Analysis, design and implementation of a suitable feedback control strategy for a dye injection process

This work is the independent work of Polarj Sapkota of Kathmandu University, Nepal (2018-2022). Contact: polarjsapkota@gmail.com | +977 9861-901-306

Table of Contents

Chapter 1 - Problem Statement

Dye injection is an important part of the textile, pharmaceutical & food industry. Dyes, both natural and synthetic are made up of different chemicals that add their distinct color to the things they're being added to. Since dyes are basically chemicals, they have their own levels of acidity and basicity along with other properties that can affect other compounds. How much they affect other compounds is decided by their usage concentration. It is obvious that the more concentrated a chemical compound is, the more it can enact its properties on another interacting compound.

Dyes, until a certain level of concentration only affects the light absorption properties of a compound and not other intrinsic properties (although they do so ever so slightly such that it's not enough to modify the intrinsic build of the compound itself). After crossing a certain threshold of concentration, they start affecting the chemical bonds more and therefore can modify the properties of the original product and create byproducts which can be harmful to users. For example, a medicine batch with a higher dye concentration than normal may modify the properties of the medicine and induce side-effects which can be extremely harmful to users. To precisely control the concentration of such dyes, we need feedback control strategies to mix the dye chemical with another compound such as water to create a solution with the required level of dilution.

The main control problem for our requirements, in one form or the other is the dye-to-water ratio. Our control system has two main goals, keeping the volumetric ratios of the dye mixture within strictly quality controlled limits and making the mixture as homogeneous as possible. Out of the two, mixing is secondary as it can be done after completion of the initial injection. The overall control philosophy depends on the type of application. For example, in the pharmaceutical industry, tight control must be implemented to make sure the concentration doesn't exceed the specified limits. The speed of the response isn't really a big issue in our process as the economic risk of producing faulty products outweighs the tiny production lag created by a slower response. But the main reason for selecting this control philosophy is the transport lag created in the mixing stage, which requires that the system be on the slower side. This makes it less susceptible to instability. Leaning towards modeling, the involved variables in the process are:

- PPM_W – measured dye/water ratio (parts per million)
- q_D – dye input flow rate (milliliters per second)
- V – valve actuation voltage (millivolt)
- V_{REF} – Valve reference voltage corresponding to desired ppm (millivolt)
- q_W – Water flow rate (milliliters per second)

Process Variables – PPM_W, V, q_W

Manipulated variables – q_D, V_{REF}

Disturbance Variables – ΔQ

Chapter 2 - Modeling and block diagram

Our system can be broken down into two main processes, the dye injection process and the mixing process. Their open-loop diagrams are shown below

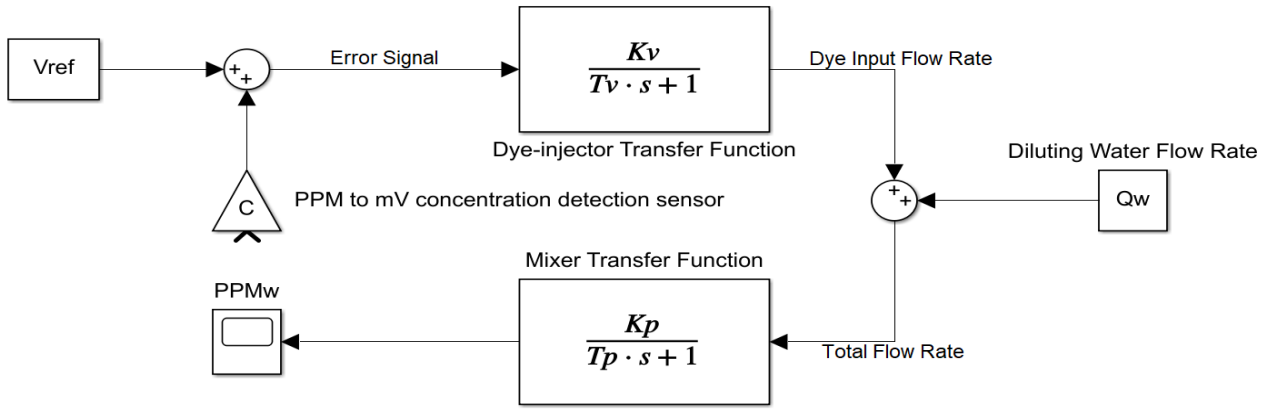


Figure 1: Open-loop configuration of the process made in Simulink

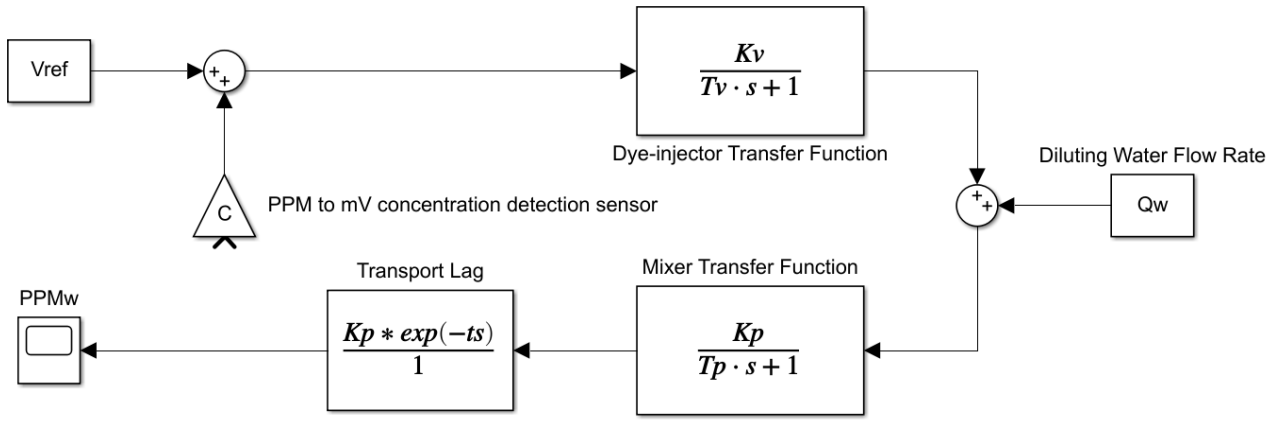


Figure 2: Open-loop configuration of the process with transport lag

The open loop diagram shows the two processes we can control. This system model can be obtained either from first principles or by usage of some experimental techniques.

Both systems are of type-1 as shown above. For the dye injection block, since the dye input rate is directly proportional to the error signal voltage, we can be sure that it is a first order system with minimal deadtime relative to the mixer and positive gain because the dye flow rate must be increased if the error increases in the positive direction.

The mixer on the other hand can be modeled as multi-order ODE or even a PDE due to the turbulent nature of the fluid being governed by the Navier-Stokes equation. This kind of modeling is extremely tedious but will provide an extremely accurate description of how the two fluids mix. However, estimates using mass balance dynamics provide very accurate results in a fraction of the time it takes to solve PDEs in real-time. And we can easily compensate for the unavailability of solutions using PDEs by using sensors to take note of the concentration of the mixed fluid. This order reduction allows us to model the resulting solution as a first-order system because the concentration of the fluid is directly proportional to. They are explained sequentially below:

2.1 First Principles Modeling

Using mass balance equations as shown in [1] can help deduct models from first principles.

2.2 Experimental Modeling

Experimental techniques include using the impulse, step and parabolic inputs to estimate parameters such as the time constant, constant gain factor and steady state errors. For example, using the step response, we can obtain the first order transfer function of form:

$$\frac{K}{sT+1}$$

To find the values of K and T, we feed a step input the system and take measurements of the output. Using these measurements, the final values of the output signal is taken in order to estimate the K value using the formula

$$\mathrm{K} = \frac{y(\infty)}{x(\infty)}$$

Then the value of T can be estimated by using the formula:

$$y(a) = y(\infty) \cdot (1 - e^{-\frac{a}{T}})$$

where by knowledge of the output y at any measured point 'a' allows us to estimate the value of T. This method is highly reliable yields accurate results for 1^{st} order systems.

Another experimental method is the method of the impulse response. Since the definition of transfer functions is directly tied to the impulse response as mentioned in several texts

& seminal papers, to quote "The transfer function is the Laplace transform of the impulse response of a signal".

An impulse response of the system wholly characterizes the response of a system at any frequency since the Laplace transform of an impulse signal is the constant 1 in the frequency domain. Therefore sampled information about the system impulse response allows us to directly obtain the system transfer function in a tabular form and apply various control strategies directly, except for when the system is of order > 2.

2.3 A heuristic treatise of transport lag

Transport lag plays a big role because it can generate a large error in the control system & easily destabilize the process. The ppm detecting sensor's position must be far enough from the mixer such that the value read by the detecting optical sensor is a good representation of the actual homogeneity of the whole liquid. This is directly dependent on the quality of the mixer itself. If the mixer generates enough homogeneity, the sensor can be placed directly at it's outlet which prevents transport lag issues to begin with. If not, the effects of the lag must be taken into consideration.

The key issue with using simple optical sensors instead of more advanced tomographic sensors for such problems is that they give localized readings of the signal i.e. optical sensors can only read the ppm concentration of the water that's in the vicinity of the sensor. If the sensor is kept far enough from the static mixer, the mixture is more likely to be more homogeneous than right at the output of the mixer The system's electrical transport lag itself isn't a problem but the problem lies in the transport lag created by the relatively low speed of water flow. If the detecting optical sensor is kept too far away from the process, the sensor is induced with a transport lag value that is proportional to the input itself multiplied by a transfer function modeled as:

$$e^{-Ts} \text{ such that input signal } \quad x(t) \rightarrow X(S) \text{ after time } T \text{ becomes}$$

$$x(t-T) \rightarrow e^{-Ts} \cdot X(s)$$

Which clearly shows that the input $X(s)$ (signal right at the outlet of mixer) gets scaled down more due to the exponential term as the value of T goes up. For our uses, this simply means that the signal at a point L, has the transport delay $T = \frac{L}{v}$ where v is the velocity of the water. v can be calculated directly from knowledge of the flow rate of liquid and cross sectional area of the mixer's pipe. Then our transport delay transfer function

becomes $e^{-\frac{L}{v}.s}$ which again shows us that T increases with increase in length, L which in turn reduces the value of the exponential term faster. For example:

v = 7 m/s, then transport lag = $e^{-\frac{L}{7}}$. It's plot is shown below:

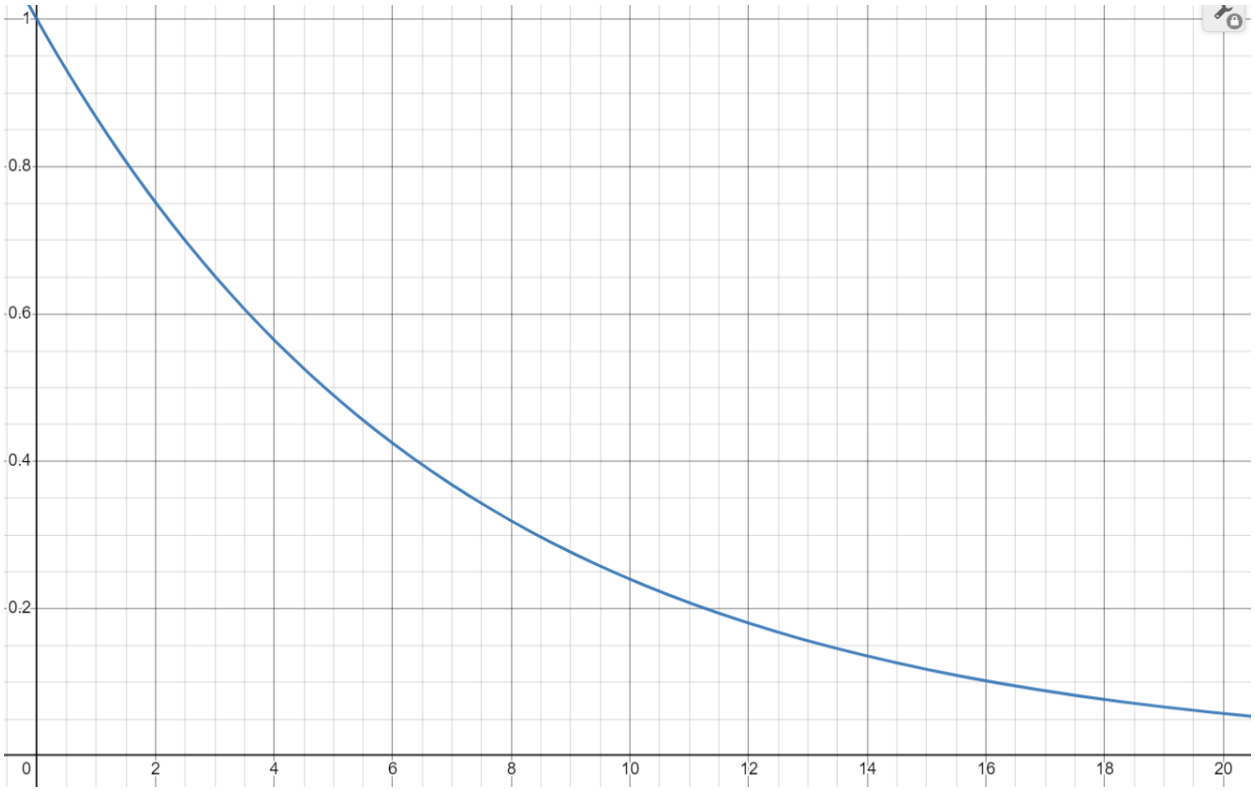


Figure 3: Transport lag equation plotted in Desmos Graphing Calculator

A set of values tabulated from the function gives a look into the problem

L (m)	Lag coefficient
1	0.87
3	0.65
5	0.49
7	0.37
9	0.28
11	0.21
13	0.16
15	0.12
17	0.09
19	0.07

Table 1: Tabulated values for the transport lag function

We can clearly see that the coefficient goes below 50% within 5m & below 10% within 16m which greatly reduces the effect of the input on the error signal of our control system. Extending this example:

$$\text{Error} \propto \text{PPM}_{\text{REF}} - e^{-\frac{L}{7}} \cdot \text{PPM}_{\text{W}}$$

Using the tabulated values,

L (m)	Error	
1	$\text{PPM}_{\text{REF}} - 0.87 \cdot \text{PPM}_{\text{W}}$	(1)
7	$\text{PPM}_{\text{REF}} - 0.37 \cdot \text{PPM}_{\text{W}}$	(2)

Table 2: Placement distance of sensor's effect on transport lag

For an unchanged PPM_{REF} we can clearly see that error equation (2) gives a greater result than (1). The actuating signals i.e. is proportional to the Error is greater resulting in greater change in the flow rate of the dye injector. But this signal is not a good representation of the actual concentration of the signal X(s) due to being heavily attenuated. The input flow rate sees a large increase due to the attenuated feedback induced large error. The larger flow rate overshoots the required PPM_{W}. This overshoot again attenuates heavily such that an even larger error is induced which further increases the flow rate. When this attenuated PPM_{W} increases beyond PPM_{REF}, the flow rate is suddenly reduced by a large margin inducing an even larger overshoot in the direction negative to PPM_{REF}. This self-reinforcing behavior keeps increasing the percentage overshoot and undershoot until the system diverges. This divergent behavior is characteristic of an unstable system, hence allowing us to argue that our system has a high chance of being unstable. If the effects of transport lag on the system dynamics are not clearly studied, this effect can make the products of the process unusable.

2.4 Process transfer function gains & time constants computation

Through the information given to us about the process, the values for each of the unknown parameters can be calculated. The given parameters are first listed.

Distance of sensor from injection point (L) = 2 m
Regulating valve steady-state gain (K_v) = 0.6 mL/mV · s
Mixing process steady-state gain (K_m) = 0.8 ppm · s/mL
Regulating valve time constant (T_v) = 0.2 s
Pipe cross-section area (A) = 5 cm^2
Pipe flow rate (Q) = 2 L/s

Also, given is the experimental data that a step change in the dye regulating valve resulted a dye concentration that is 99.3% complete in 20s. This data can be used to calculate the time constant of the mixing process.

$$ref\left(1-\dot{e}^{-t/T_p}\right)=ref\ 0.993$$

$$e^{-20/T_p}=0.007$$

$$\ln\left(e^{\frac{-20}{T_p}}\right)=\ln(0.007)$$

$$\frac{-20}{T_p}=\ln(0.007)$$

$$T_p=\frac{-20}{\ln(0.007)}$$

$$T_p=4.0308$$

Velocity of water through pipe cross-section at any point L is then give as:

Now, we plug in L=2 m to obtain the time delay value t_d for transport lag transfer function given that

$$v=\frac{L}{t_d}$$

$$t_d=\frac{2m}{4\frac{m}{s}}$$

$$t_d=0.5s$$

And since the mixing process results in a concentration that is 99.3% complete within 20s at the measuring point. Since we have information about this instant of time. We can consider the mixing process's gain as an amplifier block which has a gain $K_m = 0.993$.

Regulating Valve		Mixing Process		Transport Lag	
K_V	T_V	K_P	T_P	K_M	t_D
0.6 mL/mV · s	0.2s	0.8 ppm · s/mL	4.0308s	0.993	0.5s

Table 3: Computed constant parameters for the open-loop system

This gives us the overall process transfer function

$$G_{PRC} = G_A G_P G_M = \frac{0.6}{0.2s+1}\frac{0.8}{4.0308s+1}0.993\,e^{(-0.5s)}$$

$$G_{PRC} = \frac{0.5912 \cdot e^{-s/2}}{(s+5)(s+0.2481)}$$

which is a 2nd order transfer function with characteristic roots or poles at -5 and -0.2481. The system is constructed in simulink and a step input is fed to obtain the step-response.

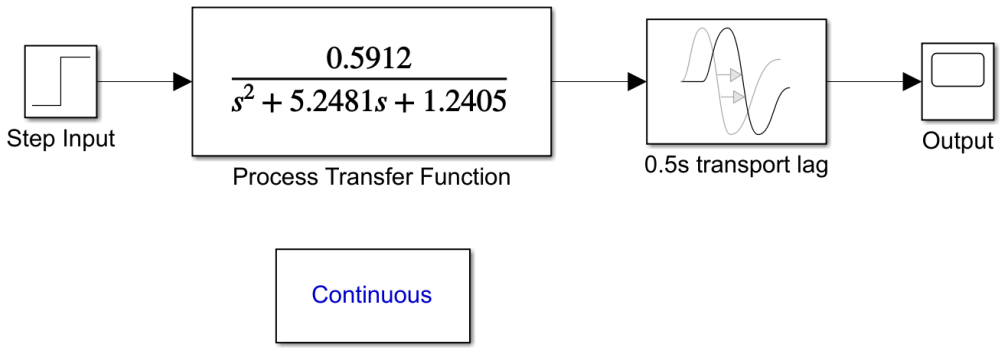


Figure 4: Open-loop system setup for step-response

The step value is set to 1, which gives us the following response

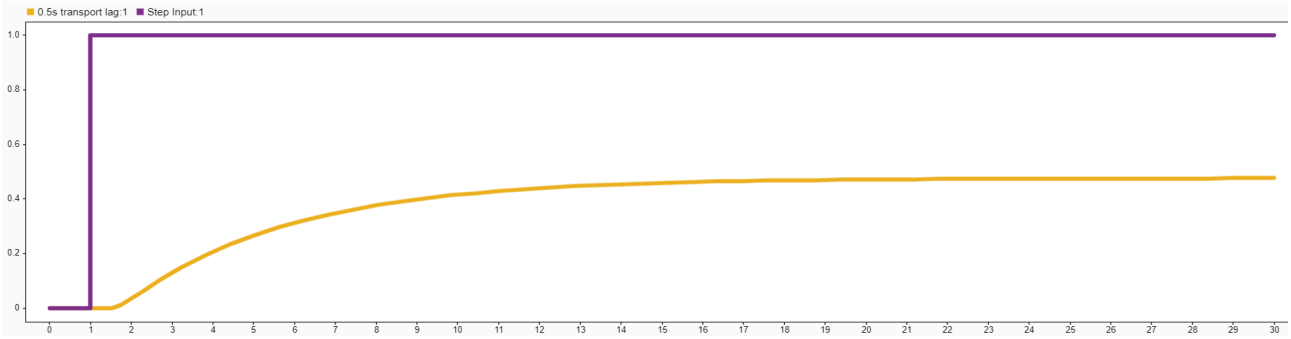


Figure 5: The step-response of the open-loop system

The system step response exhibits a large steady state error of 0.52 which is in large part due to our regular culprit, the transport delay.

Now, for empirical tuning, we need to fit a 1st order transfer function of the form

$$G_{PRC} = \frac{K\, e^{-as}}{bs+1}$$

Before starting our hit-and-trial process, by observation we can see that the process reaction curve has a peak at 0.48 which can be set directly as the gain i.e. K = 0.48 & a = 0.5 by direct comparison with the transport delay function in our original transfer function. For an estimation of b, we can compute the time taken by the original step-response to reach 63% of its final value i.e. 63% of 0.48 which is 0.3024. This time was found out from the plot to be ~5.7607 s.

With these starting points, hit-and-trial was performed to obtain the following values:

$$K = 0.48,\ a = 0.6s\ \&\ b = 4.2$$

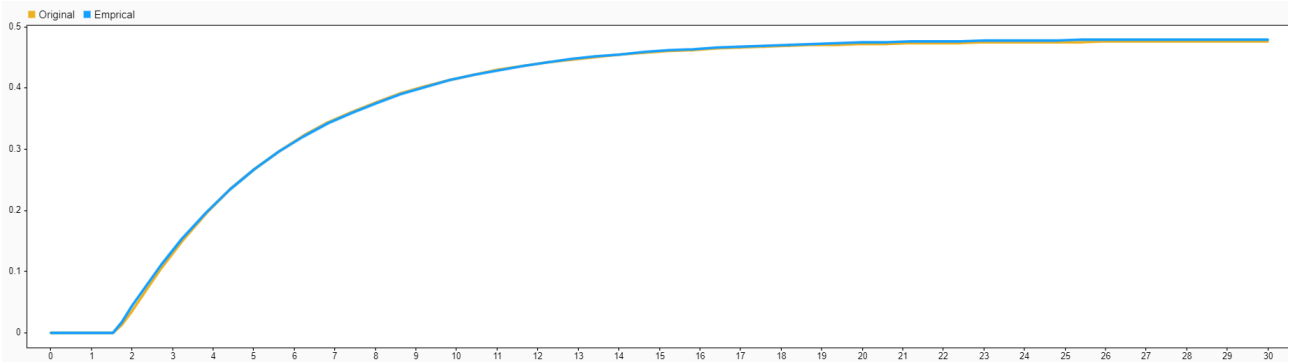


Figure 6: Step-response of empirically fitted transfer function with dead-time in blue

2.5 A rigorous treatise on disturbance

The only realistic way in which disturbances can occur in this system is in the flow rates. As the mixer is a static mixer, it can be relied upon for long-term stability. The optical sensor that measures the ppm concentration of the dye too is a very accurate sensor such that it too will have a considerably low deviations from actual measurements. The only unknown source parameter here is the input flow rate Q which directly affects t_D. This means that the disturbance can be modeled within the transport delay function by making adding a small disturbance to the flow rate in the form

$$Q = Q + \Delta Q \qquad (1)$$

Using this fact we can define a small change in t_D due to small change in Q as

$$t_D' = t_D + \Delta t_D \qquad (2)$$

$$\frac{LA}{(Q+\Delta Q)} = \frac{LA}{Q} + \Delta t_D$$

$$\frac{LA}{(Q+\Delta Q)} - \frac{LA}{Q} = \Delta t_D$$

this gives us,

$$\Delta t_D = \frac{LA}{Q}\left(\frac{-\Delta Q}{Q+\Delta Q}\right)$$

$$\Delta t_D = 0.5\left(\frac{-\Delta Q}{0.002+\Delta Q}\right) \qquad (3)$$

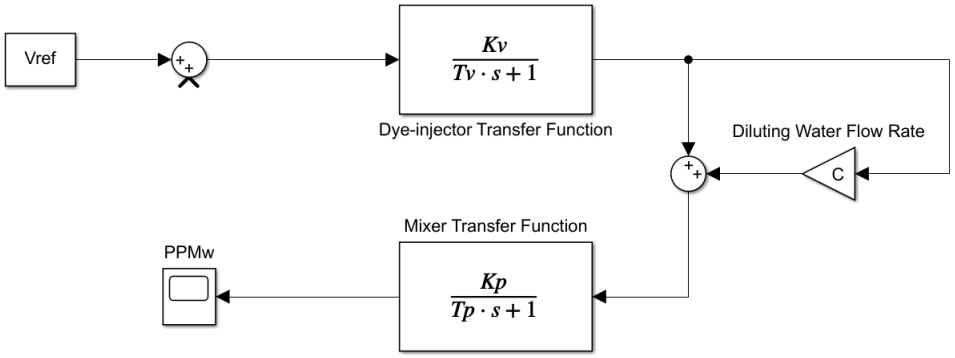
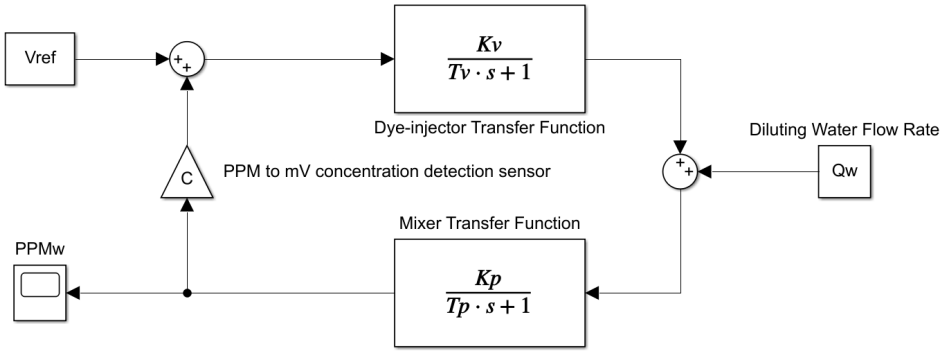
This clearly shows us the dependence of change in flow rate in the time rate. The larger the disturbance in the flow, the larger the effect on flow rate. This allows us to break down the transport delay function into two parts

$$G_M = 0.993\, e^{-0.5s} \cdot e^{0.5\left(\frac{\Delta Q}{0.002+\Delta Q}\right)}$$

Which models our disturbance transfer function

Chapter 3 - Control Strategies

A comparison of possible control strategies can be communicated through a table

Our process variables: PPM_W, V, q_W	
Feedforward	q_W can be treated as a disturbance variable in the process and be corrected instead of the dye injection rate. This method takes in the dye input flow rate as a feeding signal that drives the output water flow rate. The main goal is making $q_W \propto q_D$  *Figure 7: Feedforward Strategy* As can be seen from the diagram, this control strategy has no feedback loop and hence must be tightly controlled by V_{REF} along with an in-depth knowledge of the system dynamics. Such a system can be very fast but requires significant expertise of the system dynamics at the disposal of the engineer. The same V_{REF} here controls all three process variables.
Feedback	The process can be controlled with lesser knowledge of system dynamics. As  *Figure 8: Feedback Strategy* seen, the main process variable, PPM_W is fed back into the summing block to generate an error signal & track V_{REF}. This method is reliable and can easily be tuned by technicians. The issues with such a strategy is the transport lag it can create, which can be compensated for by fluid based

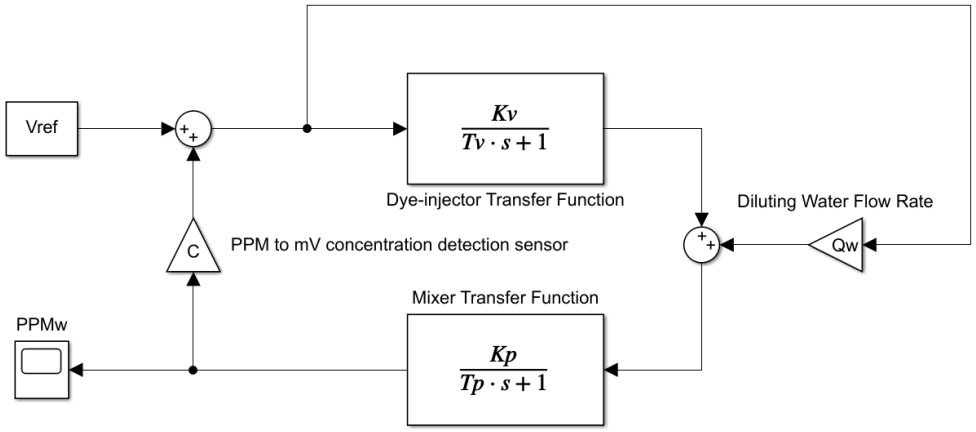
	controllers at the mixers, which are an expensive control option.
Cascade & Combination of strategies	This strategy utilizes both of the above methods to create a hybrid control model. The variable PPM_W is fed back to obtain V and the V is fed forward to create an actuating signal for the 'Diluting Water flow' forward path. These two together form a cascade strategy where one sub-system provides set-points for the other. Here the main output PPM_W will be controlled by Q_W as well so the system has a lesser chance of being unstable due to transport lag.  *Figure 9: Cascade Strategy* In this method, all of the process variables are utilized. PPM_W is fed-back, q_D & V are fed ahead. This gives the designed a chance to fine tune the process parameters even more and generate faster responses.

Table 4: Comparison of control strategies for each process variable

3.1 Recommendations

- In systems where transport lag is an issue, consider using a combination of both strategies. Q_W must be tightly controlled and therefore integrating a feedforward path into it's control is recommended
- For finer control of PPM_W, especially in pharmaceutical applications, both feedback and feedforward strategies cascaded with P, PI or PID controllers is recommended in order to have finer control over the entire fluid flow such that the system stability can be fully assured
- For control of V, normal feedforward strategies are recommended as modifying the input signal too much can lead to unwanted disturbances in the response, usually integrating themselves into the process as noise which is undesirable.

Chapter 4 - P, PI & PID Control

P, PI & PID controllers are widely used in industrial applications because of their robust nature, ease of use and replacement & most of all, simple heuristic methods of tuning to meet system specifications.

P

$$G(s)=K_p$$

These are the simplest of controllers. They are simply amplifiers that amplify a signal input to them. This gives meaning to the fact that they are named proportional controllers. They are fast and responsive but are prone to overshoot quite frequently unless the error signals are small enough.

In fast & relaxed control processes, such fast response is desirable.

PI

$$G(s)=K_p+\frac{K_i}{s}$$

Proportional-Integral controllers are more advanced controllers where the Integral term corresponds to a memory element in the circuit. This strategy prevents the system from suddenly responding to an input signal too fast and allows it to track the set point smoothly. Another added benefit is due to it's inherent memory element, which even after the error signal decays, it's remnants keep adding up over time to build up and correct any steady state error which prevents the system from correctly tracking the reference.

In tight control processes, such slow response is desirable.

PID

$$G(s)=K_p+\frac{K_i}{s}+K_d s$$

Proportional-Integral-Derivative controllers are the most advanced of these three and are the aforementioned controllers that are widely used in industry. The derivative part in this controller adds a rate of change element which allows the system to counterbalance the negative effects of the integral elements and negate the effects of overshoot and decrease the rise time with very little to no overshoot depending on the selected value of the gains. Derivative controllers aren't used in isolation as a step change in the reference signal leads to a large output which destabilizes the process easily.

Fast & tight control rocesses need accurate response of the PID

Note: Each of these controllers are to be selected based on the criterion mentioned in bold.

4.1 Expected Reaction Curves

P

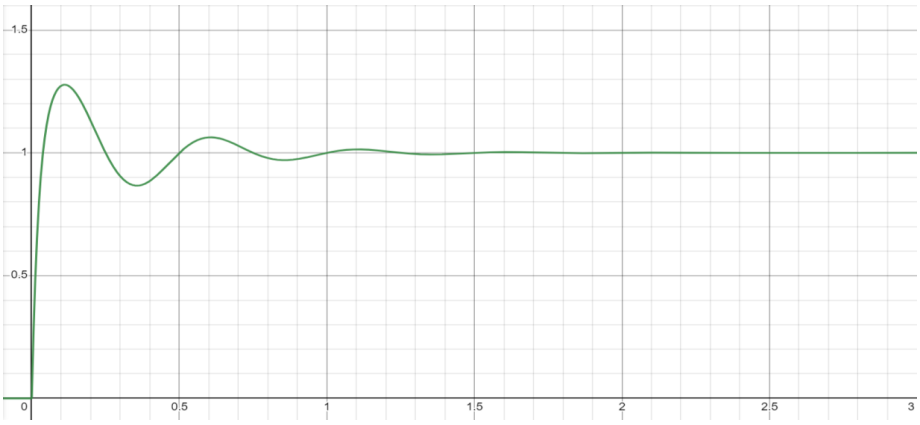


Figure 10: P controller cascaded response plotted in Desmos

PI

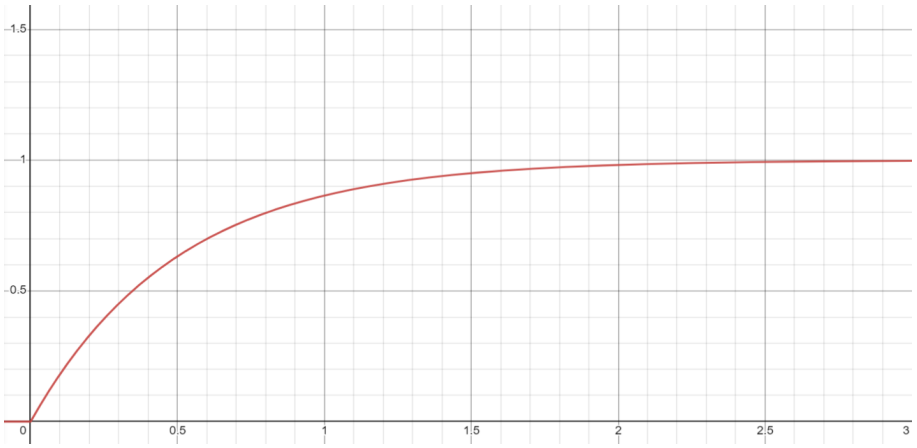


Figure 11: PI controller cascaded response plotted in Desmos

PID

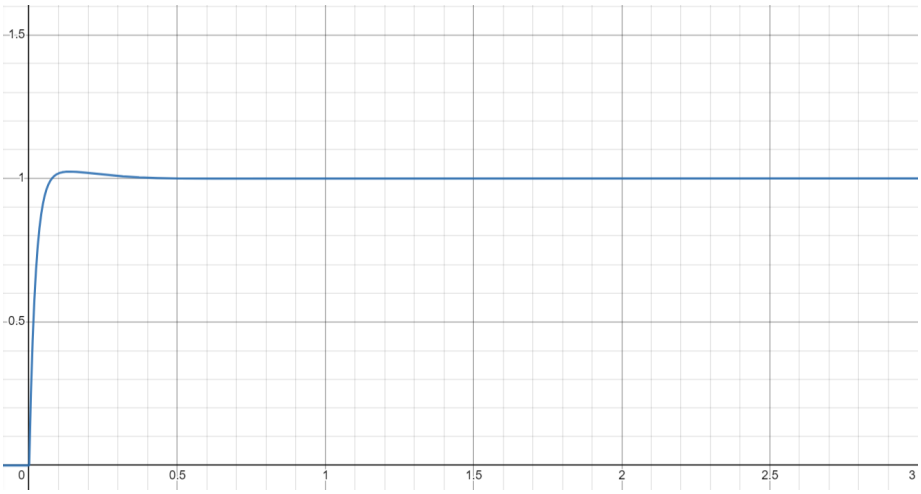


Figure 12: PID controller cascaded response plotted in Desmos

4.2 Tuning the PID controller using Cohen-Coon method

With knowledge of design trade-offs for each type of controller, we can decide what type of controller is best suited for the job. We make a few observations through our process reaction curves:

- The system severely undershoots to create a large steady-state error, which can be taken care of by using a simple proportional controller whose gain is set such that it makes the final steady state value close to 1 (multiply by a gain of 1/0.48), without changing the rise times or any other parameters. The result:

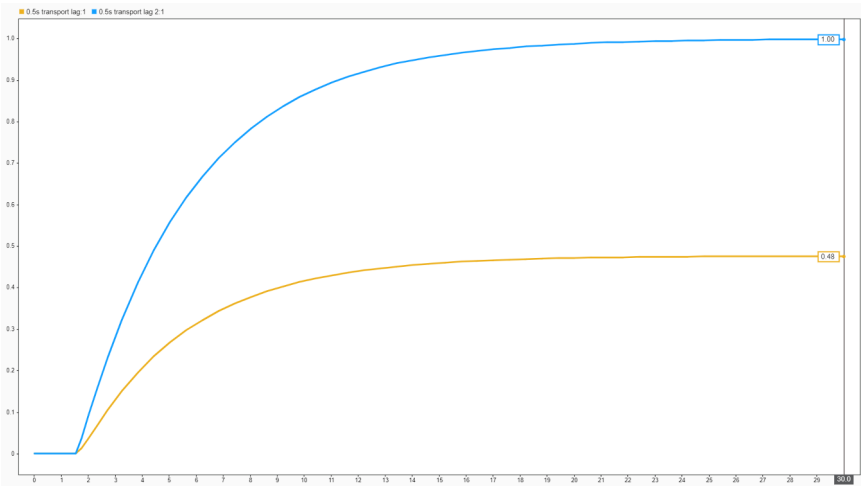


Figure 13: Making the steady-state error zero empirically with a P controller

This small improvement without closing the loop improves steady-state error only.

- Rise time is very slow, which can be improved by introducing a PI controller. It reduces the rise time but introduces overshoot and longer settling time. Steady-state error is reduced. This can be tuned to meet certain performance specifications. A PI controller with Kp = 8 & Ki = 2 has the following reaction curve

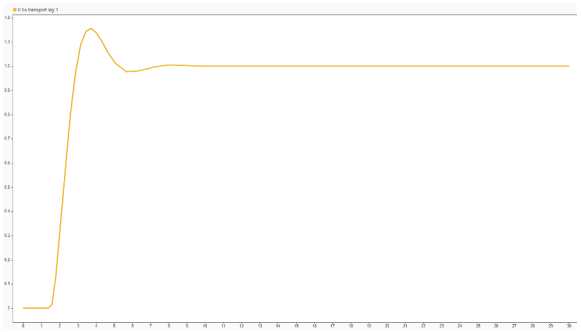


Figure 14: PI embedded reaction curve

Cohen-Coon semi-empirical method can now be used to gain better performance.

$$G_{PRC} = \frac{K\, e^{-t_d s}}{\tau s + 1}$$

Controller	Cohen–Coon
P	$K_c K = \left(\frac{\tau}{t_d} + \frac{1}{3}\right)$
PI	$K_c K = \left(0.9\frac{\tau}{t_d} + \frac{1}{12}\right)$ $\tau_I = t_d \frac{30 + 3(t_d/\tau)}{9 + 20(t_d/\tau)}$
PID	$K_c K = \left(\frac{4}{3}\frac{\tau}{t_d} + \frac{1}{4}\right)$ $\tau_I = t_d \frac{32 + 6(t_d/\tau)}{13 + 8(t_d/\tau)}$ $\tau_D = t_d \frac{4}{11 + 2(t_d/\tau)}$

Figure 15: Cohen-Coon tuning rules

From knowledge of our empirically fitted data,

$$t_D = 0.6$$

$$K = 0.48$$

$$\tau = 4.2$$

Computing PID parameters

$$K_C = 15.2778$$
$$\tau_L = 1.3939$$
$$\tau_D = 0.2127$$

Computing K_P, K_I & K_D

$$K_P = 15.2778$$

$$K_I = 10.9605$$

$$K_D = 3.2496$$

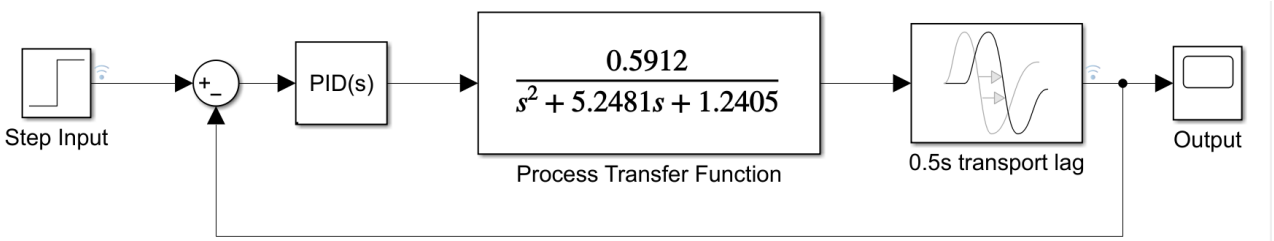


Figure 16: PID controller embedded transfer function

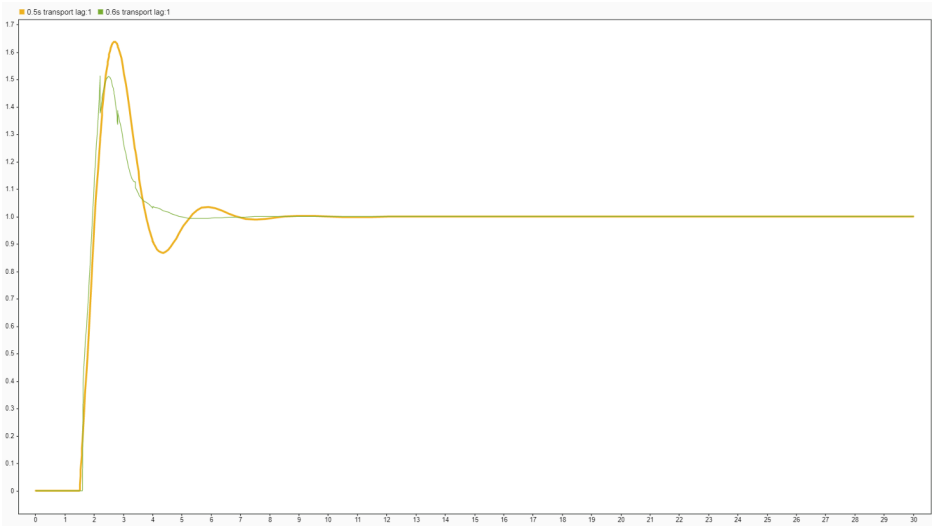


Figure 17: Reaction curves

This exhibits a slight undershoot and a large overshoot. Although the rise time and steady-state errors are considerably reduced. After some hit-and-trial with the obtained values, multiplying K_D by 1.3 & K_I by 1.05 yields the following reaction.

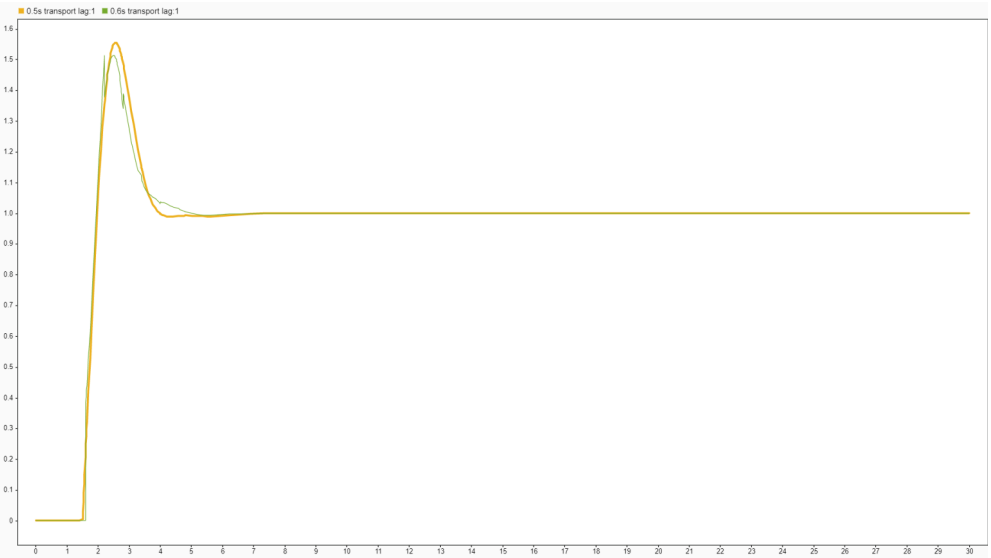


Figure 18: Reaction curves after modifications to gains

This is a good response with the system settling within 2 seconds.

4.3 Tuning using MATLAB's pidtune function

Using the pidtune function, we compute P, PI & PID response

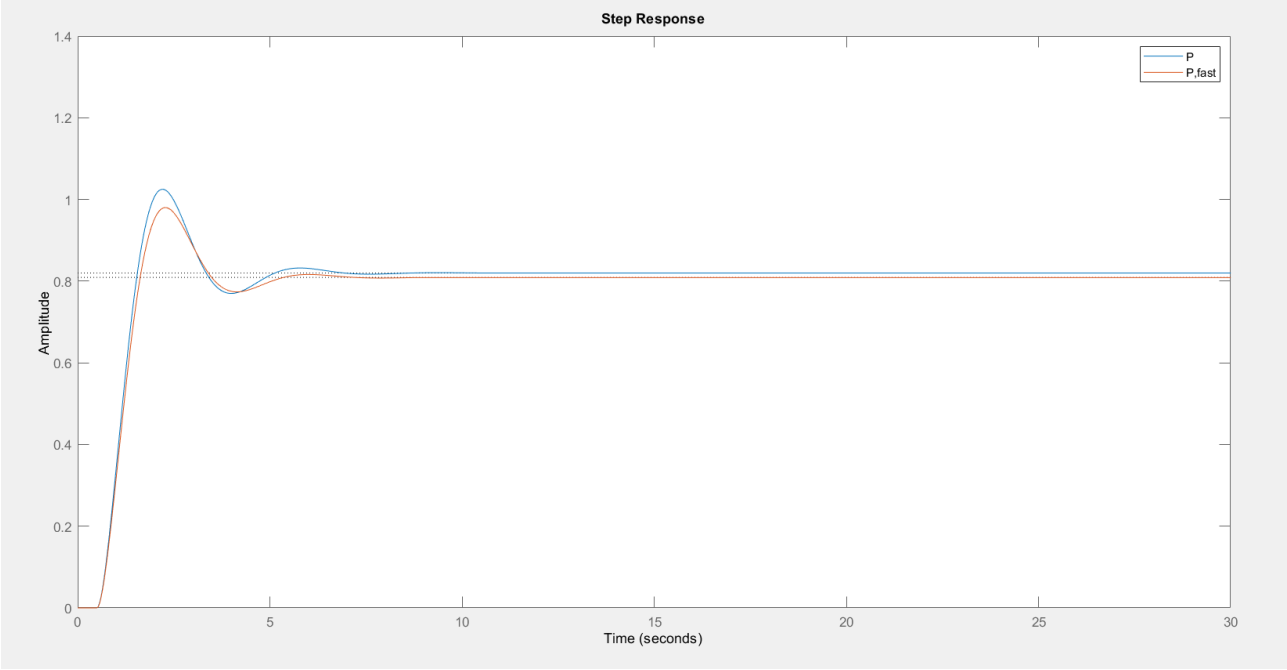


Figure 19: Tuned response with P controller ($K_P = 9.55$)

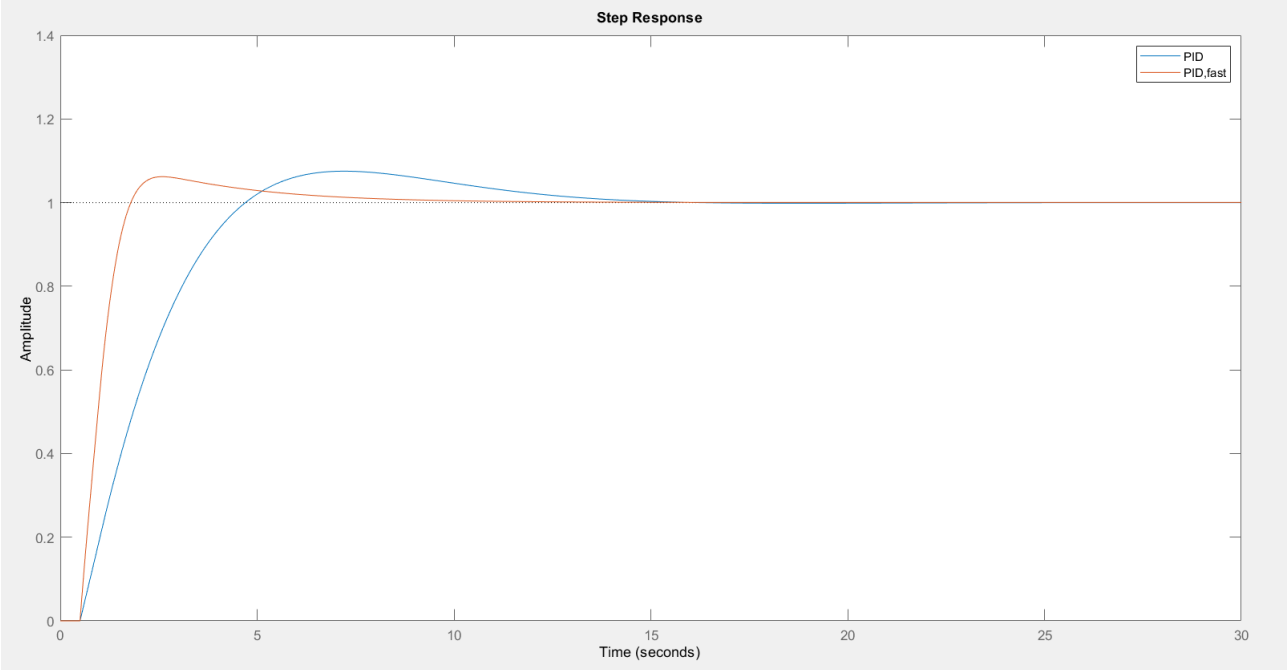


Figure 21: Tuned response with PID controllers ($K_P = 3.24$, $K_I = 1.27$, $K_D = 0.663$)

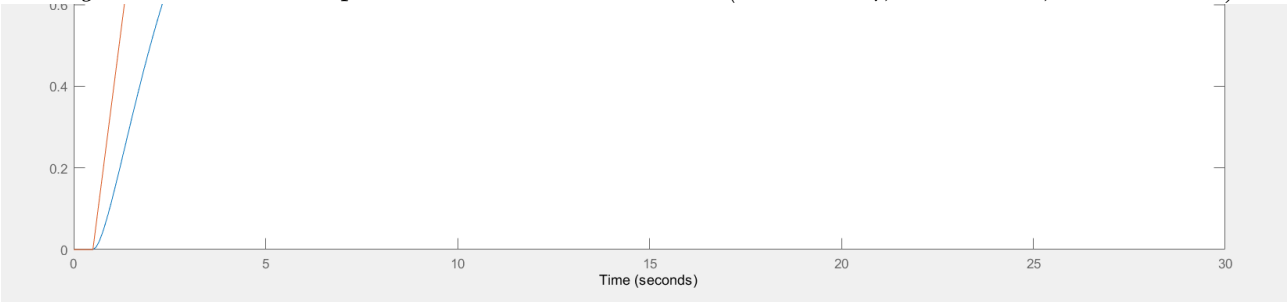


Figure 20: Tuned response with PI controller ($K_P = 3.2$, $K_I = 1.26$)

4.4 Comparison

Using the Cohen-Coon method, we we able to achieve best control action through the PID controller. This method had a settling time of 2s but percentage overshoot was close to 55%. The pidtune function of MATLAB gives a tighter control in exchange for a larger settling time as seen through both of its PI & PID controllers. Depending on the application, if tighter control is required, for example in pharmaceuticals, the pidtune() tuned parameters must be used. In case of a slightly more relaxed but fast control, for example in car coloring, the Cohen-Coon tuned controllers can be used

4.5 Simulations: Tracking, Disturbance Rejection & Comparisons

4.5.1 For semi-empirically tuned parameters without disturbance

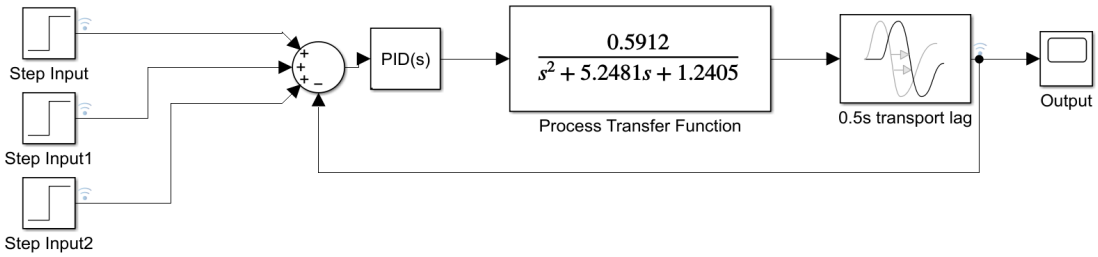


Figure 22: 3 step functions set up to turn on at regular intervals

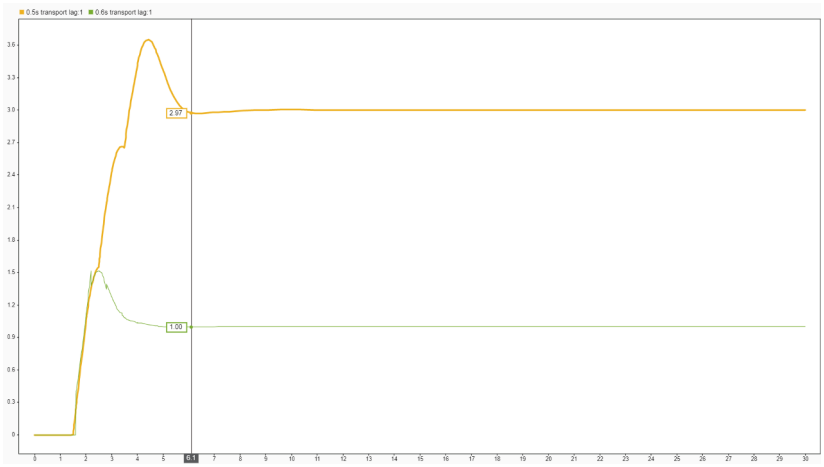


Figure 23: System settles down in 6.1 s (2 sec intervals)

Subjecting it to steps that occur in 6 second intervals

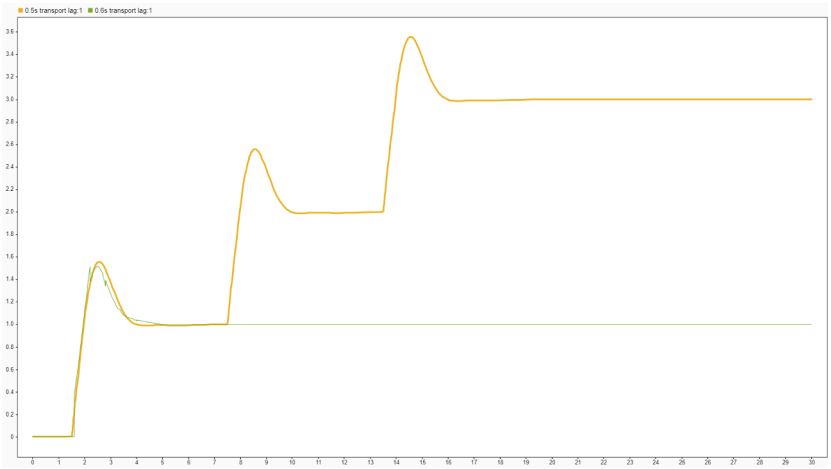


Figure 24: Each step settles down in 2 s (6 sec intervals)

4.5.2 For semi-empirically tuned parameters with disturbance

Lets set an increase of 0.1 L/s in the flow rate which translates to a change of 0.0001 m^3/s. Computing the new time delay using equations (2) & (3).

Plotting set-point tracking with this new time delay in the original transfer function.

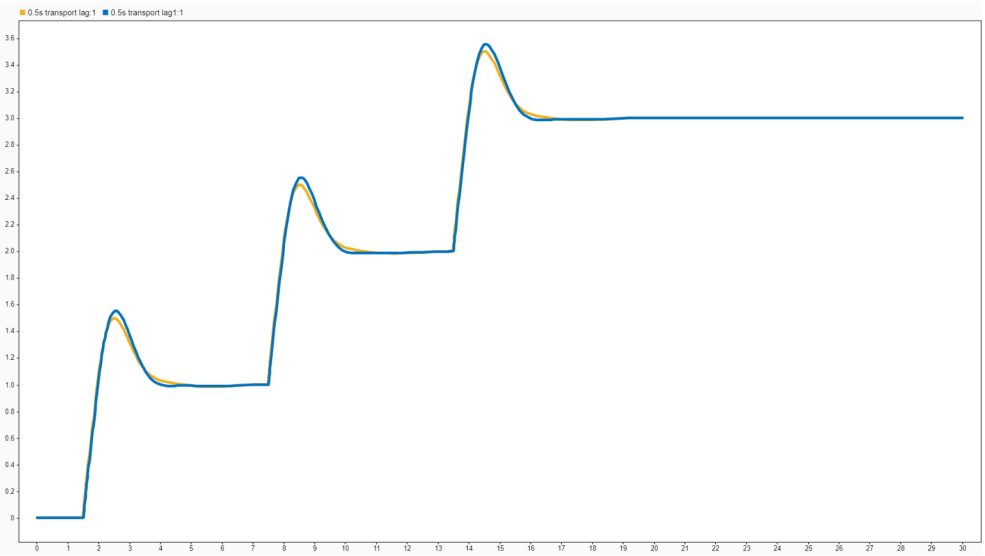


Figure 25: 0.1 L/s disturbance induced response in blue & normal response in yellow

Plotting for a larger disturbance of a decrease in 0.5 L/s which gives new $t_D = 0.6667$ s

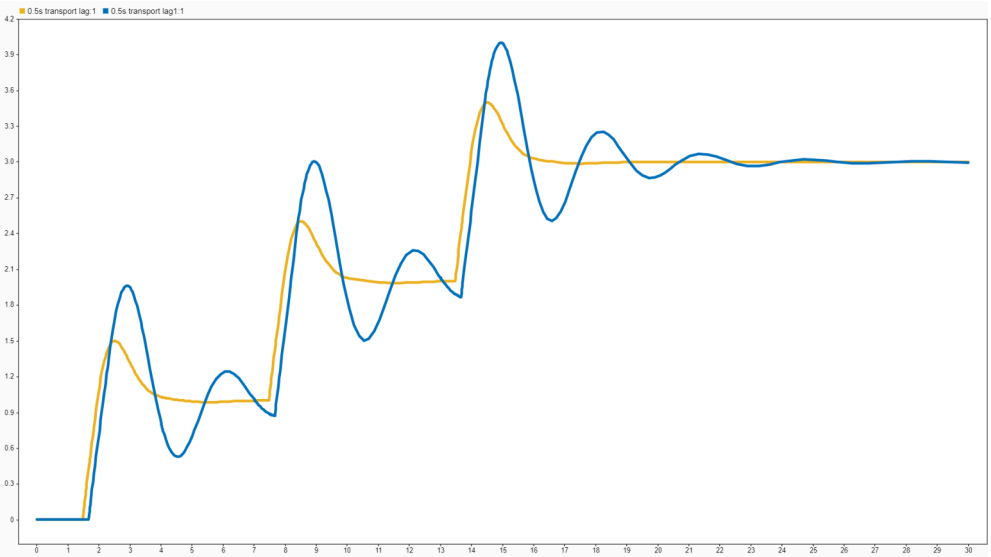


Figure 26: 0.5 L/s disturbance induced response in blue & normal response in yellow

This shows that our system performs well even for larger disturbances (Settling time ~ 9s)

4.5.3 MATLAB's pidtune tuned controller with and without disturbance

In the following plots, a large disturbance of 0.5 L/s is induced.

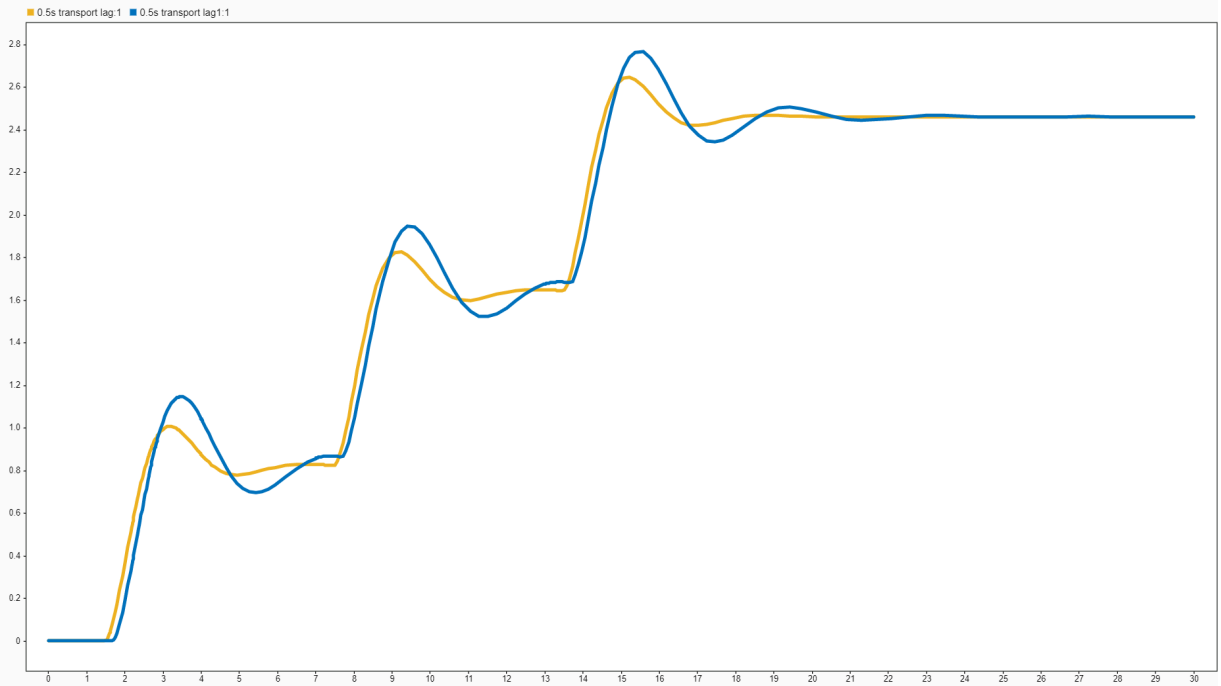


Figure 27: P Controller disturbance induced response in blue & normal response is yellow

A

a

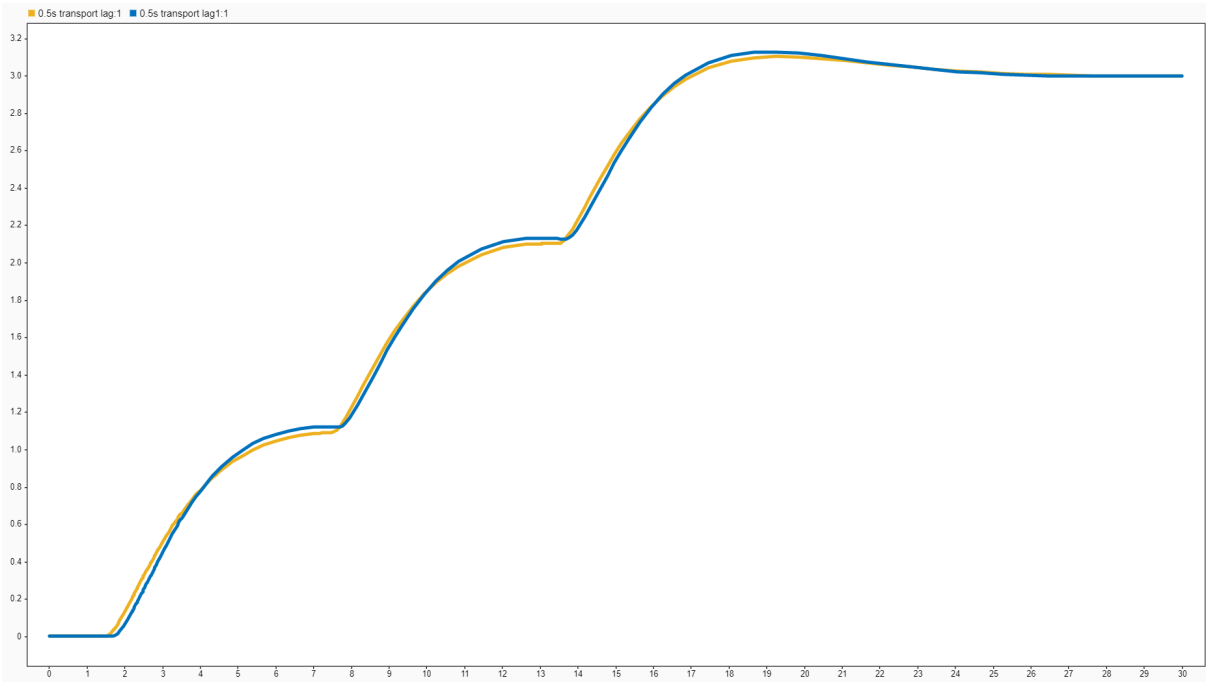


Figure 28: PI Controller disturbance induced response in blue & normal response is yellow

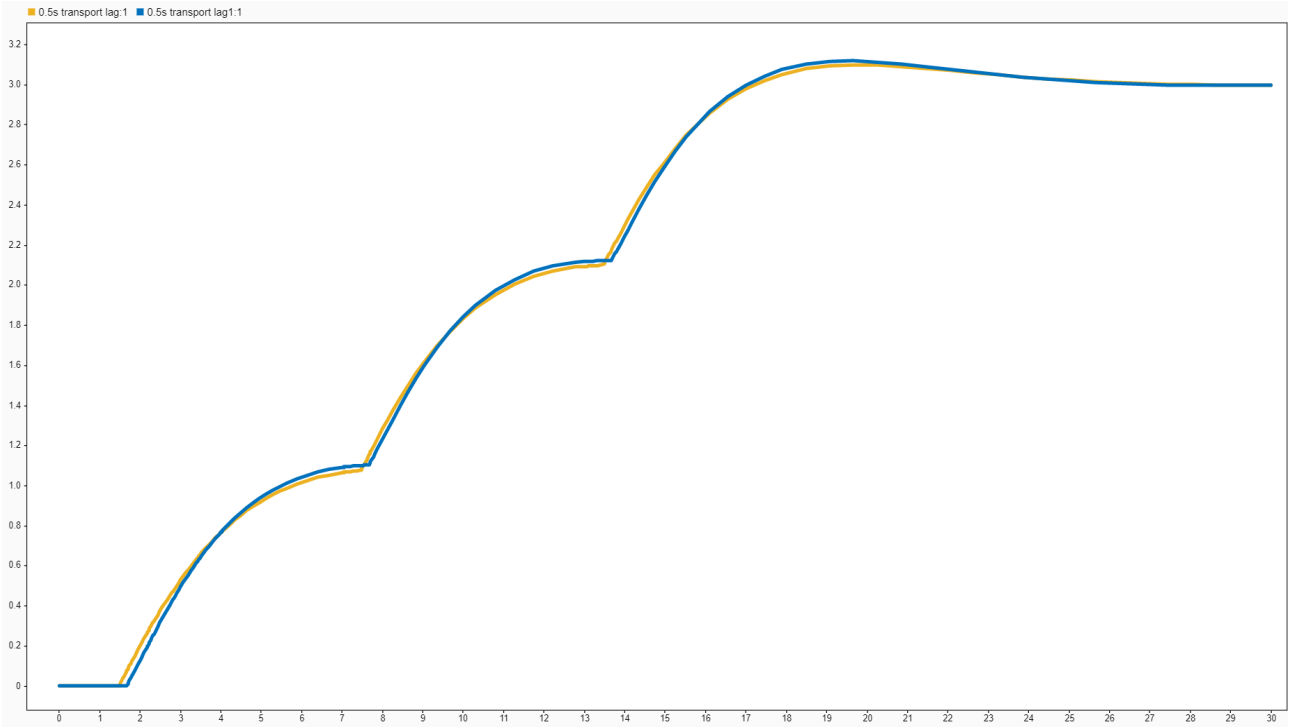


Figure 29: PID disturbance induced response in blue & normal response is yellow

As shown in section 2.7, the reaction curves of P/PI/PID controllers tuned using this method had a **trade-off of slower response against lesser overshoots** and this clearly shows in this set-point tracking example as well. Therefore for processes that might have larger process disturbances, controllers with parameters calculated using the pidtune function can be used. This, in addition to the comments in section 2.8 must be taken into consideration while selecting the controller specifications according to the application.

4.6 Stability Analysis

Taking our open loop transfer function and using MATLAB's **ltiview** function allows us to analyze the stability of our system. Before that, we take a look at our open-loop transfer function and find it's characteristic roots (poles) using MATLAB, where we approximate our time delay using the **pade** function. The obtained roots are:

-7.3575 + 7.0125i

-7.3575 - 7.0125i

-9.2799 + 0.0000i

-5.0070 + 0.0000i

-0.2481 + 0.0000i

All of which are in the left half plane. Subjecting them to feedback moves them which can be seen through the root locus shown below.

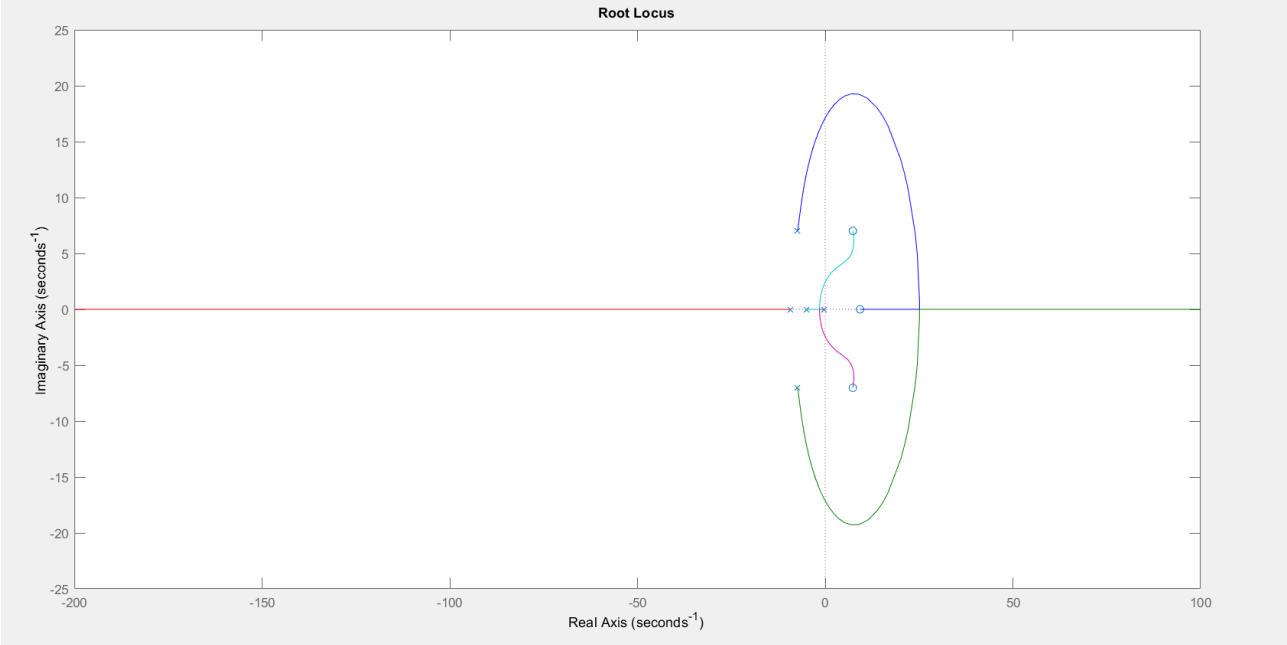


Figure 30: Root-locus plot for our process

We can see clearly that all of our closed loop poles move as we increase the gain. From the root locus, it was found that our poles cross over to the right-half plane at gains 517 & 23, hence the closed loop gain must be less than both 23 & 517 for the system to be stable. Hence the gain must be less that 23 for system to be stable. At K = 23, the system is marginally stable (oscillating).

From the bode plot, it is clear that the system is closed-loop stable albeit with a phase margin of only 1°. The stable nature of our system can be made clearer by Nyquist Plot.

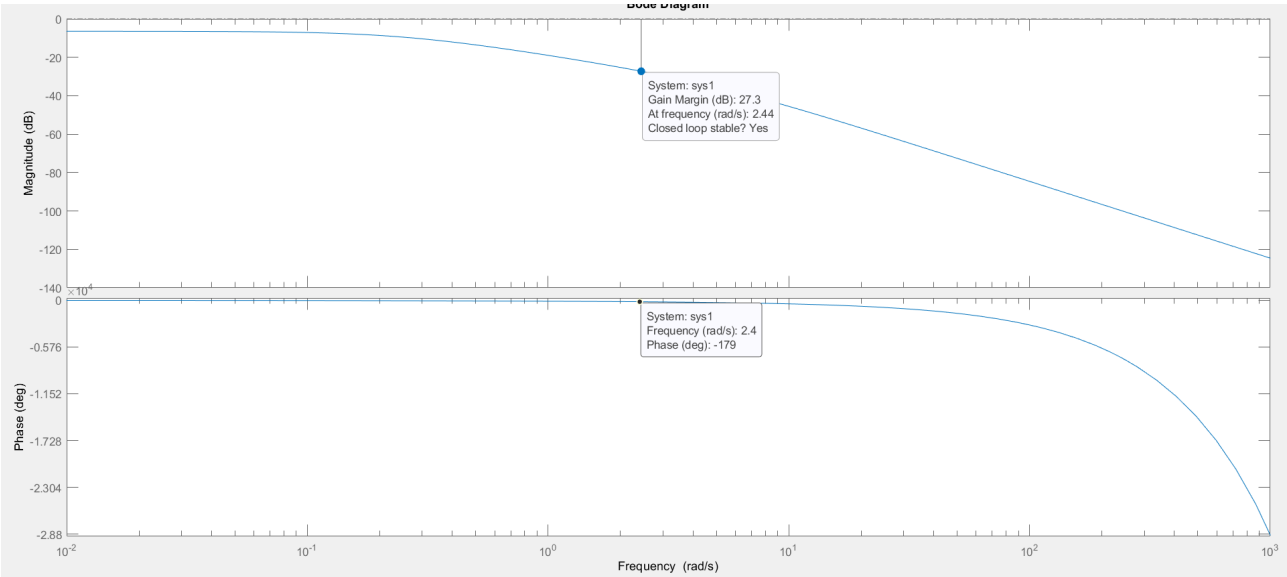


Figure 31: Bode plot of open-loop process

The Nyquist plot clearly shows that our system doesn't encircle -1 which indicates strongly that our close loop system under linear feedback will be stable

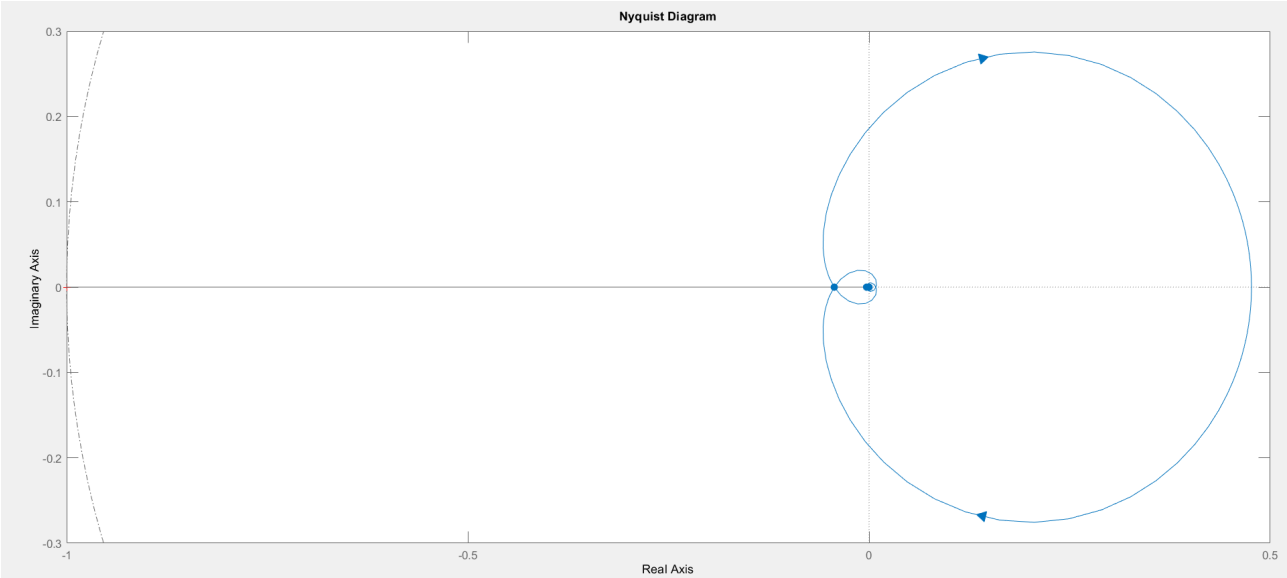


Figure 32: Nyquist plot of the open-loop process

Chapter 5 - Hardware Requirements

Usage of control hardware for this system has the basic requirement of being able to execute the discretized control action fast. This can be easily accomplished by any industrial grade PLC (Programmable Logic Controller), which can be programmed using the easy-to-use ladder logic. The minimum time taken by the controller to perform the control action must be less than 1% of the dead-time of the open loop process. For our case, since the transport lag is 0.5s, our PLCs sampling period must be no more than 5 ms during which it must be able to sample the data, perform all required computations and send out the control signal required to actuate the valve to its appropriate set-point. To this end, a good product line is available from the company Automation Direct.

Automation Direct's Cx-xxCPU line of products provides good solutions for relatively lo prices. One of their PLCs suitable for our needs is the C2-03CPU which has a contact execution time of < 2 us [5]. This means that one boolean logic step takes less than 2 us to execute. For execution of our programs in case of usage of a PID. For typical discrete PID controllers using the velocity form of PID control, there are 7 addition operations and 3 multiplication operations. Each PID computation is certain to be complete in < 5ms given the contact execution time being such a low value making the C2-03CPU is a good choice.

Cost	$197
Availability	Currently in stock Available in Automation Direct's onine stores and Amazon.

Since it is basically a CPU, the controller allows a lot of industrial rigidity combined with a limited level of flexibility with total memory of 8K (total 8K ladder steps). This rigidity is in place to ensure that the device doesn't exceed its design sample period. Our operations translate to no more than 11 bytes of information. Considering each variable is 4 bytes. Since we have a total of 12 variables in the velocity form of the PID controller. 1 byte is required to store the resulting control action. Let us consider an additional 4 variables for temporary uses during the program. Therefore our program uses only 64 bytes of memory. This is only 0.78% of the total available memory. Therefore the available storage is enough for much more complex applications such as gain scheduling and optimal control as well. Hence we have a robust controller available at a reasonable price.

Chapter 6 - Safety

In case of emergencies, for example when the input flow rate of water goes below 10%, a shutdown protocol can be initiated as t_D for this condition will be 10s which induces instability in the system. The dye concentration is too high for these conditions.

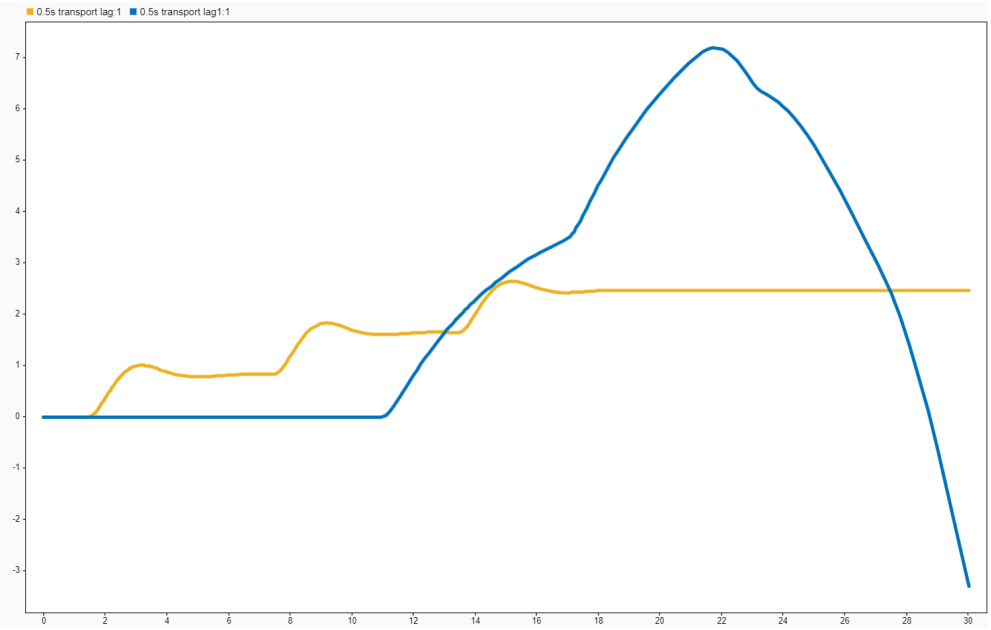


Figure 33: Divergent behavior for t_D = 10s

In this condition, we can see that the concentration of the dye diverges to uncontrollable values. This happens for $t_D = 2$ as well

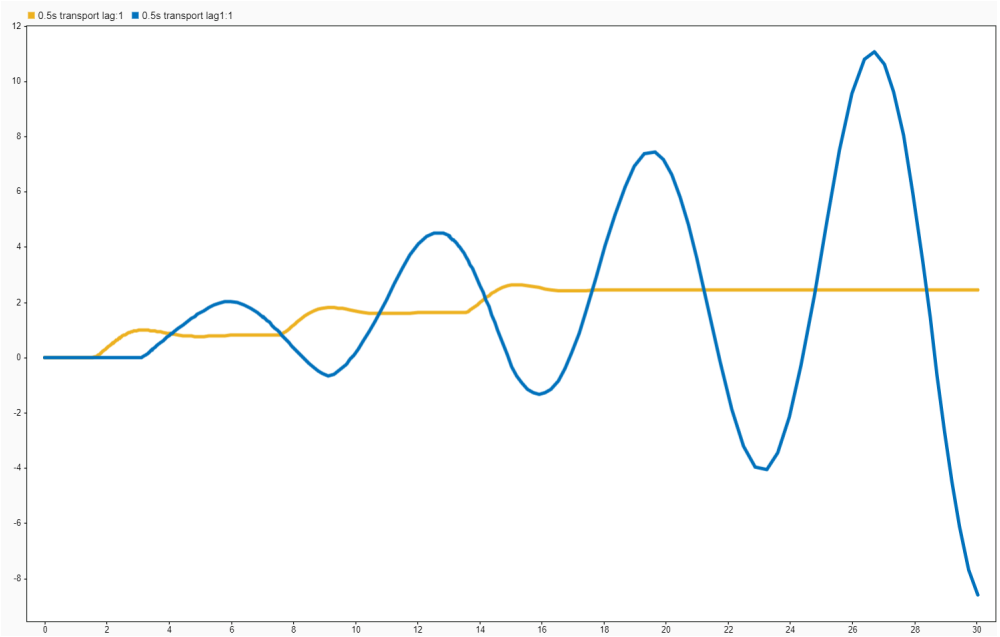


Figure 34: Divergent behavior for t_D = 2s

Two protocols can be set in place for this:

1. System shutdown or restart & operator notification for $t_D > 1s$
2. System shutdown or restart & operator notification for dye concentration > process 3-sigma QC (Quality Control) limit. Here sigma represents the standard deviation.

After execution of these safety functions, the following checks must be completed if the system doesn't resume normal operation after being restarted:

1. Check input pipes for leakages
2. Check control valve for defects
3. Check sensor placement and integrity
4. Clear PLC's memory and reload the program

A Safety Instrument System architecture can be embedded into the process loop in order to constantly monitor the system for abnormal values. In our case, since only the flow variable Q poses a serious threat, we can place a feedback path that shuts down the system set-point to zero once output crosses a certain threshold as shown in the diagram.

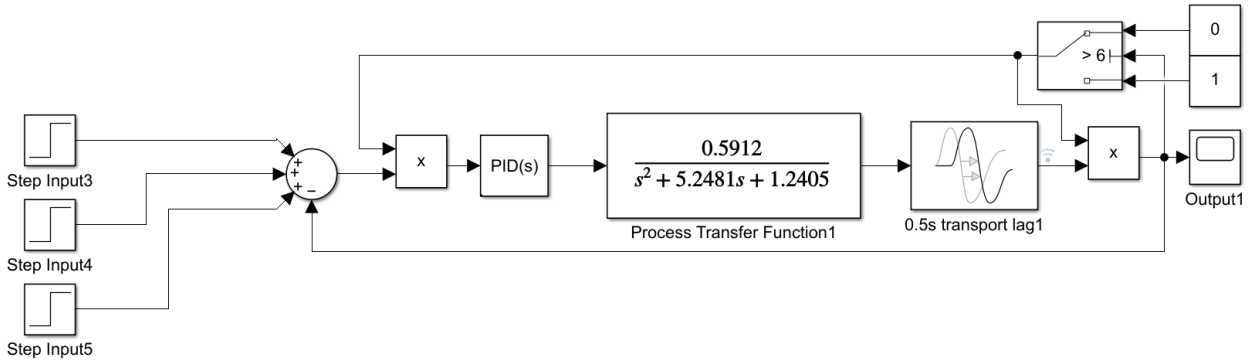


Figure 35: Safety Instrumentation System Architecture

A

This system shuts down when the output value exceeds a certain threshold, indicating that the system is diverging sharply. Therefore, the conditional switch is activated, multiplying both the error function and the output by zero, shutting down the system. This is demonstrated with a simulated shutdown of the system.

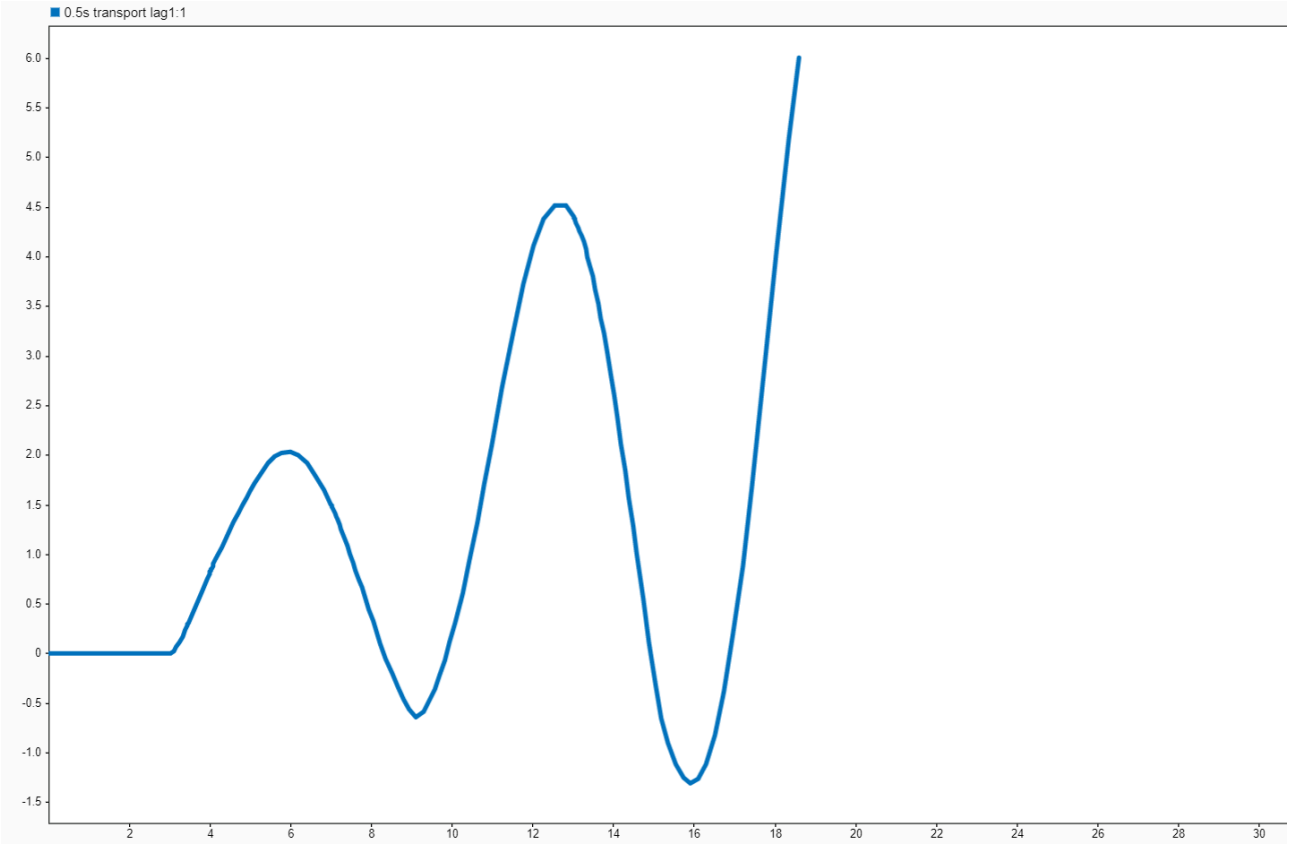


Figure 36: Simulated shutdown of system during divergent behavior

Such a SIS (Safety Instrument System) can easily detect and shutdown the system. Further automation and correcting the divergent behavior by well timed introduction of additional state-feedback is out this project's scope & the subject of clearing faults.

Chapter 7 - References

1. L. Schiop and M. Gaiceanu, "Mathematical modelling of color mixing process and PLC control implementation by using human machine interface," 2010 3rd International Symposium on Electrical and Electronics Engineering (ISEEE), 2010, pp. 165-170, doi: 10.1109/ISEEE.2010.5628522.
2. K. Ogata, Modern Control Engineering. Englewood Cliffs, NJ: PrenticeHall, 1970.
3. A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and systems*. Pearson Education Limited, 2014.
4. N. Nise, *Control Systems Engineering*. John Wiley & Sons, 2000.
5. Automation Direct, “Click PLC family overview” [Online]. Available: https://cdn.automationdirect.com/static/specs/c20xcpu.pdf. [Accessed: 14-Nov-2022].

Appendix

MATLAB source-code for pidtune, sourced from MATLAB's official documentation

```
[C_pi,info] = pidtune(sys,'PI');

T_pi = feedback(C_pi*sys1, 1);

[C_pi,info] = pidtune(sys1,'PI');

T_pi = feedback(C_pi*sys1, 1);

[C_pi_fast,info] = pidtune(sys1,'PI',1.0);

T_pi_fast = feedback(C_pi_fast*sys1,1);

step(T_pi,T_pi_fast);

axis([0 30 0 1.4]);

legend('PI','PI,fast');
```