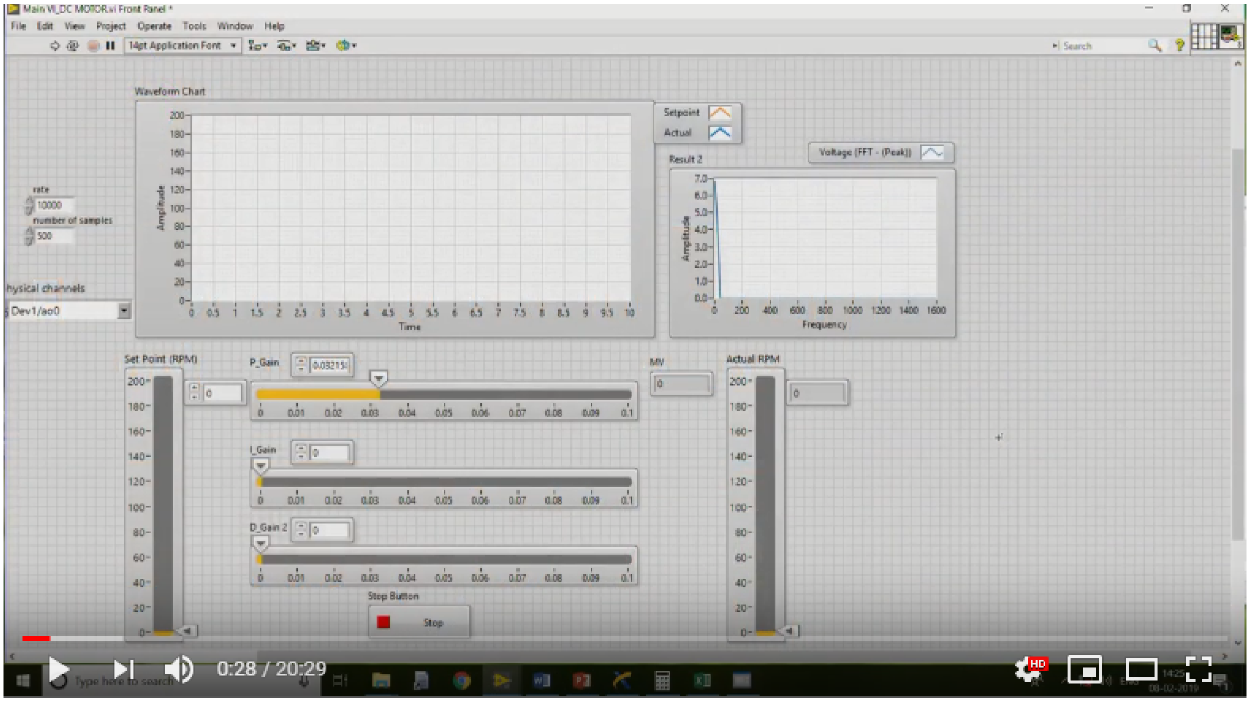
Lecture –33
Design of Speed Control of a DC Motor using DAQ: Part 3

Electronic Modules for Industrial Applications using Op-Amps

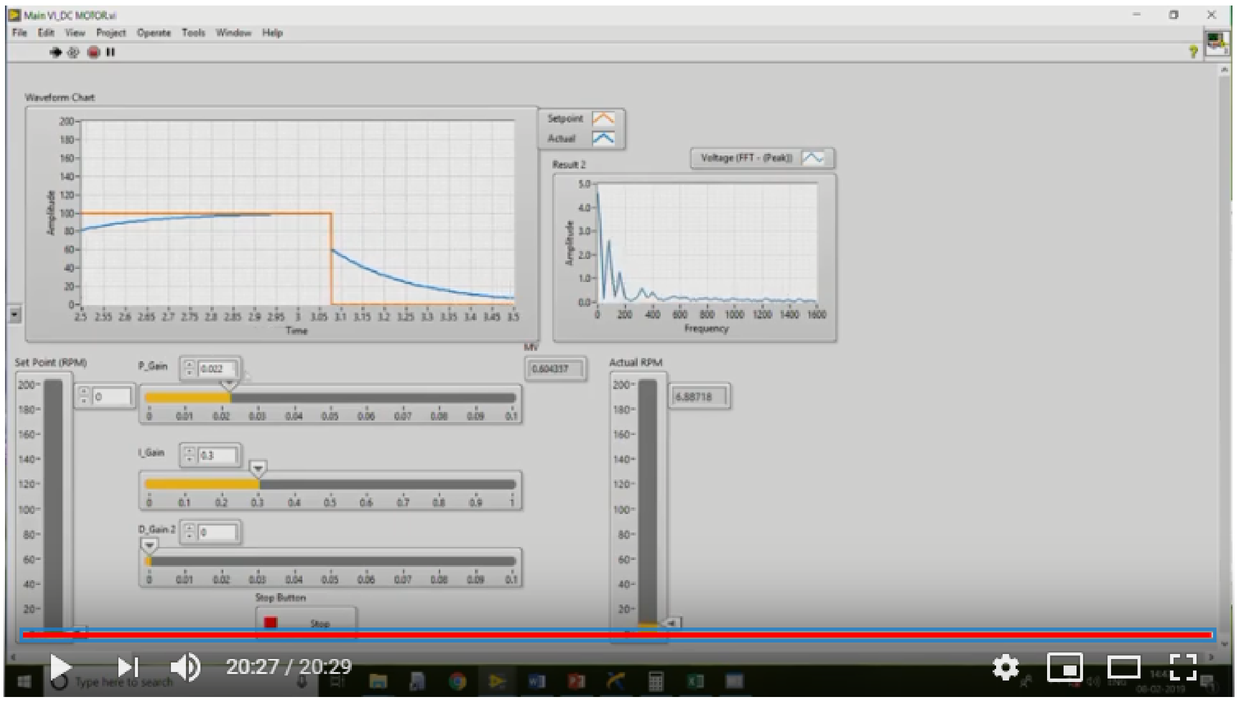
Video Start Time: (00:28)



So, now what we will be doing is that right now the gain, let me set the gain to 0.05 and so, Okay? 0.05 and if you see that the I-gain is 0, D-gain as well as 0. So, which means that, right now the controller which is being use is a P-controller. So, let me run and since a set point is 0, when you look into the motor, even the motor is not working. So, what I will be doing is that I change the RPM to somewhere around 50. So, we can see the orange indicates the set point, the blue indicates our, the actual value, and when we see the motor, the motor also started rotating. So, by looking even into the result 2, we can understand at what RPM that motor is rotating at, but there is an error, suppose if I increase the RPM, the P-gain, the value is increasing, but say, I will make it, I'll stop the value and I will go with the maximum RPM, which is of in our case say 100. So, if we observe that the output, the manipulated variable as well as the output, because it's a gain is very higher, initially the values of smaller, the set gain was higher, the RPM is of 100 RPM. So, initially it will should start from 0. So, error is higher, as a result it is you know the P-controller output is a more than 5. So, since we are using a regulatory propter which is a 5 volts, which can only maximum yellow is 5 volts, it is making the motor to on, off, on, off, on, off continuously, so that means here we cannot understand anything, this is completely because of the gain that we have set is really higher. So, one way to avoid this problem which is you know unstable situation is decreasing the gain. So, if the gain if 0.09 it is really difficult to work with that so let me change it to zero, let me clear the chart and changing the gain to 0.05 and again I will change the RPM to 100. Now, even in this case, since the gain is of really high, 0.05 we can see the fluctuation. So, even the gain 0.05 is higher so, again I will change the value somewhere around 0.03, let me clear the chart, then change it to 100 RPM. Now, see this is bit bit, Okay. Even when you are operating with a higher RPM, Right? which is a maximum

RPM, in our case so 0.03 gain is not making the DC motor to go to the saturation or unstable region, its but the problem is that since the gain value is very smaller the error if you see this is nothing but complete offset error. So, we require 100, right now it is at 40 RPM, 39.6, that offset error we cannot eliminate with simple P-controller. So, one way to eliminate is, you have to increase the gain, but increasing the gain, it goes to unstable region, but during the operation it is, Okay. But, initially when when you are going from 0 to 7 related to that value this will automatically create the system to go to the unstable reason, Okay. Let me try with going with 0.04. So, I will make I made it to 0. so I will make it as 0.04 and make the RPM to 100. Even, even this particular value, if you closely observe actually speaking the motor is started fluctuating, but here we cannot understand the motor is fluctuating so what I will be doing is that I will change it to again zero, 0.035, let me change it to 100, yes. Now, initially it may have some fluctuation, but later on it is pumping a constant value to the decimator. So, this is Okay. 0.035 gain P-gain, but but initially if the initially there are chances no motor will be has to go under saturation design, Suppose if the gain is of 0.04, this will be 0, say 100, we can see the motor is stopped again running, stopped and running, this is due to un-stability because of the gain is really higher in order to go to stable designer, it has been stabilized, in order to go to stable reason the motor is going to unstable, stable, unstable, stable but it creates a problem. So, the gain. So, one way to identify this problem is lowering the gain. So, you will set to 0.03 itself. So, again I am putting it to 100, we can see that the manipulator is smaller, but the problem in this case is that the problem will be error. So, when you look into our screen here, we can see, the set value is 100, the RPM is of only 39, because the P-controller. Since, it is a secondary system the simple P-controller cannot eliminate this offset error at all, this offset error throughout the operation will remain same, So 40 and 100, 60 RPM, suppose if I change it to somewhere around say 80 RPM, even then we will be experiencing the same amount of RPM difference, approximately, Right? So, this set is 80, the actual is 30.5, 30. So, since it is second order system, the problem is that the simple P-controller cannot solve it, so how do we avoid this problem, one way to avoid this problem is, to go with I-controller, but when we go with a I-controller, if you recall the working of I-controller where it looks in to the previous error values and accounts added with the present error and gives the manipulated variable, but the problem is that if the I-gain is really higher there are chances of getting an overshoot, that overshoot should not damage our circuit, but even though we have taken complete they have created logic in even in in in our labview programming, as well as in our you know? hardware where we have replaced the output of data acquisition device through a Zener diode circuit, Zener diode voltage of 5 volts, even if the manipulated variable is higher then the voltage which is given provided to the DC motor will be only 5 volts. So, in order to understand, let me stop this, what I will be doing is that I'll change I'll be setting it to I-gain, some I-gain value. So, I will set the I-gain to in this case I am going withy 0.1, Okay? Let me run, so initially the RPM is 0. So, even if you see that actual RPM is also 0. So, what I'll be doing is that I will change it to somewhere around 100 RPM, Right? So, here we can see, its started slowly, started increasing, increasing, increasing, these are set point and you can also observe by looking into the manipulated variable and the time it is taking is approximately of 2 seconds, 1 second, Right? Still not at least 95, 100, slowly increasing, now if you observe 97.4, Right? So, with the P-controller, we couldn't achieve, we cannot reach to the actual set point because of our secondary resistance, but whereas with the PI controller, depends up on the gain that I am setting it. So, it is go, it is reaching to our required set point, but what we have to see here is, what is the time taking to reach to, to reach to the set point is here if you see actual RPM is also 100, set point is also 100, which means that it is reached to the value, but the value it is taking somewhere around 2 seconds in order to reach the set point, Now, is it, Okay? To have 2 seconds, if not can we modify our gains in order to reach even little faster, yes. So, now let's see let me also show you another decreasing it. So, suddenly I change it to 10 RPM, but here when you look in to the system, so manipulated variable is 4, means it is keep on fluctuating, keep on fluctuating, this is because the gains are really higher, now it has stabilized, so here we can observe it's output suddenly keep on increasing, decreasing, increasing, decreasing, increasing, decreasing again slowly started decreasing it. So, still the gains whatever we have considered is a little higher incase of when we are suddenly going from 100 RPM to 0 RPM, 0 RPM to 100 RPM. So, we have to tune our P-controller gain as well as I-controller gain. So, still it takes 2 seconds in order to reach to 10 RPM, but our intention was to have everything smooth, smoothly increasing, smoothly decreasing without any you know? stage of un stable reason, so in order to do that what we have to what I will be doing is that I'll stop, I'll making it as 0 RPM, I will stop this, I will decrease our, I will increase our I-gain and I will decrease our P-gain to somewhere around 0.02, Okay? This I am changing it 2.1 and this value I make it as 0.2, this I will make it as 0.01, let me clear that chart that operation, clear chart, I'll run, right now it is at 0, so that's why the manipulating variable is also 0, then, I'll change it to somewhere around 100 RPM. So, here we can see the RPM is slowly increasing, but one thing it is clear that we do not experience any jerk, its smoothly, smoothly keep on increasing whereas, in previous case it was not like this, So, right now simulation time is 0.5, in order to properly match it, what we can also do is that we can change the simulation settings to smaller value updation rate and everything. So, it will be the value whatever we can see will be very close, But still the values are not enough, so what I'll do is that, I will change the simulation settings once again and I will change the gain, P-gain as well as the I-gain and we will see that, change the value . So, let me change it to 0. So, starting with 0 the expect to 0.02, Okay. Let me run. So, now see that it is already 0, I will change it to 100 RPM. So, this point I have change it to 100 RPM, started slowly increasing, increasing. So, this tells that with respect to real time within 1 second, not even 1 second within 0.5 second, you can reach to the desired set point, 100 RPM, Right? We have already reach to 100 RPM, so which is checking somewhere around 1, 1.5 seconds, suppose if I make it to 0 again, and it is maintaining, the idea is that the whole idea is having a controller is to control, maintain the RPM that user or set, user wants to do that. So, when you see the applications of this, you can see, you can understand the quiz control system, where the you know user, the driver which are the speed that the car has to go, he can set the value and he can add on say, without having leg on the pedal, the car can drive at the particular RPM. So, now I am changing it to 0, started slowly decreasing and still even for decreasing also it takes same 2 seconds in order to come to halt position, here we can see, so what if I change my gain little higher, then we can see very fast output. So, what I will do is that right now, I will change it to 0.3, this to 0.22, let me stop once again then run the program. So, since the integral will always, will calculate for the previous error. So, we have to understand whether these gains can be use for that. So, run again once again. So, I will change it to 100 RPM, since I-gain is higher, now we can say that with respect to the real time, if the same thing has been implemented with the real time, If we see that with this gain parameters within 1 second, not even 1 second, within 0.5 second, it ca, it could be still the required set point, so the maximum RPM 1.5, but the problem is that, when we observe, when suddenly drastically decreasing from 105 RPM to 0, the gain that we have set, initially it is creating a problem at here, let me zoom it, So, in this case here you can observe there is a sudden decrease and increase, now suppose if I have go again 100, Right? started somewhere around 2.25, with the same parameters actually speaking it will take only 2.8, so, 2.3 to 2.8, 0.5 seconds approximately, does it reaches the set point, but initial jerks we can observe, so when I am making it to 0. So, with this we can understand, how exactly the working of the PI controller, how to understand the data acquisition device, how to interface the real time signals to the PC, and how can, how let the, how to monitor and control a DC motor. So, with these I'll stop this module

Video End Time: (20:27)


Main VI_DC MOTOR.vi
File Edit View Project Operate Tools Window Help
Waveform Chart
Setpoint
Actual
Amplitude
Time
Result 2
Voltage (FFT - (Peak))
Frequency
Set Point (RPM)
P_Gain
I_Gain
D_Gain 2
MV
0.604337
Actual RPM
6.88718
Stop Button
Stop
20:27 / 20:29