EE208 EXPERIMENT 9

SIMULATION AND CONTROL DESIGN ON SIMULINK

Group no. 9
Aryan Bansal – 2020EEB1162
Hiya Kwatra – 2020EEB1173
Ishaan Sethi – 2020EEB1174

OBJECTIVE

To analyse the effectiveness of Ziegler-Nichols rules in presence of poles with high multiplicity.

SYSTEM

The basic OLTF unit provided is to be considered at different multiplicities (as an example, multiplicity order of one to ten) –

$$G(s) = \frac{1}{(s+1.5)^k};\ k = 1 \dots 10$$

TASKS

SETUP

1. For each multiplicity, create a block diagram on Simulink, and conduct the appropriate Ziegler-Nichols test (OL for lag systems, CL for oscillatory systems). Hence obtain the tuned constants for P, PI, and PID control by the Ziegler Nichols rules.
2. Cascade the respective controllers to the OLTF systems created in Simulink.

OBSERVATIONS AND DISCUSSIONS

1. With the unit step as the change in CLTF reference, obtain the dynamic response on screen.
2. In terms of different dynamic performance measures (say rise time, peak overshoot, settling time, etc.) analyse the effectiveness of the "ZN rules" for multiple poles systems.
3. Discuss the merits or demerits exhaustively in terms of the performance measures that are chosen to be tabulated and reported.

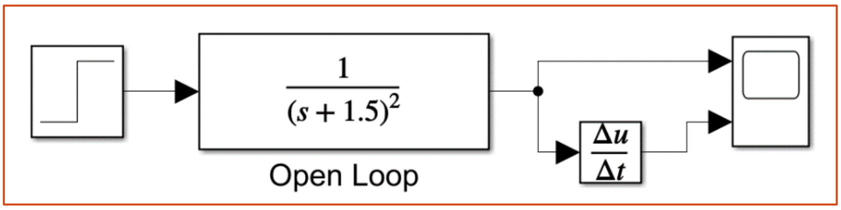
SETUP

We shall discuss tuning for $k = 1$ to $k = 15$.

SIMULINK BLOCK DIAGRAMS

We use the Zero-Pole block in Simulink to create the transfer function. The derivative block is added to find the constants required for Zeigler-Nichols rules.

Since the block diagrams will be similar for all multiplicities, we are only presenting one of them –



ZIEGLER-NICHOLS TEST[1]

Since all the systems are Lag-type (only real poles, no complex conjugate or integrators), we shall use the first Z-N method.

From the Open-Loop step response, we shall find:

- R = slope of the tangent line drawn at the curve's inflection point, and
- L = time delay/lag

We shall graphically find the maximum derivative of the step response (i.e., the first inflection point) with the Cursor and Peak Finder tools in the Scope block.

L is the time at which the maximum-slope tangent cuts the time-axis.

Thus, it can be found with this formula – $\boldsymbol{L = T_{max} - (Y_0 / R)}$.

Here, T_{max} is the time at which slope is maximum, and Y_0 is the value of the step response at this time.

This method results in a closed-loop step response transient with a decay ratio of approximately 0.25, meaning that the transient decays to a quarter of its value after one period of oscillation.

[1] 4.3.6, Feedback Control of Dynamic Systems (FPE), 8th Ed.

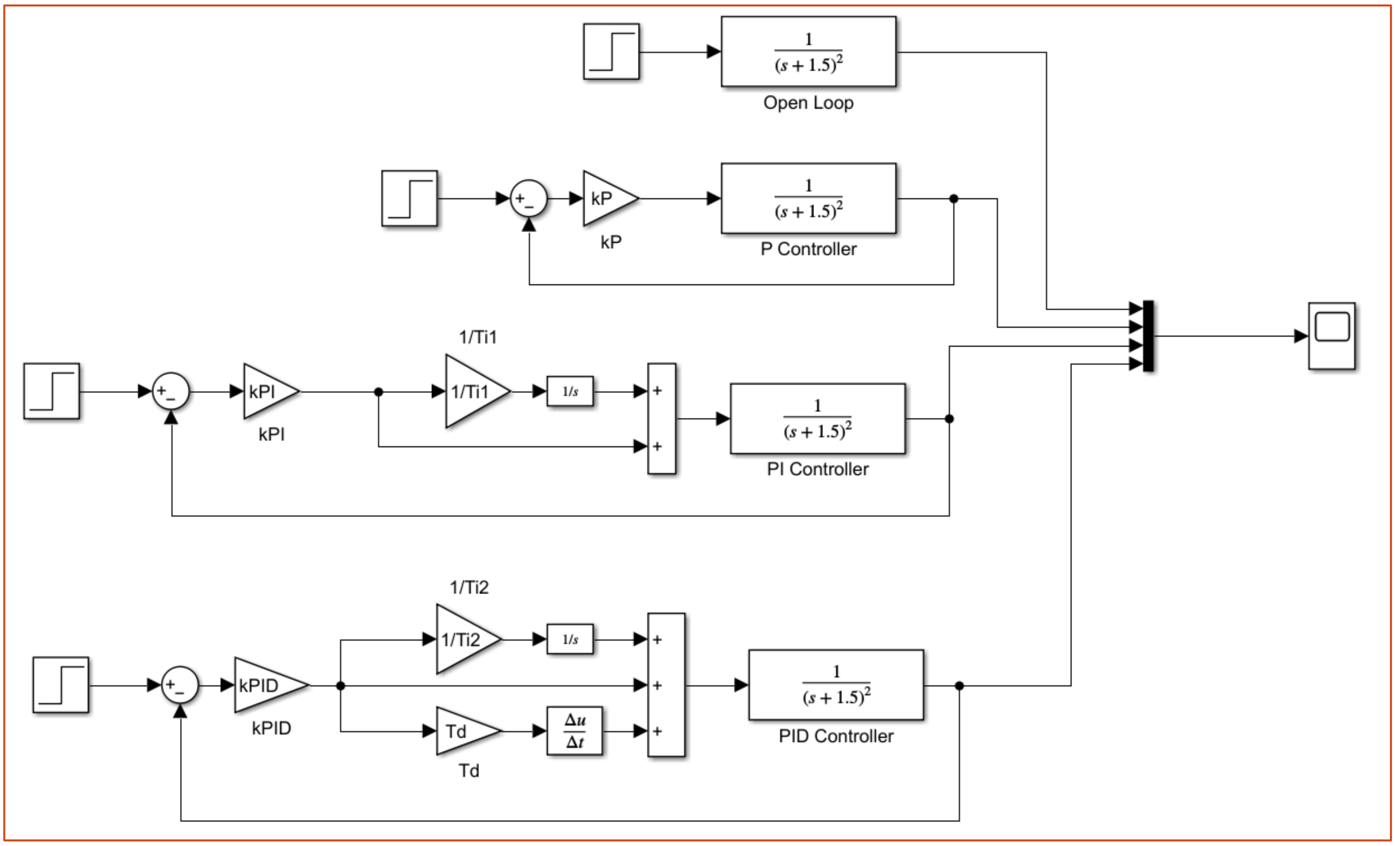
TUNED CONSTANTS[2]

Having found R and L, we shall use the following formulae to find PID coefficients for a controller of the form $\boldsymbol{k_P\left(1+\frac{1}{T_i s}+T_d s\right)}$ –

Controller	k_P	T_i	T_d
P	$\frac{1}{RL}$	∞	0
PI	$\frac{0.9}{RL}$	$\frac{L}{0.3}$	0
PID	$\frac{1.2}{RL}$	$2L$	$0.5L$

CASCADED CONTROLLERS

We now implement the controllers in Simulink –



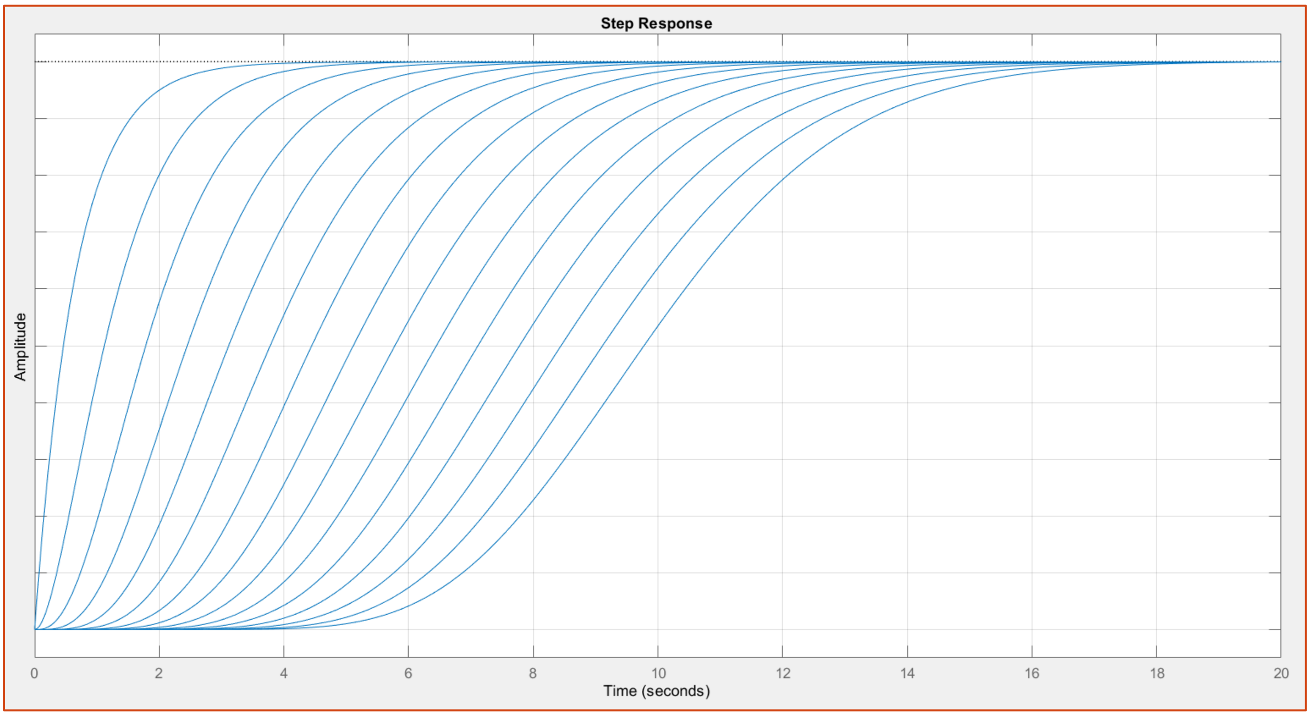
- PID Controller implemented is of the form $k_P\left(1+\frac{1}{T_i s}+T_d s\right)$
- Blocks used for controllers – Gain, Integrator, Derivative
- Simulation time step = 0.001s

[2] Table 4.2, Feedback Control of Dynamic Systems (FPE), 8th Ed.

OBSERVATIONS

Open-Loop Step Responses of all the cases

The normalized step responses are (k=1, to k=15) –

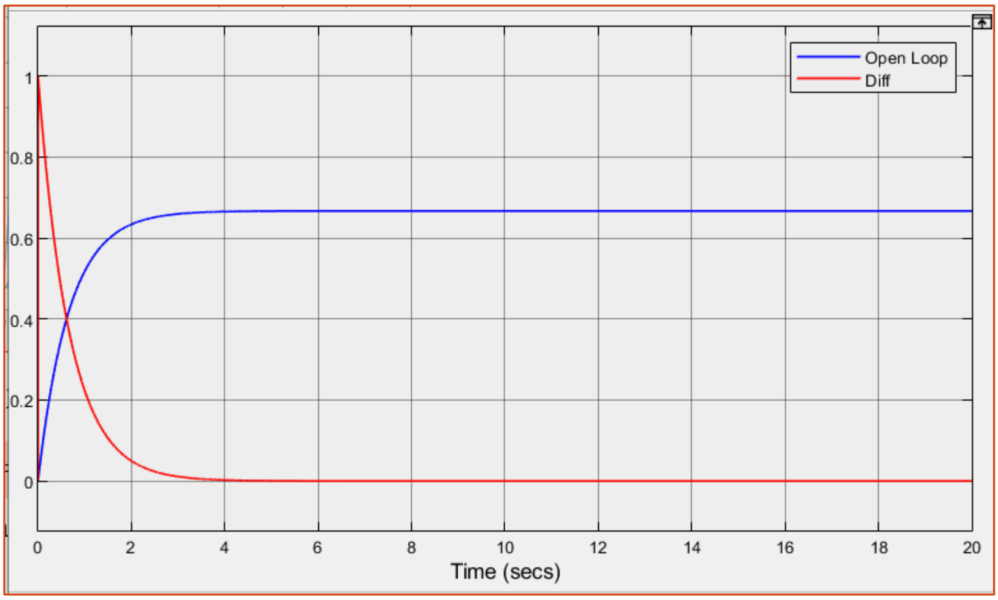


All the responses are lag type; hence we will use the first ZN method for tuning.

k = 1

OL STEP RESPONSE

The open-loop step response is –



ZN TUNING

There is clearly no "lag" in this system – i.e., **L = 0**.

The ZN rules cannot be used directly for this system, since they all involve a term of the form $1/L$.

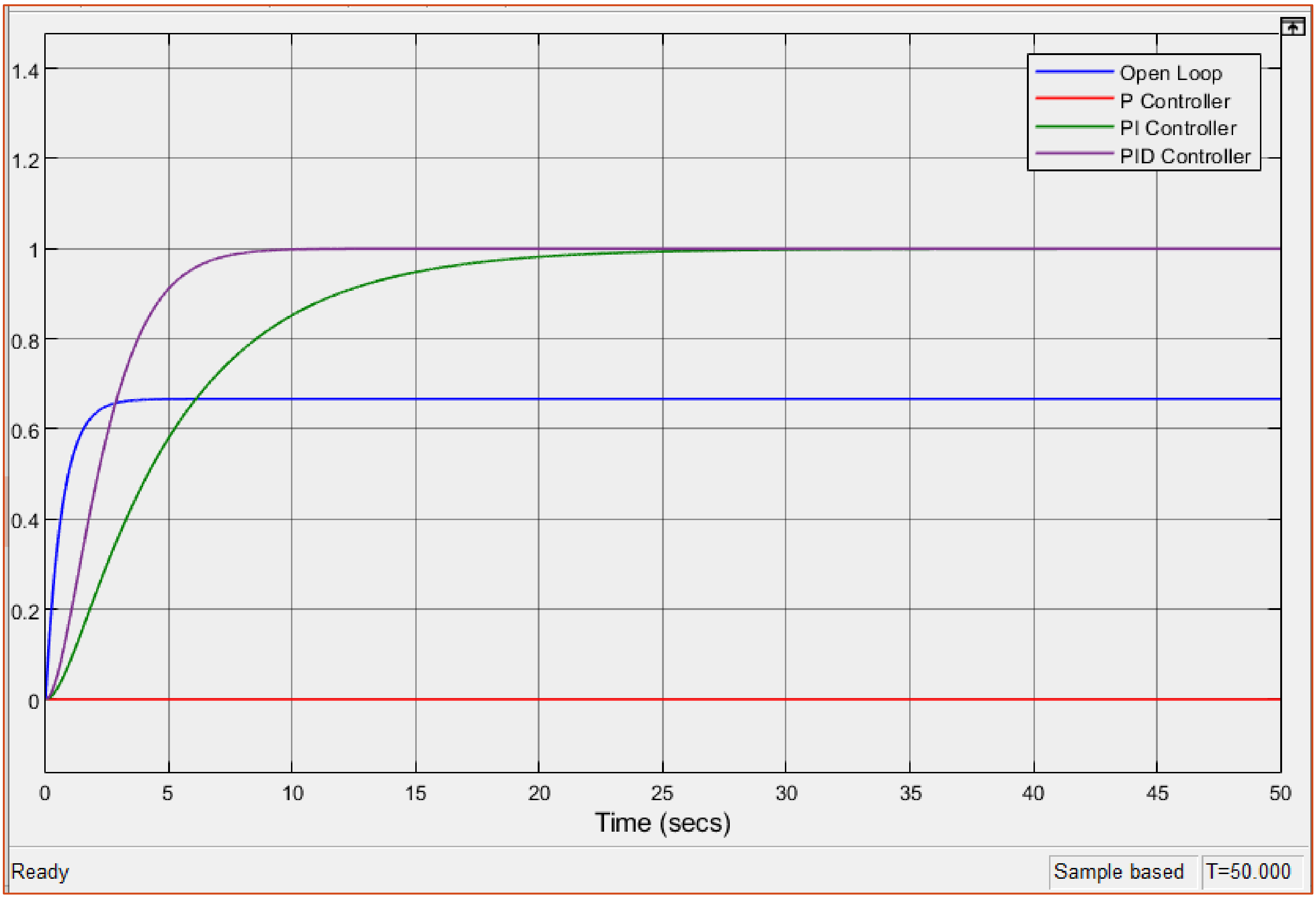
For the sake of experimentation, we may try some very small value of L, say 10^{-7} and observe the results.

With $R = 1$ and $L = 10^{-7}$, the coefficients are –

1. $k_P = 10^{-7}$
2. $k_{PI} = 9 * 10^{-8}, T_{i_1} = 3.333 * 10^{-7}$
3. $k_{PID} = 1.2 * 10^{-7}, T_{i_2} = 2 * 10^{-7}, T_d = 5 * 10^{-8}$

Note: These are very small values, so a different control technique is advisable.

CLOSED LOOP RESPONSES



STEP RESPONSE PERFORMANCE MEASURES

We obtain the performance characteristics as –

Performance Measure	**Open Loop**	**P Control**	**PI Control**	**PID Control**
Rise Time	1.4648 s	1.4648 s	10.7423 s	4.1343 s
Settling Time	2.6080 s	2.6080 s	19.5400 s	7.0628 s
% Overshoot	N/A	N/A	N/A	5.1974e-04 %
Steady-state Value	0.6667	6.6667e-08	1.0000	1.0000

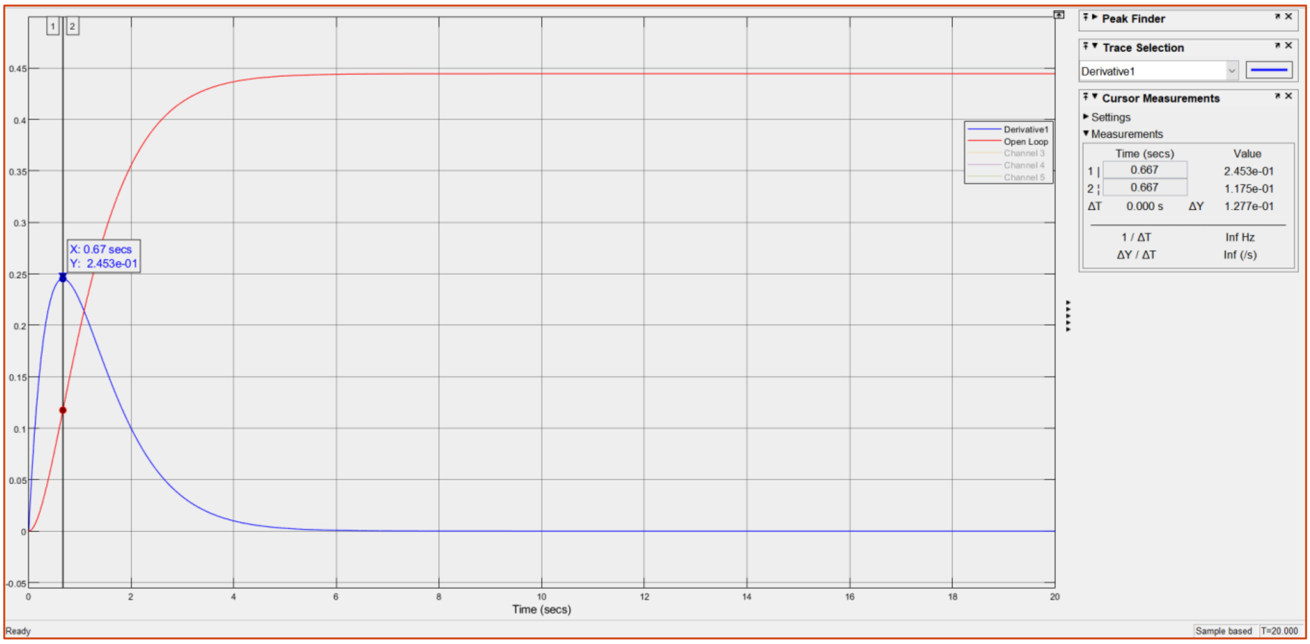
We observe that –

- P control scales the entire response down to a very small value, practically 0 response.
- PI control greatly increases rise and settling times, but eliminates steady-state error.
- PID control reduces rise and settling times wrt PI control, and the overshoot added is negligible for most applications.

k = 2

OL STEP RESPONSE

The response is lag type, there are only real poles.



ZN TUNING

From the plot –

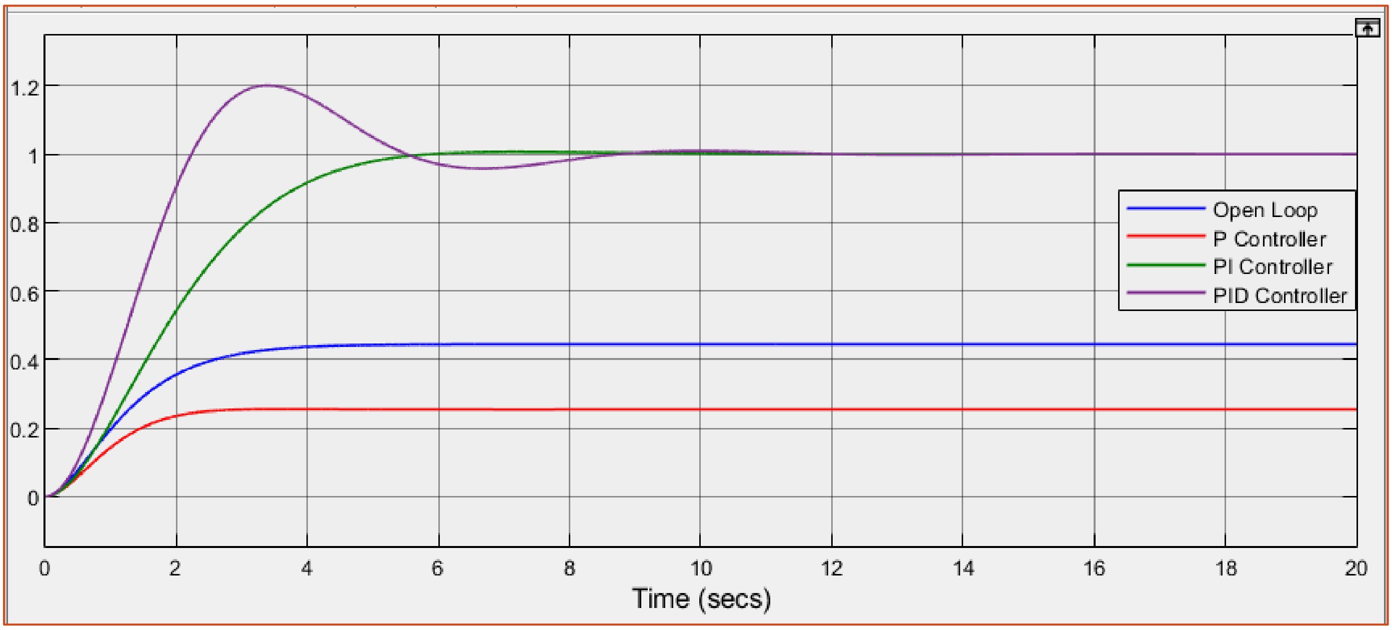
1. $R \; = \; 0.2453$
2. $T_{max} = 0.667$ sec, and $Y_0 = 0.1175$
3. $\Rightarrow L \; = \; 0.1880$

Using the table, we can find the ZN coefficients –

4. $k_P = 0.7664$
5. $k_{PI} = 0.6898, \; T_{i_1} = 0.6267$
6. $k_{PID} = 0.9197, \; T_{i_2} = 0.3760, \; T_d = 0.0940$

CLOSED LOOP RESPONSES

Thus, we obtain the Closed Loop responses –



STEP RESPONSE PERFORMANCE MEASURES

We obtain the step response performance measures using the *To Workspace* block and the *stepinfo* function.

Performance Measure	**Open Loop**	**P Control**	**PI Control**	**PID Control**
Rise Time	2.2386 s	1.5684 s	3.2040 s	1.4990 s
Settling Time	3.8893 s	2.4884 s	5.0368 s	7.8902 s
% Overshoot	N/A	0.4594%	0.6193%	20.0589%
Steady-state Value	0.444	0.2541	1.0000	1.0000

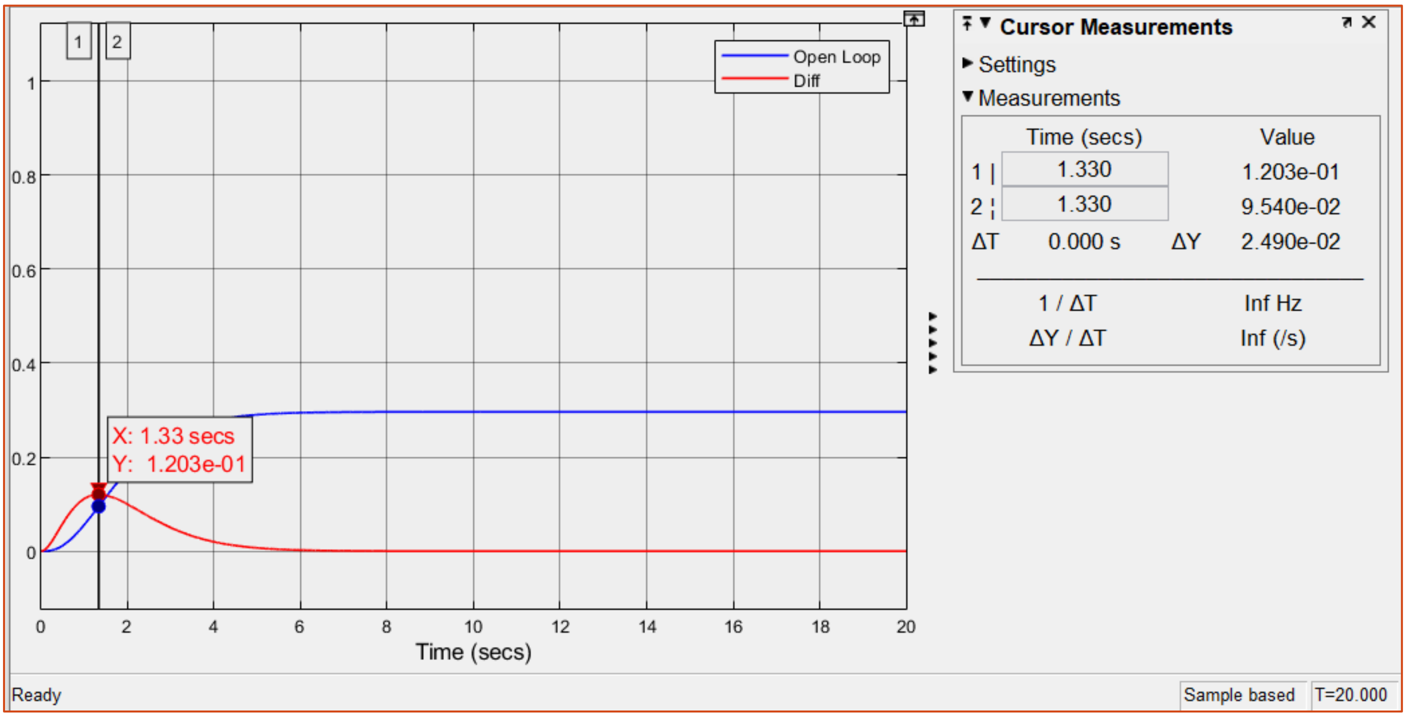
We observe –

- P Control reduces steady-state value and rise time.
- PI Control increases rise time, and 0 steady-error is achieved.
- PID Control reduces rise time wrt PI control but greatly increases the overshoot and increases settling time.

k = 3

OL STEP RESPONSE

The Open-Loop response is –



ZN TUNING

From the plot –

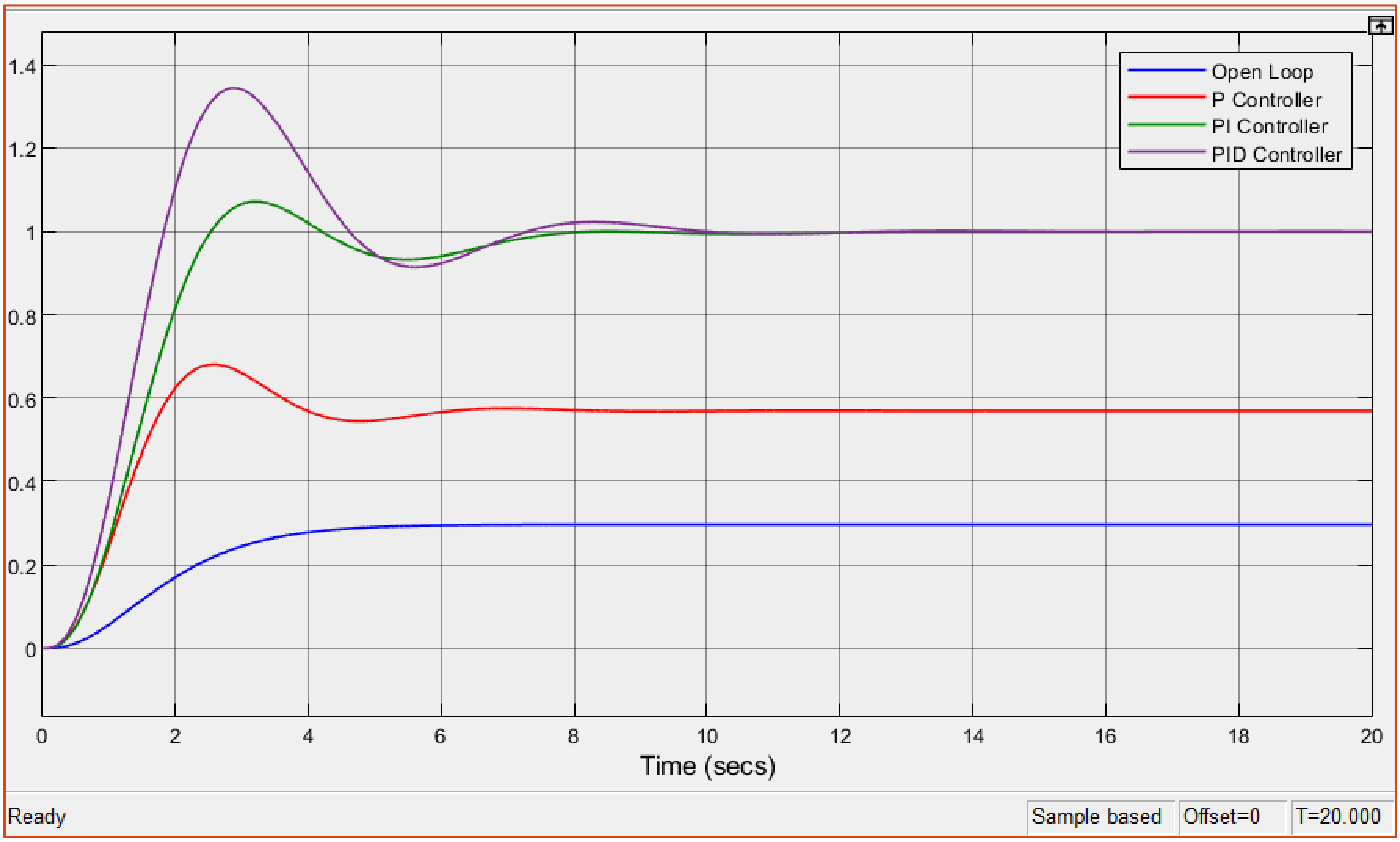
1. $R \;=\; 0.1203$
2. $T_{max} = 1.33$ sec, and $Y_0 = 0.0954$
3. $\Rightarrow L \;= 0.5370$

Using the table, we can find the ZN coefficients –

1. $k_P = \; 4.4637$
2. $k_{PI} = 4.0173, \; T_{i_1} = 1.7899$
3. $k_{PID} = 5.3564, \; T_{i_2} = 1.0740, \; T_d = 0.2685$

CLOSED LOOP RESPONSES

Thus, we obtain the Closed Loop responses –



STEP RESPONSE PERFORMANCE MEASURES

We obtain the step response performance measures using the *To Workspace* block and the *stepinfo* function.

Performance Measure	**Open Loop**	**P Control**	**PI Control**	**PID Control**
Rise Time	2.8135 s	1.0969 s	1.5481 s	1.1110 s
Settling Time	5.0111 s	5.6117 s	7.1163 s	8.7083 s
% Overshoot	N/A	19.3657 %	7.1578 %	34.4309%
Steady-state Value	0.2963	0.5694	1.0000	1.0000

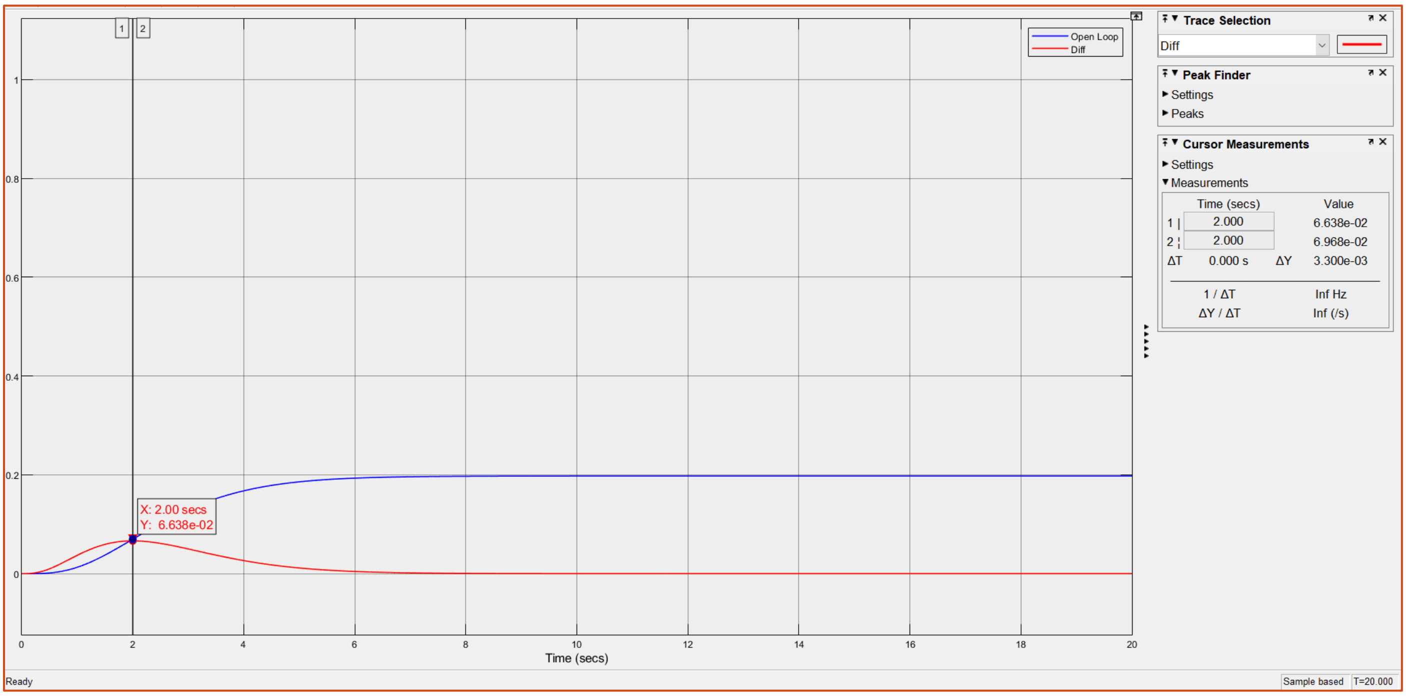
We observe –

- P Control reduces rise time and adds an overshoot of approx. 20%.
- PI Control increases rise time wrt P control (lesser than OL however), and 0 steady-error is achieved. Overshoot is also reduced to around 7%. However, settling time has been increased by ~2 sec.
- PID Control reduces rise time wrt PI control but greatly increases the overshoot and settling time. The overshoot is ~34.4%.

k = 4

OL STEP RESPONSE

The Open-Loop response is –



ZN TUNING

From the plot –

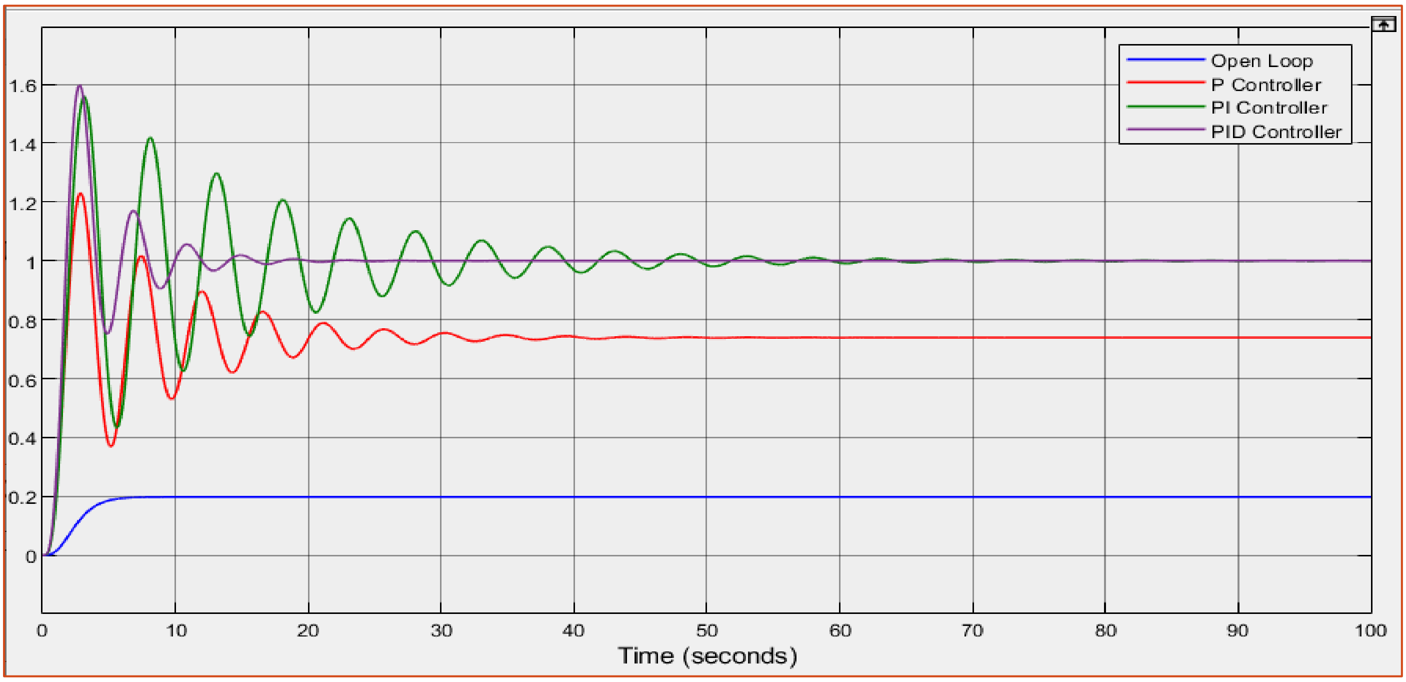
1. $R \ = \ 0.06638$
2. $T_{max} = 2$ sec, and $Y_0 = 0.06968$
3. $\Rightarrow L \ = 0.9503$

Using the table, we can find the ZN coefficients –

1. $k_P = \ 14.3159$
2. $k_{PI} = 12.8843, \ T_{i_1} = 3.1676$
3. $k_{PID} = 17.1790, \ T_{i_2} = 1.9006, \ T_d = 0.4751$

CLOSED LOOP RESPONSES

Thus, we obtain the Closed Loop responses –



Even from a glance it is clear that the ZN tuning rules do not work well for this system, there are excessive and slowly decaying oscillations, which are undesirable.

STEP RESPONSE PERFORMANCE MEASURES

We obtain the step response performance measures using the *To Workspace* block and the *stepinfo* function.

Performance Measure	**Open Loop**	**P Control**	**PI Control**	**PID Control**
Rise Time	3.2907 s	0.9459	1.0872	0.9393
Settling Time	6.0561 s	30.5708	48.4594	13.4808
% Overshoot	N/A	66.5343	56.0115	59.6598
Steady-state Value	0.1975	0.7388	1	1

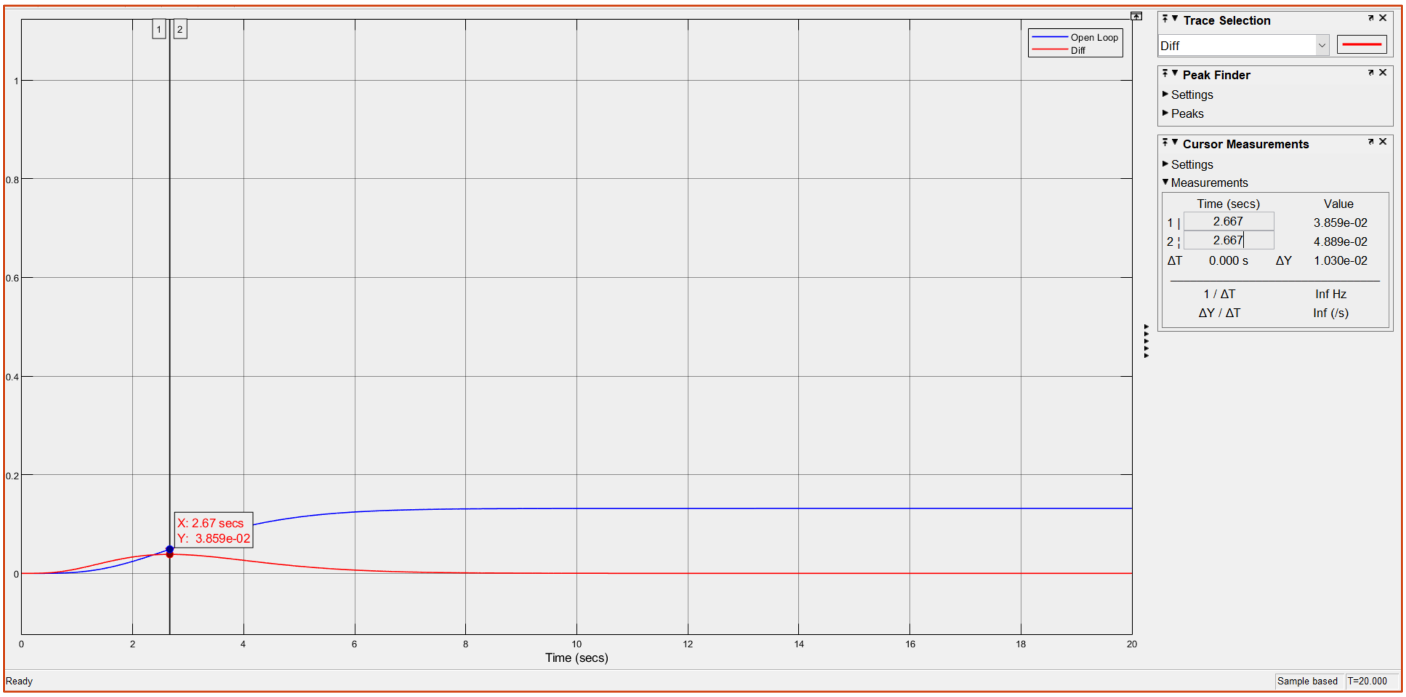
We observe –

- All three controls – P, PI, PID – are undesirable, due to the large settling times and overshoots.
- If large overshoots are not an issue, PID control is the most desirable – it rises and decays the quickest.

k = 5 and greater

OL STEP RESPONSE

The Open-Loop response is –



ZN TUNING

From the plot –

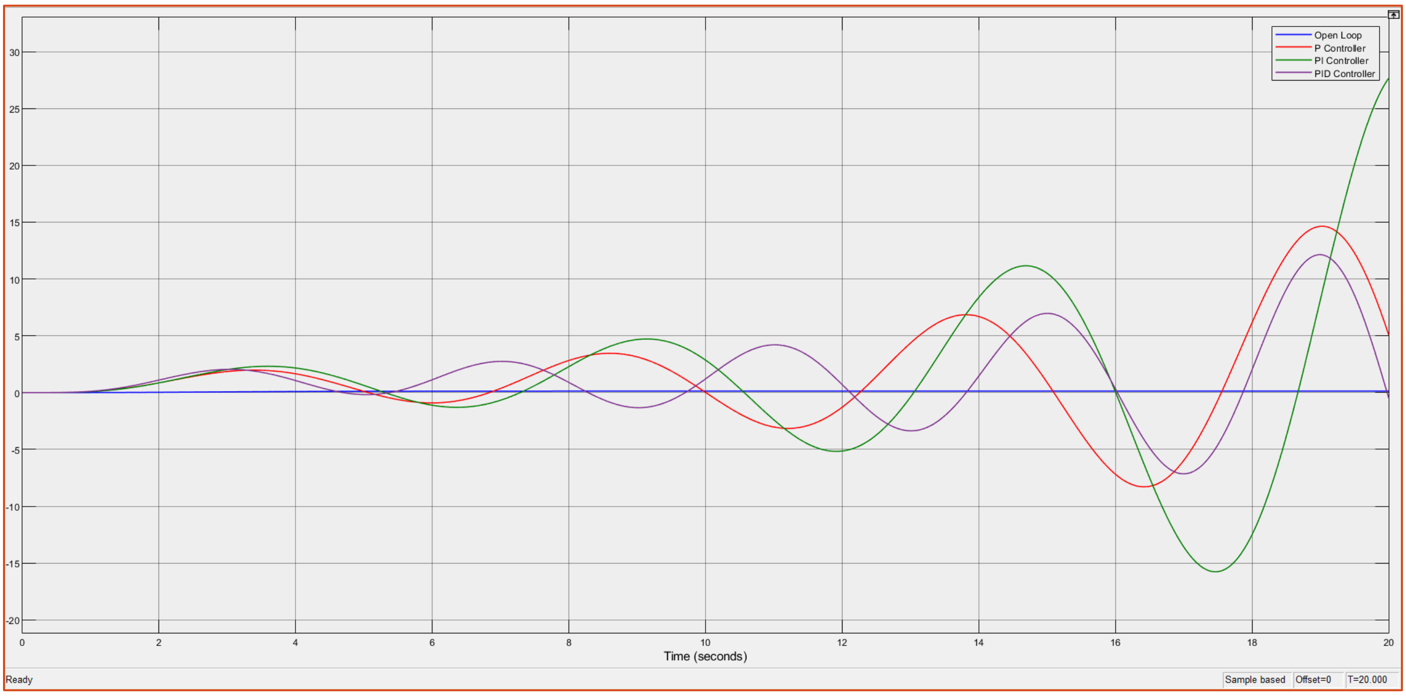
1. $R \ = \ 0.03859$
2. $T_{max} = 2.667$ sec, and $Y_0 = 0.0489$
3. $\Rightarrow L \ = 1.3998$

Using the table, we can find the ZN coefficients –

1. $k_P = \ 36.2745$
2. $k_{PI} = 32.6470, \ T_{i_1} = 4.6661$
3. $k_{PID} = 43.5294, \ T_{i_2} = 2.7997, \ T_d = 0.6999$

CLOSED LOOP RESPONSES

Thus, we obtain the Closed Loop responses –



Now, the ZN rules fail completely – the CL system is unstable! The gains from the Zn method have caused the CLTF to have unstable poles/zeros.

This will remain true for all the systems beyond this one, hence we will not show the observations for each of them.

DISCUSSIONS

EFFECTIVENESS OF Z-N RULES FOR K = 1, 2, 3, 4

k = 1

- Here we have arbitrarily assumed L to be 10^{-7} for the sake of analysis.
- For this case, ZN tuning rules apply well only for PID control.
- The trade-off wrt the Open-Loop case is between rise/settling time (quickness of response).
- Depending on the application, other tuning methods (or controller designs) are recommended.

k = 2

- P control is useful only for obtaining a faster response, i.e., reduced Rise Time.
- Due to the large overshoot and settling time, PID control may not be desirable in many situations even though it reduces rise time relative to PI control and Open-Loop response.
- It may be possible to further tune the PID coefficients to reduce the overshoot.
- If a slower response is acceptable, PI Control may be more suitable absence of large overshoot.
- ZN rules work well for PI control in this case, depending on the exact application.

k = 3

- If only disturbance rejection is wanted, P control based on the ZN rules is recommended over direct PID control as calculated by the ZN rules.
- Similar to the previous case, the choice between PI and PID control is about overshoot v/s rise time, if large overshoot is acceptable.
- The large overshoot is not acceptable for a large number of applications. As such, reducing the effect of the derivative term is recommended, with the trade-off between rise time and overshoot to be decided based on the exact problem at hand.

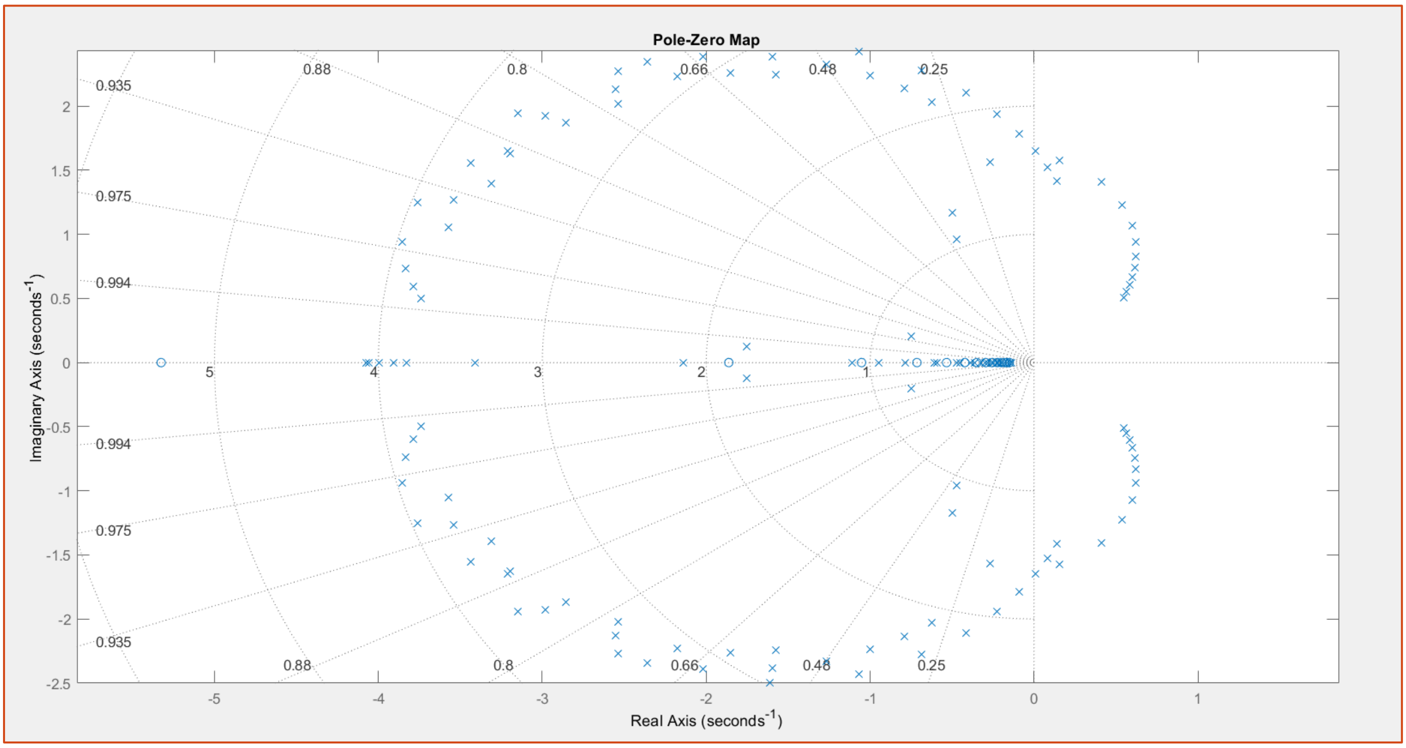
k = 4

- It is at this stage that the limitations of the ZN method can be clearly seen.
- The ZN rules simply do not work well for this case – they are optimized to provide a quarter decay, and not for less overshoot.
- Other methods such as Kappa-Tau tuning or Model Based Tuning is recommended.

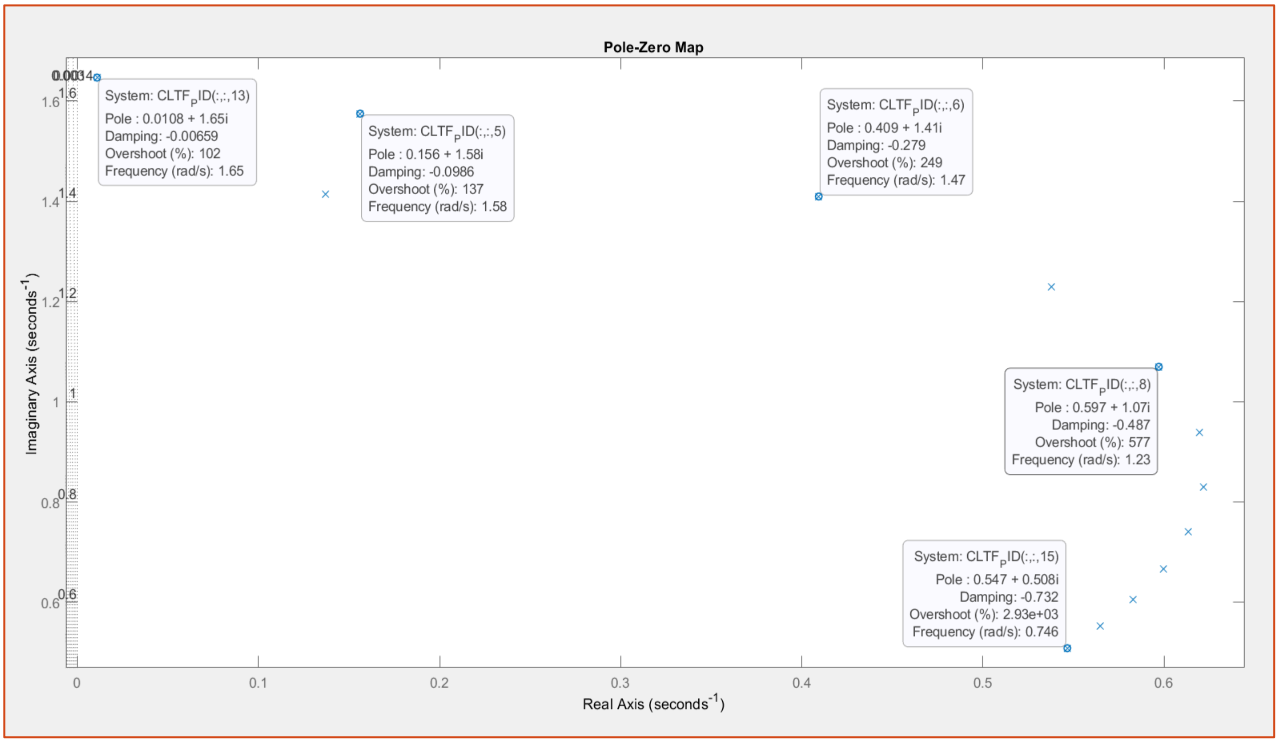
FAILURE OF ZN RULES FOR K = 5, 6, 7, …

The ZN rules fail for $k \geq 5$, resulting in unstable systems. This occurs due to the PID controller causing some CLTF poles to lie in the RHP.

We can check this by viewing the pole-zero maps for the PID Control CLTFs (the P and PI control maps are somewhat similar):



Zooming onto the LHP, we see –



The ZN rules are heuristic, and work well for a certain type of system - a one-pole transfer function with an ideal time delay[3]:

$$\frac{K \cdot e^{-sL}}{Ts + 1}$$

This transfer function roughly models a system with a response that is dominated by a single-pole exponential lag, and adds a linear phase shift from the time delay to model the phase effects of higher order poles.

For many systems, this works well. However, as we have seen, for multiplicities higher than 4, the gains as per ZN rules move some of the CL poles into the RHP leading to instability.

ZN rules are thus not suitable for poles of high multiplicity – other control methods based on root loci, frequency domain or advanced PID tuning methods should be used.

CONCLUSIONS

In this experiment we used Simulink to simulate a PID controller based on the Zeigler Nichols Tuning Method for given systems of increasing pole multiplicity. Our findings can be summarized as follows –

1. The ZN tuning method works best for a specific type of system as shown in the previous section.
2. For lower multiplicities (1, 2, and 3), the ZN method works appropriately, albeit suffering from the common disadvantages of this method such as non-robustness and large overshoots.
3. For a multiplicity of 4, ZN tuning is not very useful as it results in closed loop systems having slowly decaying oscillations of large amplitudes in the step responses. Other tuning methods are recommended.
4. For multiplicities greater than 4, the ZN rules result in unstable closed-loop systems, and are thus wholly unsuited for these types of systems. Other control strategies should be more useful.

MATLAB SCRIPT

https://drive.google.com/file/d/1kVPXNWcor5my0TLpvBMSUWF9SYef1ueZ/view?usp=sharing

[3] https://www.mstarlabs.com/control/znrule.html