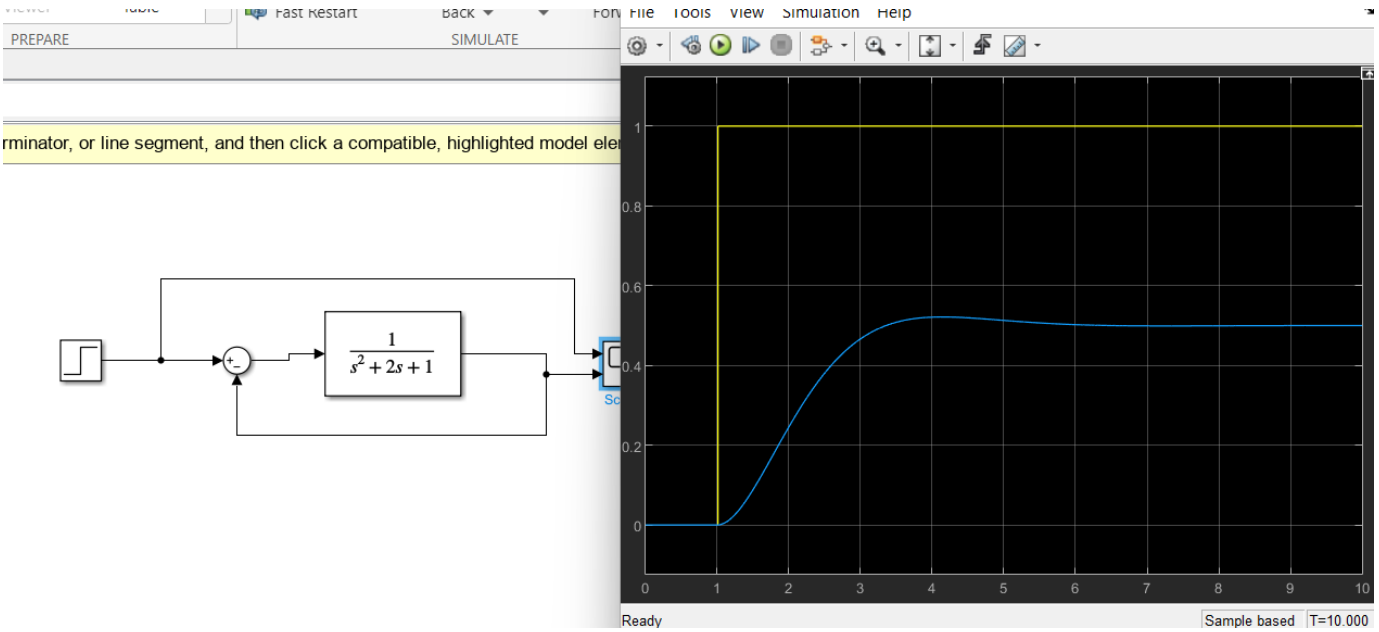
Mazen HossamEldeen MBD_diploma

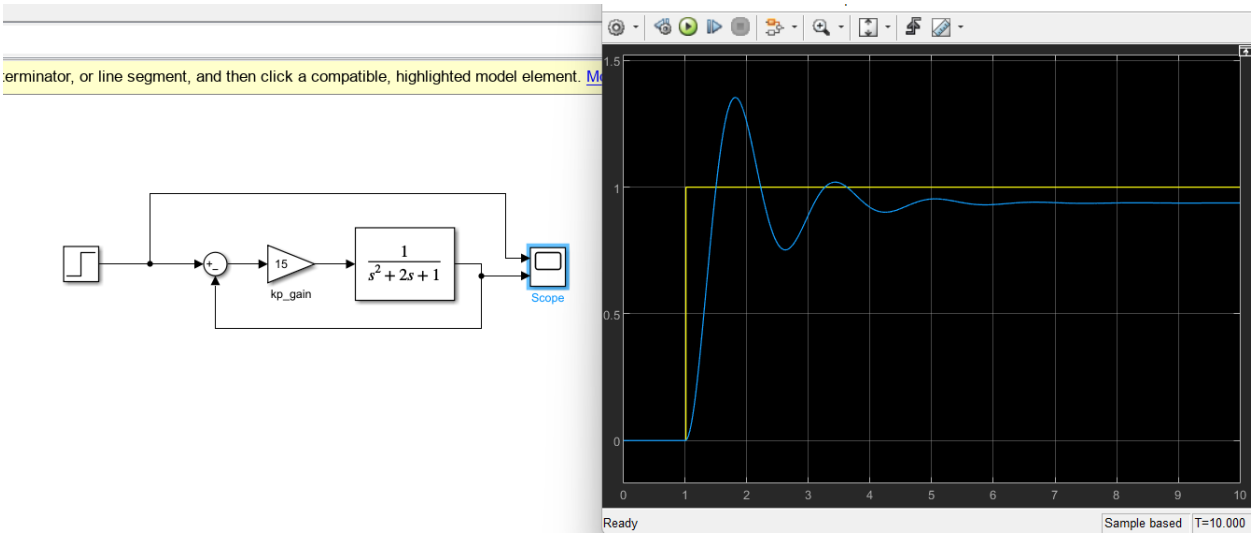
Lab: Practical Demonstration of P, I, and PI Controllers

Normal second order system without any controller

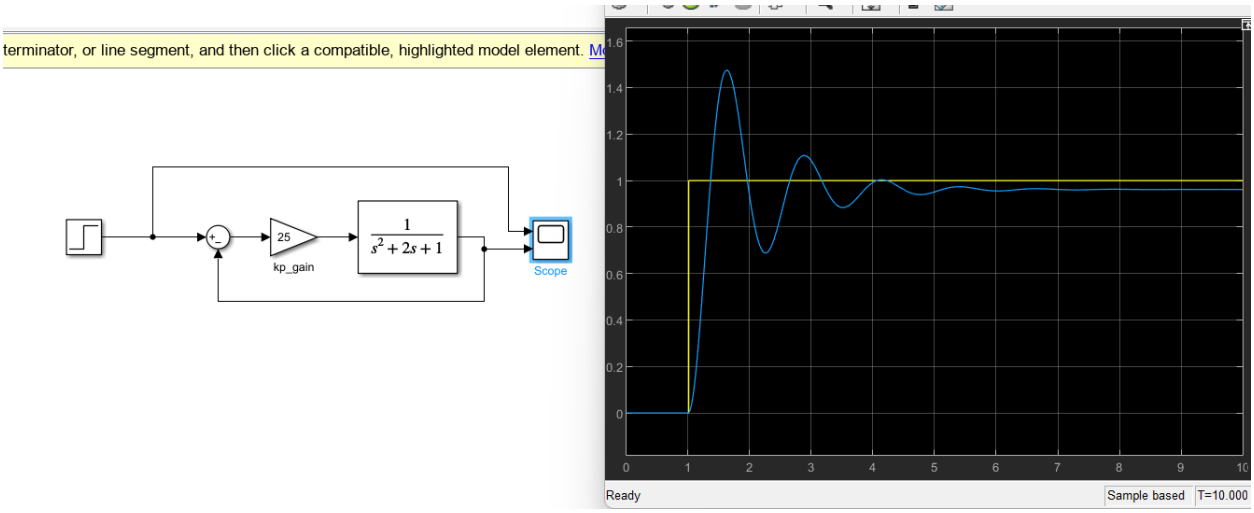


->There is a small overshoot below 1 and steady state error of 0..5

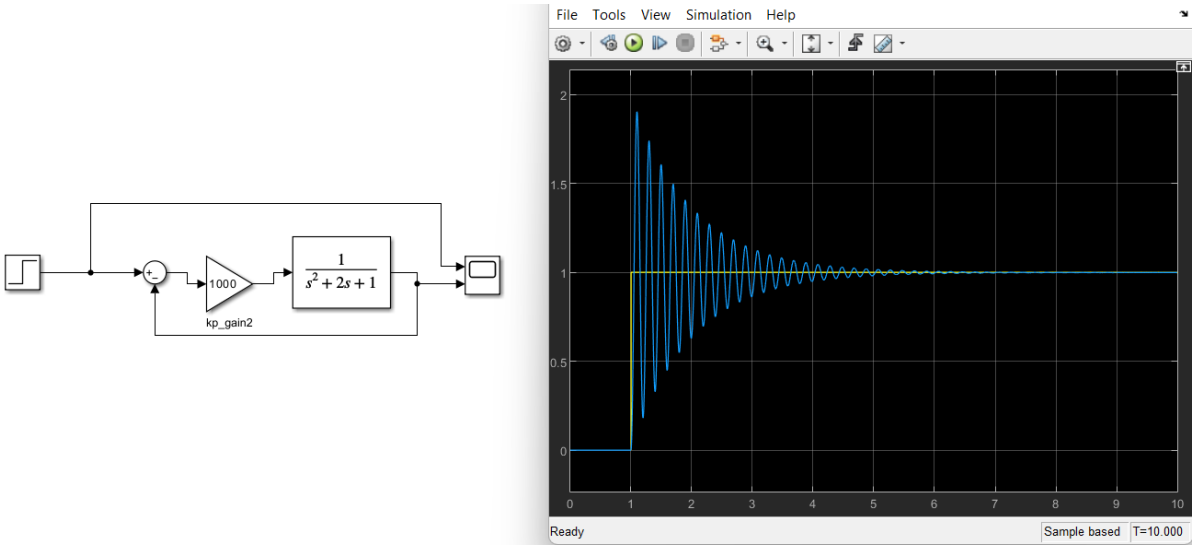
P_Controller



->After we added proportional controller (kp_gain) with a value of 15 oscillations and overshoot has occurred also the overshoot was big and passed 1 but the steady state error decreased from 0.5 to 0.2 .



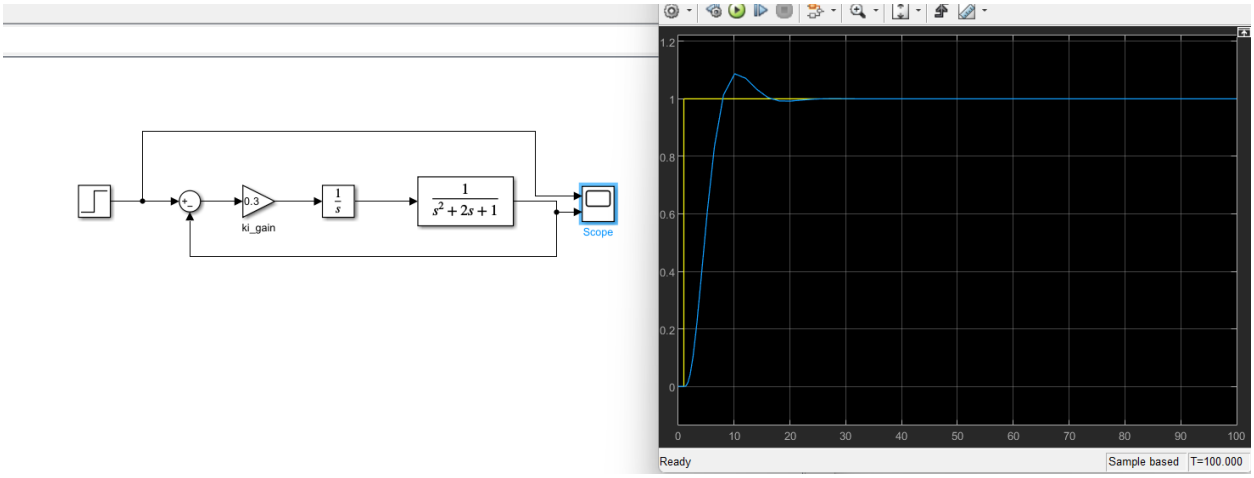
->When you increase the value of kp such as 25 in this example the steady state error decreases which in this case its very close to 1 but in return oscillations increased and overshoot increased but kp has limitations.



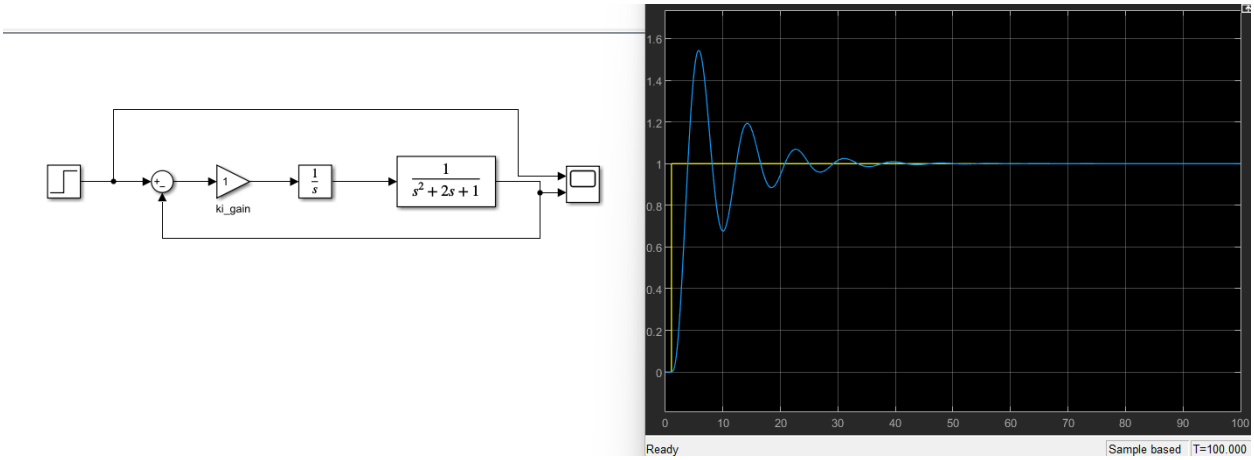
->In this case the value of kp is 1000 which as we can see in the scope number of oscillations and overshoot has increased although the steady state error decreased a lot and its very close to 1 yet the oscillations and overshoot can affect the system in a bad way .

So by conclusion increasing the value decreased steady state error and focuses on the present error but increasing value kp soo much can lead to increase in oscillations and overshoot that's why it is always preferred to use tunning not try and error method to choose the best value that suits kp.

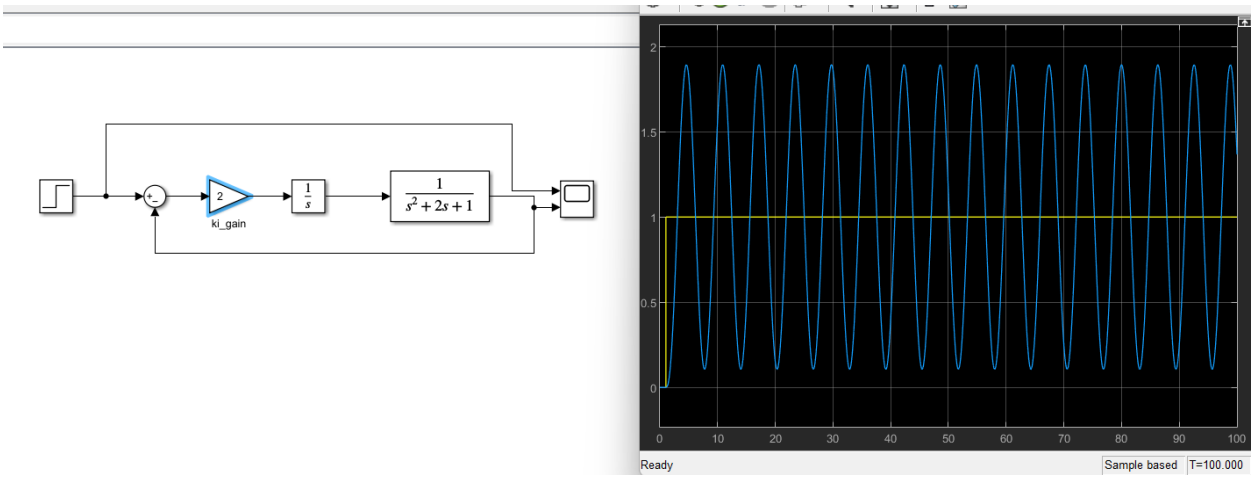
I_Controller



->When I added I controller with a value of 0.3 small overshoot happened but it has eliminated the steady state error.



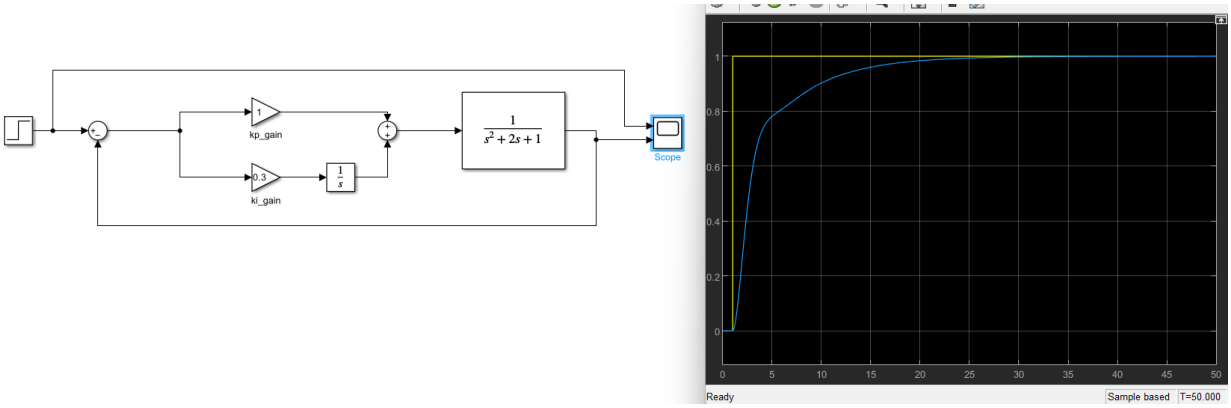
->As you can see here in this case I have increased the value of the ki_gain to 1 which eliminated the steady state error but also increased the oscillations and the overshoot.



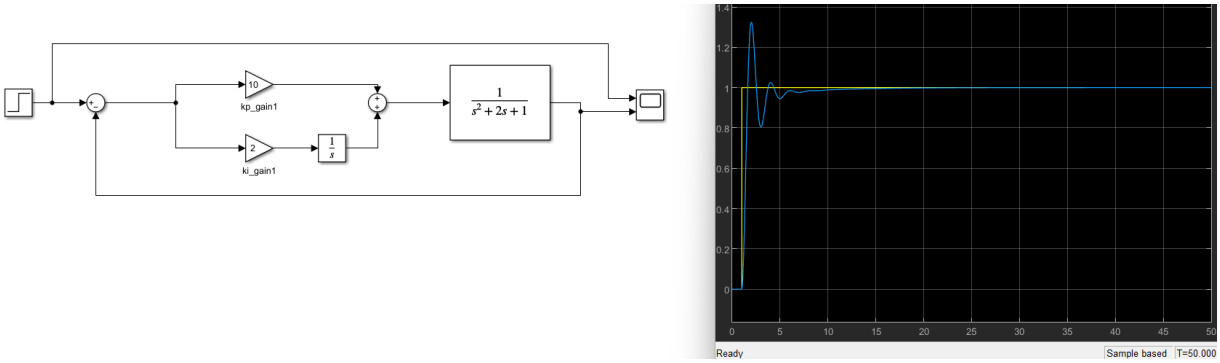
->In this case this is prove to us that the value of the ki has also limitations although it eliminated the steady state error yet increasing it without tunning or knowing the right value that suits ki causes oscillations which can affect the system in bad way.

->So in conclusion the value of ki has limitations increasing it without tunning and using try and error method can cause the system to oscillate which will affect the system in bad way .Knowing the right methods to choose the best value of ki will help you eliminate the steady state error by focusing on the past error

PI_Controller



->After we added PI controller with these values kp=1 and ki = 0.3 the system was able to reach a steady state error = 0 and there no overshoot nor oscillations.



->Increasing the value of kp and ki had an strong effect on the system when kp = 10 and ki = 2 the system had a fast response yet still and oscillation happened with huge overshoot but eliminated the steady state error that's why its preferred tunning methods to choose the best value that suits both kp and ki.

Also here were comes the pid using d controller with pi helps us to remove the overshoot since d controller focuses on the future error.