this lecture is that how to design a reconfigurable PID controller.

(Refer Slide Time: 13:40)


Real-time experiment of DC servo position control using FPAA
Computer Tuning Algorithm
RS-232
FPAA K_i
Reconfigurable Analog PI Controller
Desired Position
P_r
u
DC servo motor
Driving circuit
DC Motor
Actual Position
P_o
Position sensing device
$K_p, K_i = f(K, \theta, T)$
K, θ, T
Photograph of an experimental setup
NPTEL

It has been found that recently the Anadigm Corporation has introduced a device named as field programmable analog array in place of field programmable gate arrays; field programmable gate arrays are used for designing digital PID controllers; whereas field programmable analog arrays can be used for designing analog reconfigurable controller. So, reconfigurable analog PI controller or PID controller or PD controller can be implemented using FPAA blocks or using FPAA. FPAA are Field Programmable Analog

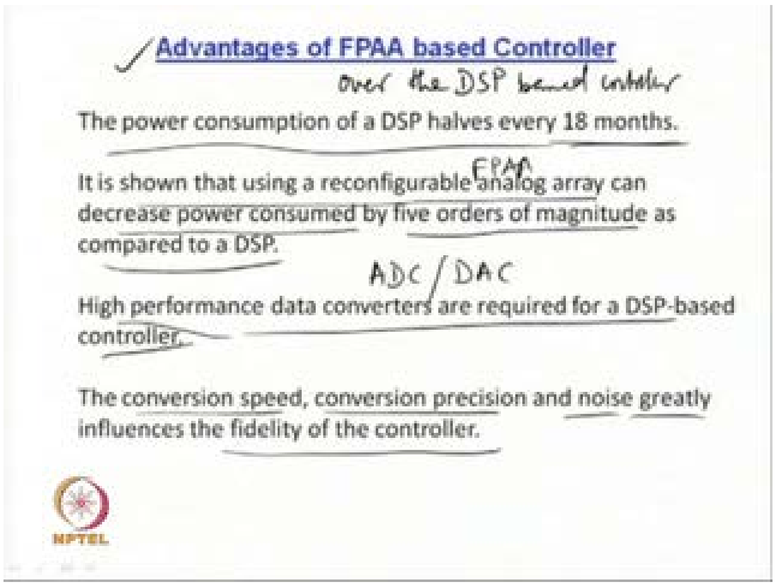
Array has got a number of configurable analog modules, which can be easily accessed by a computer, and can be configured by a computer, changed by a computer.

Now using this scheme, it is possible to design the real time systems. So, in this scheme, the control for a DC servomotor has been done using the FPAA controller sorry FPAA kit. So, the FPAA kit can provide us reconfigurable analog PI controller, also the relay dynamics can be realized, implemented by the FPAA. It is now connected through RS 232 with a personal computer, where the tuning formulae, the computation part of finding the PI controller parameters K p and K i will be done. So K p and K i are found based on the plan model parameters.

So, the this is servomotor dynamics will be identified first in terms of the K, theta and T; a first order plus transfer function model K theta and T or T 1; then based on K theta and T 1, you design the PI controller parameters. So, this formula, formula will be simulated or will be implemented in a computer. It is difficult to have computational facility in FPAA rather you can reconfiguration facilities are there, you can reconfigure anything. But you may not be able to compute or find or estimate values. So, that will be taken care of by the personal computer.

So, this is the photograph of an experimental set up, we have in our lab, where we have used the FPAA kit the FPAA kit along with the signal conditioning circuit, and various measuring test, and measuring equipments, and this is the DC servomotor, which is under control now. So, this experimental setup has been developed to see whether the FPAA can be applied to retune the parameters of a PI controller or to design a controller for a real time systems such as DC servomotor.

(Refer Slide Time: 17:18)



Advantages of FPAA based Controller

over the DSP based controller

The power consumption of a DSP halves every 18 months.

It is shown that using a reconfigurable $\overset{\text{FPAA}}{\text{analog}}$ array can decrease power consumed by five orders of magnitude as compared to a DSP.

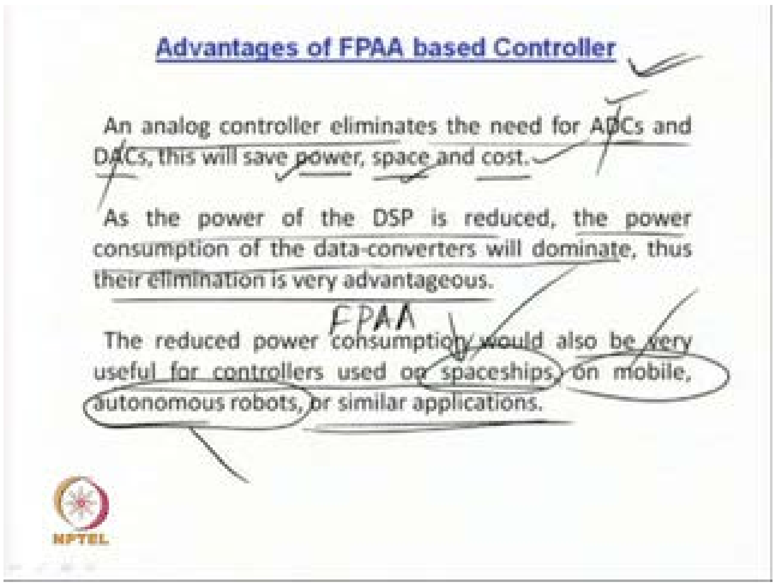
ADC / DAC

High performance data converters are required for a DSP-based controller.

The conversion speed, conversion precision and noise greatly influences the fidelity of the controller.

Now, I will talk about the advantages of FPAA based controller over the DSP based digital signal processors based controller. The power consumption of DSP halves every 18 months, this is one demerit it has, it is shown that using a reconfigurable analog array that means FPAA, we can decrease power consumption by five orders of magnitude as compared to a DSP. So, the digital controllers are easy to implement, but they are subjected to cost in, because they consume more power. High performance data converter converters like such as analog to digital converter or digital to analog converters are required for a DSP based PID controller. Those are not required for FPAA based analog controller. The conversion speed, conversion precision and noise greatly affect the fidelity of the digital controller, that is not so in case of analog controllers.

(Refer Slide Time: 18:40)



Advantages of FPAA based Controller

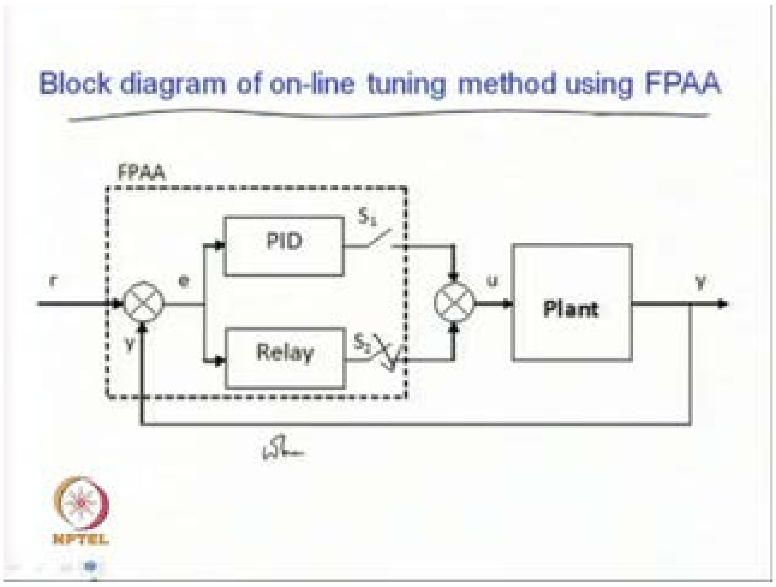
An analog controller eliminates the need for ADCs and DACs, this will save power, space and cost.

As the power of the DSP is reduced, the power consumption of the data-converters will dominate, thus their elimination is very advantageous.

The reduced power consumption would also be very useful for controllers used on spaceships, on mobile, autonomous robots, or similar applications.

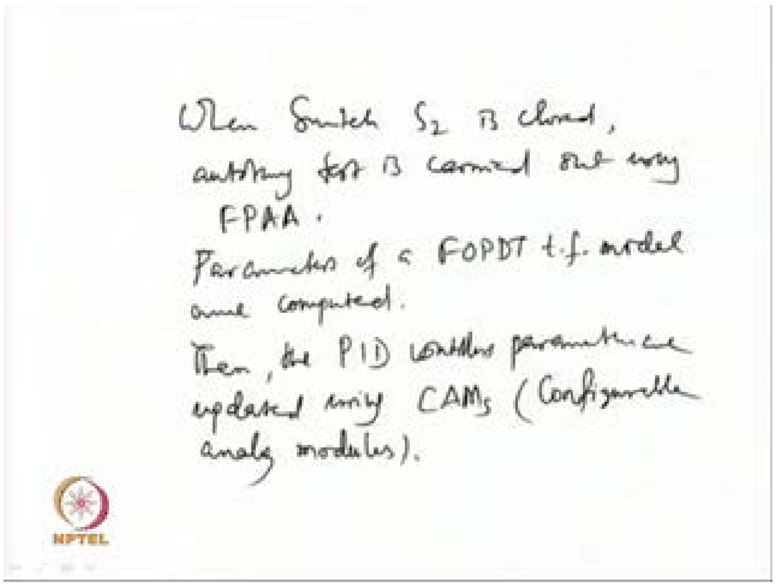
Further advantages of FPAA based controller are that the analog controller eliminates the need for converters particularly, which in turn result in saving of power, space and cost. You do not require analog to digital, digital to analog converters therefore, you save cost, you save space also, and you save power, because powers are consumed by the converters. As the power of the DSP is reduced, the power consumption consumption of the data-converter will dominate; thus there elimination is very advantageous. That would also be very useful for controllers used on which one, the FPAA based controllers can find used on especially on spaceships, where you have constant on space; on mobile phones size has to be small; now autonomous robots, because you you have to have less number of ADCs and DACs and the cost will come down or many similar applications. So, FPAA finds application particularly in spaceships, mobile phones, autonomous robots and many more applications.

(Refer Slide Time: 20:17)


Block diagram of on-line tuning method using FPAA

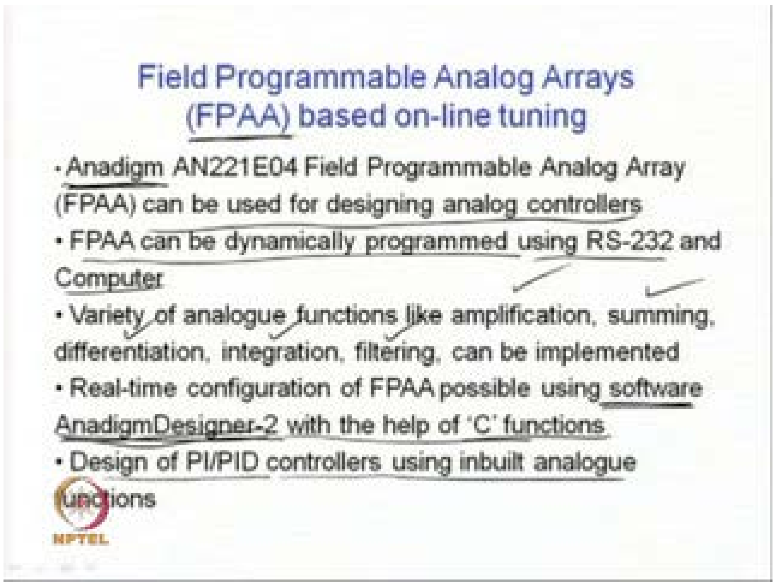
Now, how the FPAA is implemented; we will see. This is the basic block diagram of online tuning scheme for a PID controller using FPAA. Initially, what is done? The switch 2 is closed; then the plant parameters are identified.

(Refer Slide Time: 20:42)


When Switch S_2 is closed, autotuning test is carried out using FPAA.
Parameters of a FOPDT t.f. model are computed.
Then, the PID controller parameters are updated using CAMs (Configurable analog modules).

When switch when switch S 2 is closed, then auto tuning test auto tuning test is carried out using of course, FPAA. Then parameters of parameters of a first order plus dead time transfer function model are computed. Then the PID controller parameters parameters are updated using the CAMs; what are CAMs? Configurable Configurable Analog Modules; this is how online tuning is done using FPAA.

(Refer Slide Time: 22:04)

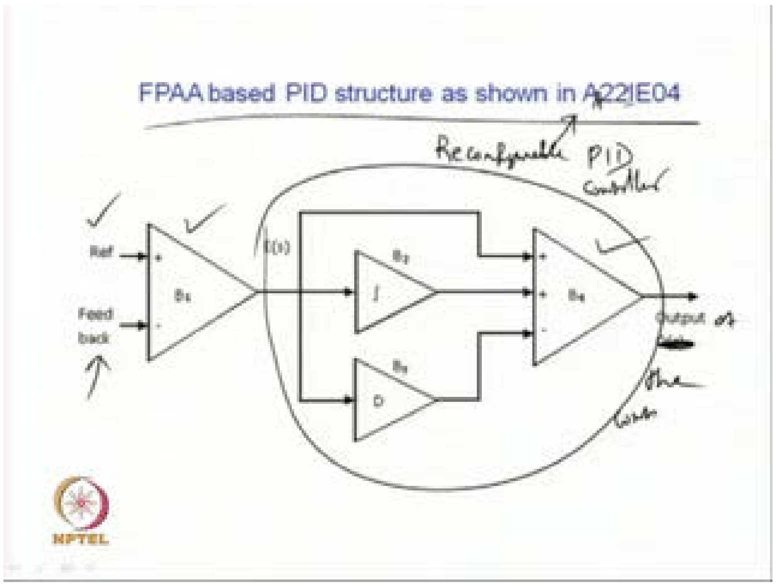


Field Programmable Analog Arrays (FPAA) based on-line tuning

- Anadigm AN221E04 Field Programmable Analog Array (FPAA) can be used for designing analog controllers
- FPAA can be dynamically programmed using RS-232 and Computer
- Variety of analogue functions like amplification, summing, differentiation, integration, filtering, can be implemented
- Real-time configuration of FPAA possible using software AnadigmDesigner-2 with the help of 'C' functions
- Design of PI/PID controllers using inbuilt analogue functions

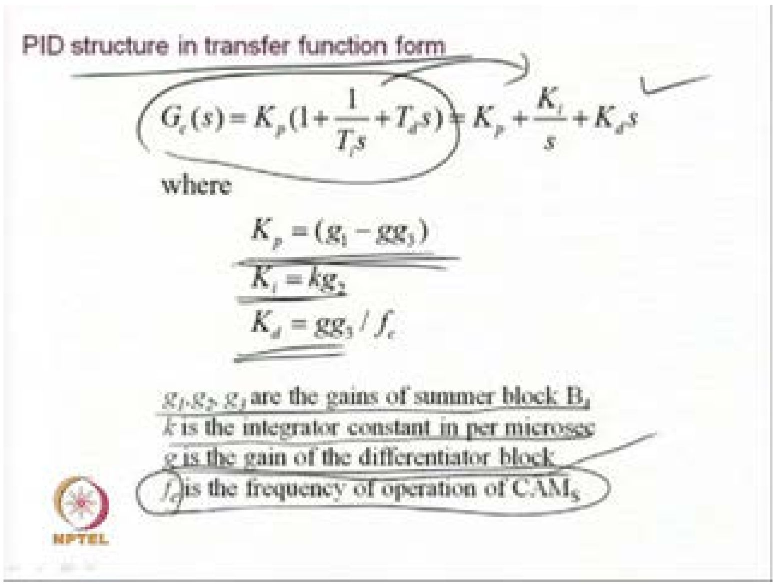
Now, what is that FPAA? Let us discuss a little bit about the device. This FPAA product of the company Anadigm, Anadigm Corporation AN221E04 can be used for designing analog controllers. FPAA can be dynamically programmed using RS-232 and a computer, a PC. Variety of analog functions like amplification, summing, differentiation integration, filtering can be implemented in it. Now real time configuration of FPAA is also possible using dedicated software such as AnadigmDesigner-2 we have used this software for real time configuration of the PID controller and PI controller in our lab; and this software runs with the help of C functions also. Now, design of PI or PID controllers can be done using the inbuilt analog functions.

(Refer Slide Time: 23:26)


FPAA based PID structure as shown in A221E04
Reconfigurable PID Controller
Ref
Feed back
B1
E(s)
B2
∫
B3
D
B4
Output of the contr

Now, this gives the basic structure of a PID controller as implemented in the Anadigm AN, Anadigm is AN will be there, so AN221E04. So, you will have blocks like this in the FPAA kit, where the first block B 1 collects reference inputs and the feedback signal. The signal is processes processed by the controller now. The controller is implemented in this fashion, and the controller pass on signal to another summing block, which ultimately gives you a reconfigurable configurable PID controller. Now different blocks have got different gains, the blocks are implemented with the help of gains. So, this is the output of the controller, not the final output, output of the controller.

(Refer Slide Time: 24:42)



PID structure in transfer function form

$$G_c(s) = K_p\left(1+\frac{1}{T_i s}+T_d s\right) = K_p + \frac{K_i}{s} + K_d s$$

where

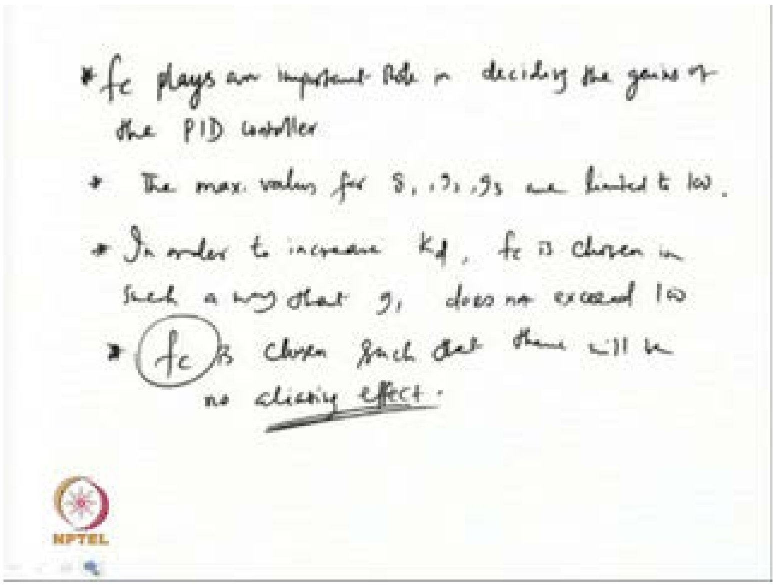
$$K_p = (g_1 - g g_3)$$
$$K_i = k g_2$$
$$K_d = g g_3 / f_c$$

g_1, g_2, g_3 are the gains of summer block B_4
k is the integrator constant in per microsec
g is the gain of the differentiator block
f_c is the frequency of operation of CAM_s

Now, the PID controller given in the standard form such as K p times 1 plus 1 upon T i s plus T d S can be expressed as G c (s) is equal to K p plus K i upon S plus K d S, where look at carefully the way the gains are found now, using the gains of the individual blocks we have here. How many blocks are there? There are four blocks 1, 2, 3, 4 each blocks have their own gains or constants that can be used.

So, K p is implemented as K p is equal to g 1 minus gg 3; K i is equal to kg 2; K d is equal to gg 3 upon f c; what are g(s)? g 1 g 2 g 3 are the gains of summer block 4. So, the summer block has got g 1, g 2 and g 3, the gains like this; the summer block has gains like this. The k is the integrator constant in per microsecond; g is the gain of the differentiator block, so the differentiator block has a gain of g; and f c is the frequency of operation of the configurable analog modules; this frequency is very important for us, because unless this frequency is set accurately or properly, you will land up in trouble and the real time implementation of FPAA will have some real difficulties.

(Refer Slide Time: 26:41)

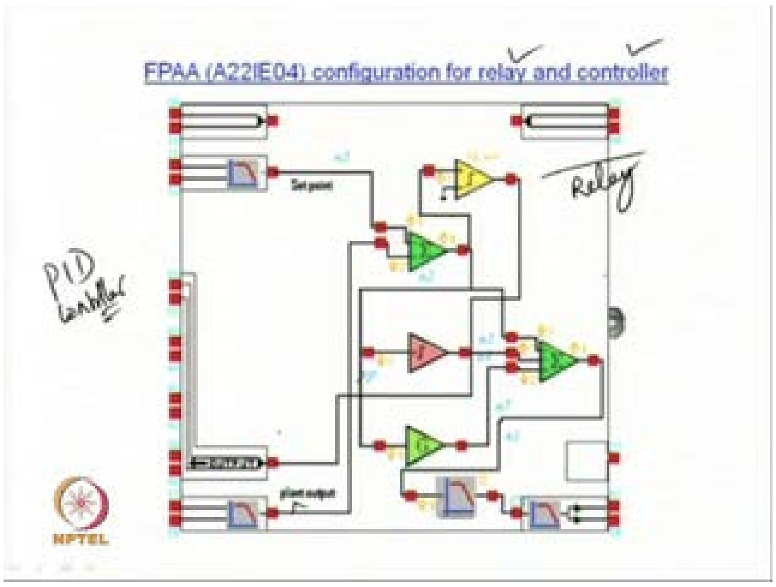


* f_c plays an important Role in deciding the gains of the PID controller
* The max. values for g_1, g_2, g_3 are limited to 100.
* In order to increase K_d, f_c is chosen in such a way that g_1 does not exceed 100
* $\boxed{f_c}$ is chosen such that there will be no clicking effect.

Now, how to choose the f c? And what are the effects of f c? The frequency of operation of the analog modules or blocks f c plays an important role role in deciding the gains of the PID controller. How that is so, the maximum value the maximum values for g 1, g 2 and g 3 are limited to 100. Therefore, in order to increase in order to increase K d, the derivative constant you have if the if the PID controller f c is chosen, in such a way that way that way that g 1 does not g 1 does not exceed 100. Otherwise what happens, if you look at this one, g is our gg 3; gg 3 is equal to K d f c; therefore, K p can be written as g 1 minus K d f c. So, if f c is not chosen properly, then we may not get proper value of K d, and we may get an ultimate value, which is greater than g 1 rendering a negative proportional gain of the PID controller.

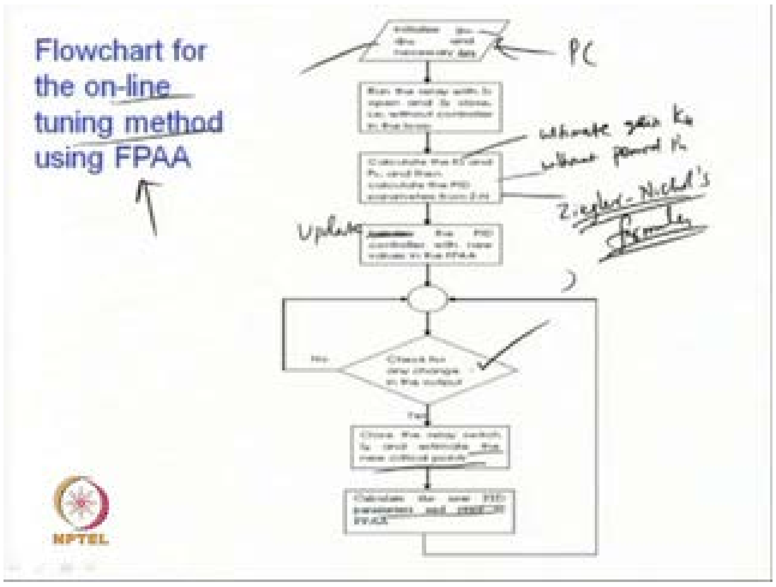
Then what will happen to the system, when you have got a negative gain of the PID controller, the controller will not act rather you will develop an unstable closed loop control system, that is the demerit associated with the choice of K f c. So, please be careful in choosing f c particularly; and proper care must be choosing f c, choose f c in such a way that there will be no aliasing effect. So, you are when you process the signal, you will be subjected to aliasing effects that can be also overcome with proper choice of the f c, the frequency of operation of the configurable analog modules or blocks.

(Refer Slide Time: 29:37)



Now, I will go to little bit of the FPAA configuration; this is the hysteresis we have assumed in our relay, this is the relay is implemented in the FPAA, and this is your PID controller implementation. So, we have the PID controller realized in this fashion, where this is the plant output is collected here through this port, and this output means, this is not the plant output, this is not the plant output. Be careful, this is the output you are getting from the relay. So, relay output is basically collected and given or collected and given to proper point proper node. So, this is the real time implementation of a FPAA configuration having relay and controller. It is unless you go through and see the kit, it will be difficult for you to follow these symbols what are shown over here, different modules, but please keep in mind, we have got a number of camps configurable analog blocks in it.

(Refer Slide Time: 31:13)



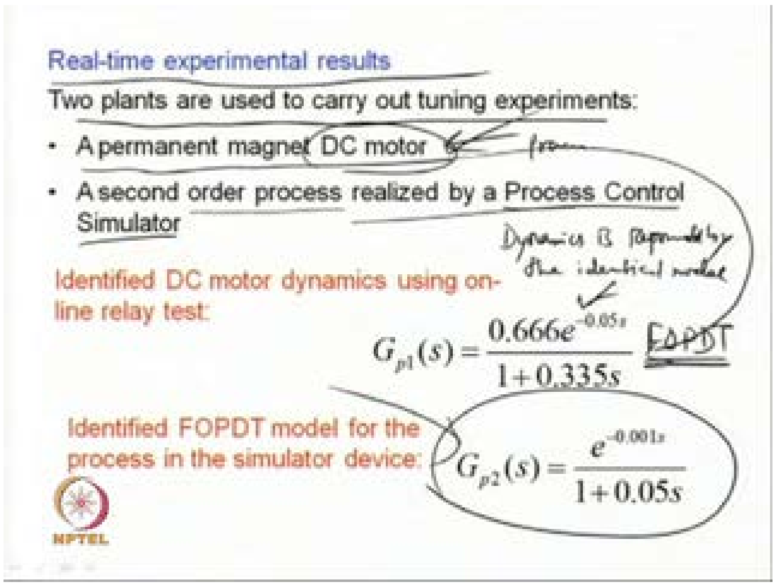
Now, I will come to again the block diagram of on-line tuning using FPAA; having shown this, we will go through some flow chart that gives you the way, the FPAA is used to design reconfigurable PID controllers. So, the flow chart for on-line tuning method is given over here. Initialize the gain margins, and phase margins, and necessary data; where you do this data processing? This is done in the PC, in the personal computer. As you see, your system is connected with a computer, which will compute the gains of the or parameters of the controller, configurable controller.

Now, then run the relay with S 1 open and S 2 closed that is without controller in the loop. Calculate the ultimate gain K u, and ultimate period P u, and then calculate the PID parameters using Ziegler-Nichols formula. So, we have in the literature, we have formula, many formulas given by Ziegler-Nichols, so use that one to find the PID parameters. Now, initialize the PID controller with new values that means update, not initialize, it will be update the PID controller with new values in the FPAA. So, with certain commands, it is possible to reconfigure the PID controller with the new values of new calculated values.

Now go to the decision block; check for any change in the output particularly, when this tuning will be initiated, the output will be tracked continuously. So, this is your output, system output, so the output will be tracked continuously by the FPAA; and as and when the output increases or decreases above or below certain threshold value, then the auto tuning is invoked; the relay is invoked, and then the auto tuning process starts, and the updation of PID parameters are carried out till the output is acceptable to you.

Now, check for any change in the output; if it is yes, close the relay switch that means, go for the auto tuning test, and estimate the new critical points; calculate the new PID parameters and send it to FPAA. And if the output is within bounds, then do not go for any action that means the FPAA will remain silent or the… The FPAA will not remain silent; updation of PID parameters will not take place, when the output is within the tolerable limits or if the output is satisfactory.

(Refer Slide Time: 34:49)



Real-time experimental results

Two plants are used to carry out tuning experiments:

- A permanent magnet DC motor ← from
- A second order process realized by a Process Control Simulator

Dynamics is represented by the identical model ✓

Identified DC motor dynamics using on-line relay test:

$$G_{p1}(s) = \frac{0.666e^{-0.05s}}{1+0.335s}$$ FOPDT

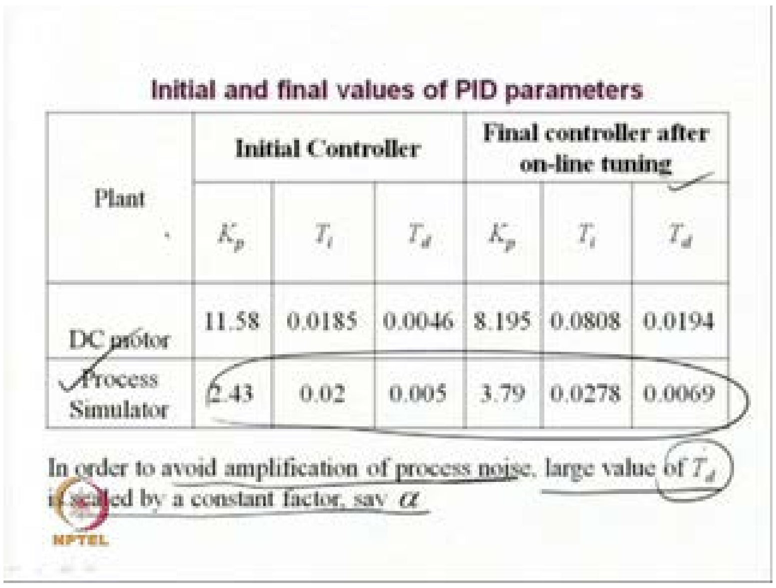
Identified FOPDT model for the process in the simulator device:

$$G_{p2}(s) = \frac{e^{-0.001s}}{1+0.05s}$$

Now, I will describe now, the results of two real-time experimental setups. So, two plants were used to carry out tuning experiments; one plant was a permanent magnet DC motor PMDC, and the second one was realized by a process control simulator. So, thus we have got two plants or processes a DC motor as process one, which dynamics was identified using the relay test as G p1 (S) G p 1 (S) is equal to 0.666 e to the power minus 0.05S divided by 1 plus 0.335 S. So, the DC motor dynamics the PMDC motor dynamics is represented by the identified model G p1 (S). So, we have obtained a first order plus dead time model for the real-time system a DC motor. The DC motor can be, the dynamics of a DC motor can be given by, obtained by this first order plus dead time transfer function model.

Similarly the process realized by the process control simulator was found to be giving a transfer function of the form G p2 (S) is equal to e to the power minus 0.001 S divided by 1 plus 0.05 S; you carefully, please carefully observe the time constant values and the time delays, those are very small values, those are very small values and it is difficult to implement those values using Op-amp and resistors capacitors. So, use of FPAA enables us to design controllers, reconfigurable controllers for real-time plants such as a DC motors, and a one realized by a process control simulator.

(Refer Slide Time: 37:27)



Initial and final values of PID parameters

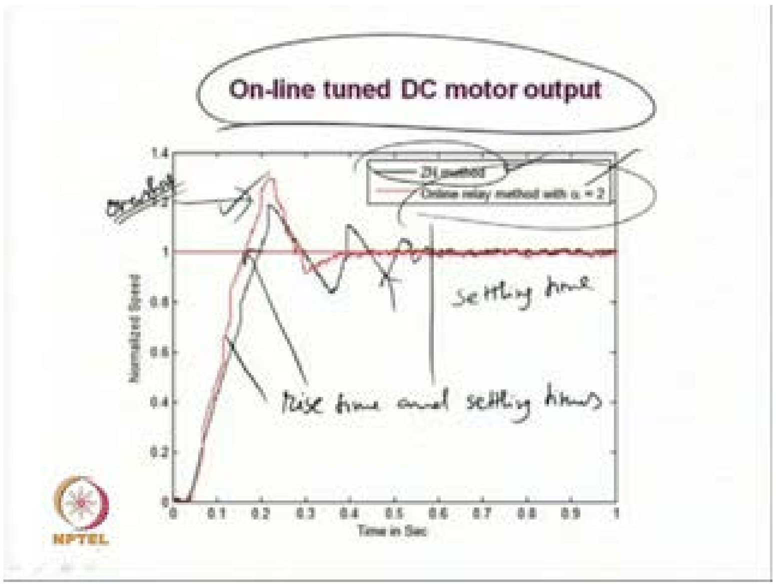
Plant	Initial Controller			Final controller after on-line tuning		
	K_p	T_i	T_d	K_p	T_i	T_d
DC motor	11.58	0.0185	0.0046	8.195	0.0808	0.0194
Process Simulator	2.43	0.02	0.005	3.79	0.0278	0.0069

In order to avoid amplification of process noise, large value of T_d is scaled by a constant factor, say α

Now, the initial and final values of PID parameters are tabulated here in this table; for the DC motor, initial controller parameters are K p is equal to 11.58, T i is equal to 0.0185 and T d is equal to 0.0046. How do you find the initial controller parameters? Using the Ziegler-Nichols method; so perform a relay test and obtain the initial controller parameters using the formula given by Ziegler and Nichols. Then, make use of r tuning formula, we have discussed a number of PID, PIPD, PDPI controller tuning formula, so make use of r tuning formula and estimate the final controller after auto tuning, and those are found to be K p, T i and T d as 8.195, 0.0808, 0.0194 respectively.

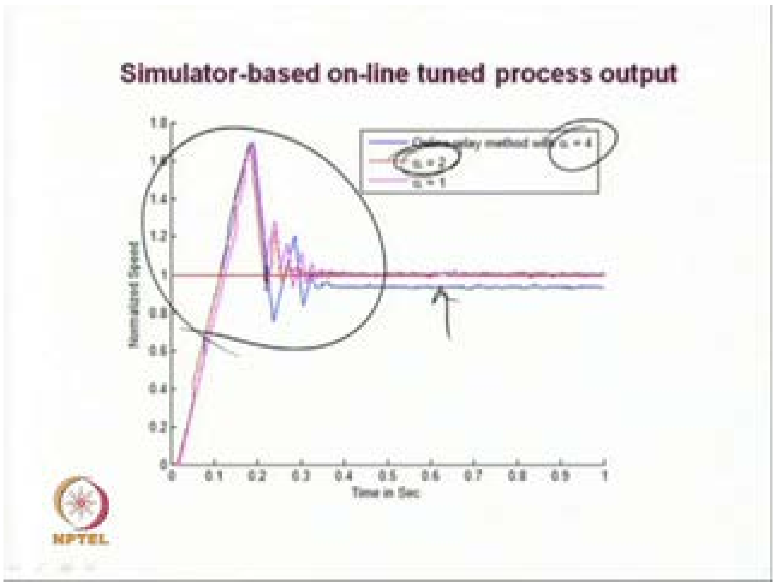
So we have found these PID parameters after conducting a relay test with the initial controller values given by K p, T i and T d as shown over here. So, the the the final controller is obtained after second stage of the relay auto tuning test. Similarly, the controller parameters for the process simulator are obtained and tabulated here. In order to avoid amplification of process noise, the measurement noise, large value of T d should be avoided; how can you avoid large values of T d? It can be scaled by a constant factor say alpha. So, choose certain alpha value to scale down T d values, so that you can reduce the measurement, noise during real-time experiment auto tuning test and control.

(Refer Slide Time: 40:03)



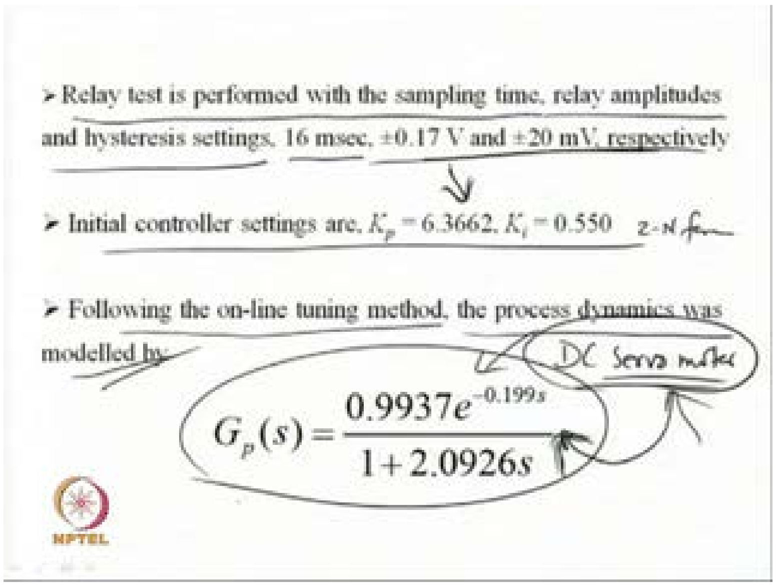
Now, what sort of output we did get? The final output when the reconfigurable PID controller was employed; the Ziegler-Nichols PID controller gives us the one shown by the black. Now this has got a long settling time settling time, but the method we have adopted with the choice of alpha equal to 2 results in PID parameter such that we have got a time response of this form given by the red one. Now this time response is although subjected to a little bit of larger overshoot, the overall performance in terms of improvement in rise time, and improvement in settling times are concerned; we have got superior performance compared to that given by the Ziegler-Nichols method.

(Refer Slide Time: 41:14)



So, simulator has also given similar results. We have got improved performances, we have not made made any comparison of results over here; for different values of alphas of course, now the alpha chosen should be 2 as it shows, you have if you go for higher value of alpha also, you have got steady state error; when alpha is two, we get suitable time response from the closed loop system.

(Refer Slide Time: 42:04)



- Relay test is performed with the sampling time, relay amplitudes and hysteresis settings, 16 msec, ±0.17 V and ±20 mV, respectively
- Initial controller settings are, $K_p = 6.3662$, $K_i = 0.550$ Z-N form
- Following the on-line tuning method, the process dynamics was modelled by:

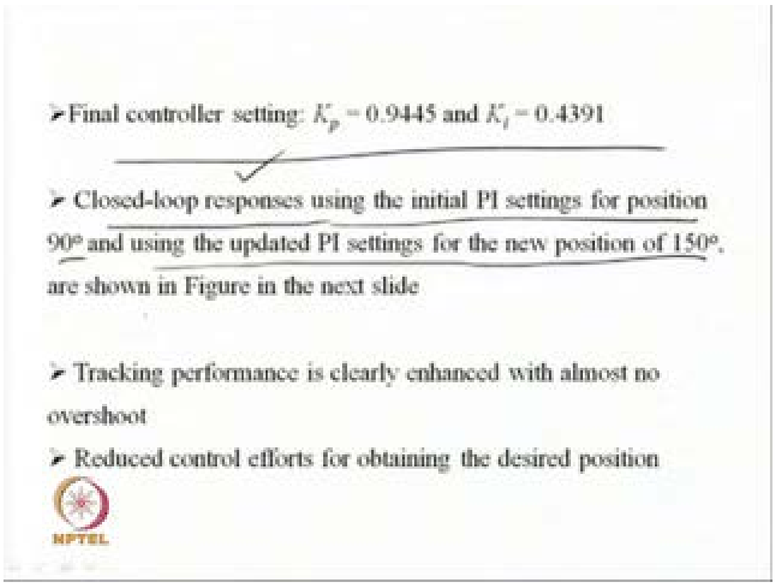
$$G_p(s) = \frac{0.9937e^{-0.199s}}{1+2.0926s}$$

DC Servo motor

Now, real-time experiment of a DC motor, servomotor is now carried out using a reconfigurable PI controller. In the second case, the hardware setup is shown over here, then when it is also a relay test is performed to with the sampling time, relay amplitudes and hysteresis settings of 16 milliseconds, plus minus 0.17 volts and plus minus 20 millivolts respectively. We are able to find the initial control controller settings as K p is equal to 6.3662, K i is equal to 0.55 with the help of Ziegler-Nichols formula; of course, after obtaining the limit cycle parameter such as the critical gain and critical frequency.

Now, following the on-line tuning identification method; on-line identification method, the process dynamic dynamics was estimated as G p (S) is equal to 0.9937 e to the power minus 0.199 S divided by 1 plus 2.0926 S. So, this is the dynamics of a DC servomotor that means, a real-time system such as a DC servomotor can be described by this transfer function. So, whatever output you get from this transfer function will be the one you get from a real-time DC servomotor, when appropriate signal is applied. This is the transfer function model of the dynamics of a DC servomotor.

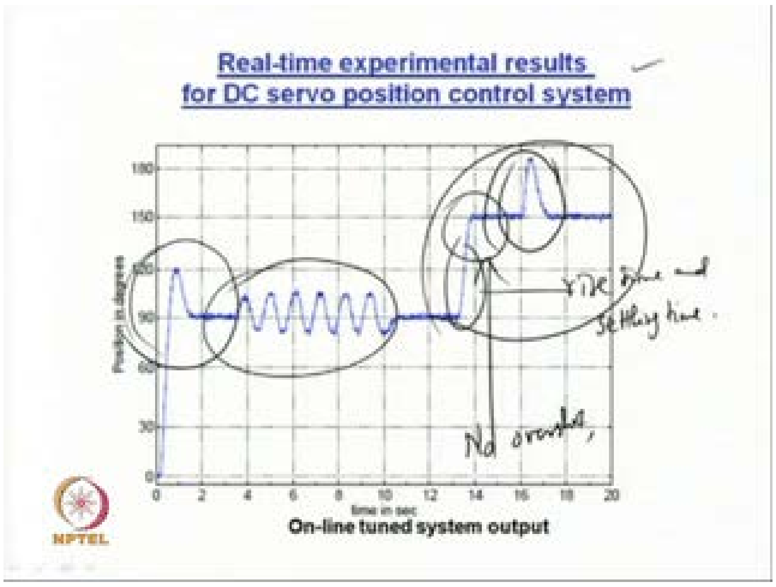
(Refer Slide Time: 43:37)



- Final controller setting: $K_p = 0.9445$ and $K_i = 0.4391$
- Closed-loop responses using the initial PI settings for position 90° and using the updated PI settings for the new position of 150°, are shown in Figure in the next slide
- Tracking performance is clearly enhanced with almost no overshoot
- Reduced control efforts for obtaining the desired position

Now, final controller settings are done as K p is equal to 0.9445, and K i is equal to 0.4391; now closed loop responses using the initial settings for position 90, and using the updated PI settings for the new position of 150 degree; what are these positions I mean, we are having a DC servo motor, and we are going to control the position of the shaft. So, for different reference values, we have got closed loop responses and those responses are shown in the figure.

(Refer Slide Time: 44:13)

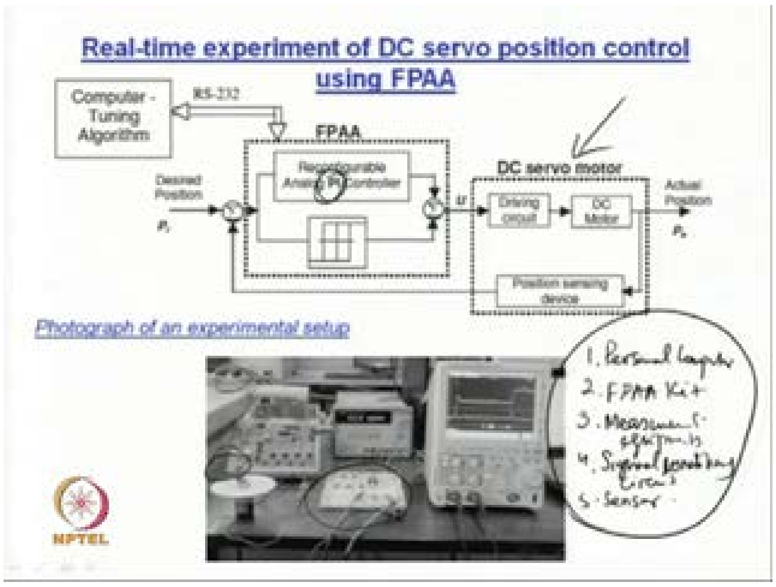


This is the time responses we get from the DC servo position control system. Now as you see this is the normal operation; during the normal operation, we get the response; during the auto tuning test, we get the time response as this one. And when the position is changed from 90 degree to 150 degree 150 degree, you see 150 degree all these are in degrees then we get a time response of this form. Now after some time intentionally, we inject some disturbance; that means what? You change the shaft position and leave, then automatically the shaft will come back to the original position, and that is shown by this time response. So, given a load disturbance means change it, then you will leave it automatically, the shaft will come back to its original position. So this shows that the developed control system can successfully reject static load disturbances.

Tracking performances is clearly enhanced with almost no overshoot. See if I look at this performance given by the updated PI controller, then there is no overshoot over here, no overshoot, whereas we have got quite satisfactory, rise time and settling time. So, I can say we have got overall satisfactory performances given by the reconfigurable PI controller as far as controlling the position of, shaft position of a DC servo motor is concerned. This is what we get the beauty the beauty of this technique FPAA based controller technique.

We have now coming to another point that we do get also reduced control efforts for obtaining the desired position. So the control efforts have been not been shown here, that is been seen that in terms of utilization of power, in terms of use of hardware hardware and in terms of space and so we have got benefits. So, that way it is said that reduced control efforts are made for obtaining the desired position, which is now 150 degree from the initial position of 90 degree. So, we started with 90 degree, keep in mind; and we changed it to 150 degree, and thus we got a time response of this form.

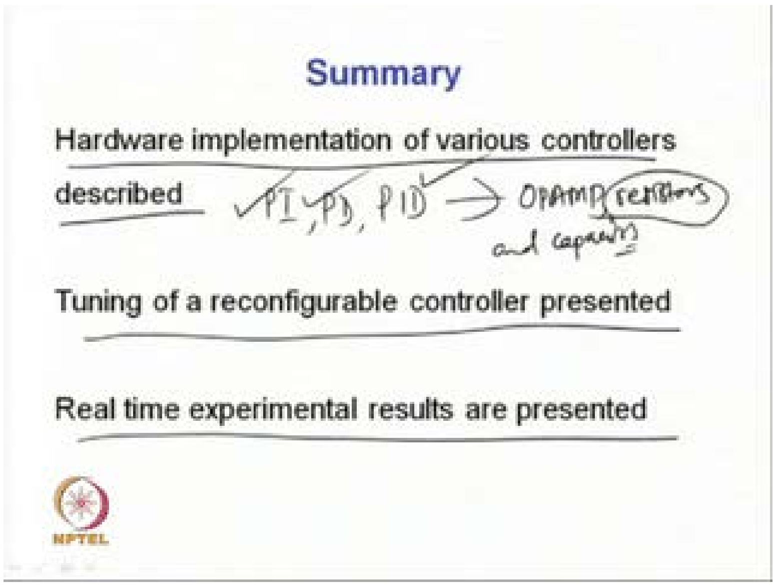
(Refer Slide Time: 47:38)



Real-time experiment of DC servo position control using FPAA

Now, I will go back to the hardware setup little bit. The hardware setup has got a personal computer, so requirements for this FPAA based reconfigurable controller are a personal computer Anadigm FPAA kit, then you have got measurement equipments, then signal conditioning circuit, you have to develop a signal conditioning circuit signal conditioning circuit, then your sensor obviously, we have used it is not visible clearly, a tacho generator, so sensor and of course, you have got some ic(s). So, these are the basic requirements one needs, we one has for implementing real-time FPAA based controller.

(Refer Slide Time: 48:48)



Summary

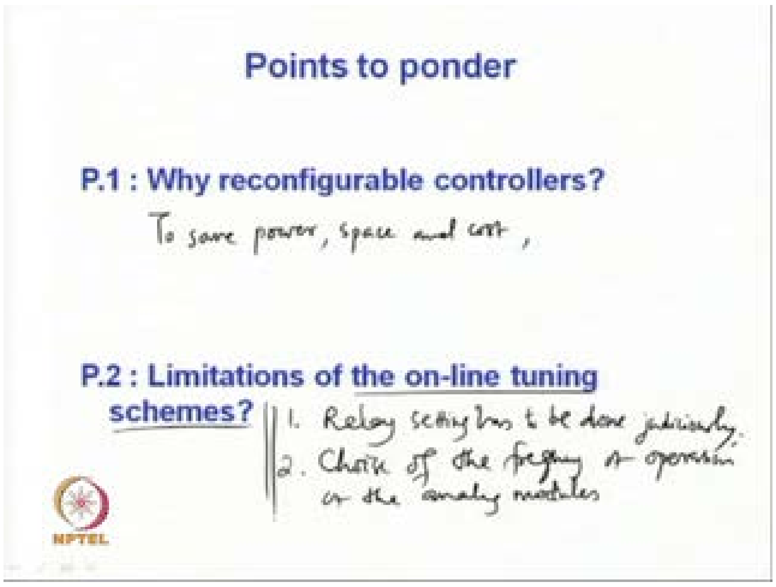
Hardware implementation of various controllers described ✓PI, ✓PD, PID✓ → OPAMP, resistors and capacitors

Tuning of a reconfigurable controller presented

Real time experimental results are presented

I will come to the summary now. So, hardware based implementation of various controllers has been described. So, a PI, PD, PID controllers can be realized using Op-amp resistors and capacitors. But there are certain limitations, like it is very difficult to change the resistors value particularly, and that requires us to go for reconfigurable PI, PD and PID controllers. Tuning of a reconfigurable controller is presented in the lecture and also real-time experimental results are presented to show that it is not difficult to design FPAA based reconfigurable PID controllers.

(Refer Slide Time: 49:58)



Points to ponder

P.1 : Why reconfigurable controllers?

To save power, space and cost,

P.2 : Limitations of the on-line tuning schemes?

1. Relay setting has to be done judiciously.
2. Choice of the frequency of operation or the analog modules

Why reconfigurable controllers? As you have seen to save power, space, and cost, one can go for the use of FPAA based controller then the use of DSP based controllers. Are there any limitations of the on-line tuning schemes? Yes, one has to be very careful when you are doing real-time experiments particularly, you need to carefully set the relay parameter. So, relay setting has to be has to be done accurately or I would say judiciously; secondly, the choice of f c, choice of the frequency of operation of the analog modules. These are the two things you must take care of while designing a FPAA based reconfigurable PID controller. Thanks.