

Scott Dedoncker Control & Instrumentation Ass 1 FINAL SUB

by Scott Dedoncker

FILE	12968_SCOTT_DEDONCKER_SCOTT_DEDONCKER_CONTROL_INSTRUMENTATION_ASS_1_FINAL_SUB_156167_2098160836.DOCX (18.01M)		
TIME SUBMITTED	28-FEB-2018 08:23PM (UTC+0000)	WORD COUNT	4587
SUBMISSION ID	83869791	CHARACTER COUNT	32511

HND Electrical & Electronic Engineering

PSDE 5011: Control & Instrumentation



Assignment 1

Control System Analysis, Modelling,
Simulation and Practical Testing

Scott Dedoncker

Valero Energy Electrical Apprentice

Date of Submission: 28/02/18

Word Count: 4567

Contents

Table of Figures.....	1
1. Research	3
1.1. Flow Transducers.....	3
1.2. Signalling Conditioning Methods.....	5
1.2.1. Amplification	5
1.2.2. Excitation	5
1.2.3. Filtering.....	6
1.3. Bytronic Process Rig & Control Loop	7
2. Flow Control System Modelling, Simulation and Testing.....	9
2.1. The Caldwell Model	9
2.2. Simulink Model	11
2.3. Tuning the PID	15
2.4. Practically Tuning the PID Controller.....	24
2.5. Applying the Ziegler Nichols Method.....	29
3. Conclusions & Recommendations.....	30
3.1. The Caldwell Method	30
3.2. Simulink Tuning Method	30
3.3. Practical Tuning Method	30
3.4. Ziegler Nichols Method	30
4. Citation List	31
Bibliography	31
5. Appendices	32

Table of Figures

Figure 1 - Ultrasonic Flow Sensor (Viswanath, 2016).....	3
Figure 2 - Single Jet Flow Sensor (EnggCyclopedia)	3
Figure 3 - Level Measurement Transducer (House).....	4
Figure 4 - Pressure Sensor (Gems Sensors & Controls).....	4
Figure 5 - Signal Conditioning in a Process (Webster, 2012)	5
Figure 6 - Bytronic Process Rig.....	7
Figure 7 - Ziegler Nichols Method of Calculation Table (Kurser).....	8
Figure 8 - Matlab Flow Array Formula	9
Figure 9 - Process Reaction Curve.....	9
Figure 10 - Second Process Reaction Curve	10

Figure 12 - Simulink Model	11
Figure 13 - Step Input Values	11
Figure 14 - Tuned Response 1	12
Figure 15 - PID Values	12
Figure 16 - Lookup Table Values	12
Figure 17 - Entering the Transfer Function	13
Figure 18 - Time Delay Pico Scope	13
Figure 19 - Time Delay Value	13
Figure 20 - Transport Delay.....	14
Figure 21 - Tuned P controller	15
Figure 22 - P Controller Transport Delay	15
Figure 23 - Second Tuned P Controller	16
Figure 24 - P controller Transport Delay 2	16
Figure 25 - Third Tuned P Controller.....	17
Figure 26 - P controller Transport Delay 3	17
Figure 27 - Tuned PI Controller	18
Figure 28 - PI Controller Transport Delay 1.....	18
Figure 29 - Tuned PI Controller 2	19
Figure 30 - PI Controller Transport Delay 2.....	19
Figure 31 - Tuned PI Controller 3	20
Figure 32 - PI Controller Transport Delay 3.....	20
Figure 33 - Tuned PID Controller 1.....	21
Figure 34 - PID Controller Transport Delay 1	21
Figure 35 - Tuned PID Controller 2.....	22
Figure 36 - PID Controller Transport Delay 2	22
Figure 37 - Tuned PID Controller 3.....	23
Figure 38 - PID Controller Transport Delay 3	23
Figure 39 - HMI Screen 1.....	24
Figure 40 - Response time 1.....	24
Figure 41 – Pico Scope Response 1	25
Figure 42 - HMI Screen 2.....	26
Figure 43 - Response Time 2	26
Figure 44 - Pico Scope Response 2.....	27
Figure 45 - Response Time 3	28
Figure 46 - Pico Scope Response 3.....	28

1. Research

1.1. Flow Transducers

A flow transducer is a device used in an engineering/process system to measure the flow velocity of either air or a liquid through a system. Flow transducers analyse units of flow detected to determine a flow level or velocity. The most commonly used transducers in process operation are ultrasound/ultrasonic transducers that detect flow levels by means of ultrasound. A benefit of this is that the transducer sensor is not exposed to any liquids or particles being carried by airflow in the process that could damage the sensor. Examples of processes that use transducers can be water circulation units, air ventilation systems. Figure 1 shows an example of a system using ultrasonic sensors to measure flow velocity in the system. The transducer counts the time it takes for the ultrasonic burst to reach the second sensor. (PCE Instruments UK Ltd)

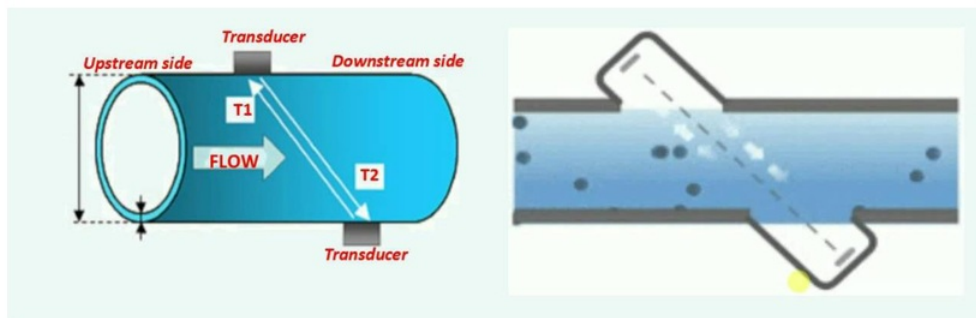


Figure 1 - Ultrasonic Flow Sensor (Viswanath, 2016)

There is also the possibility of using flow transducers with physical meters such as an impeller. For example in an air flow system as the impeller rotates the transducer will count the rotations of the impeller and convert this data into a readable format showing the air flow velocity in the system. An example of this system can be seen in figure 2.

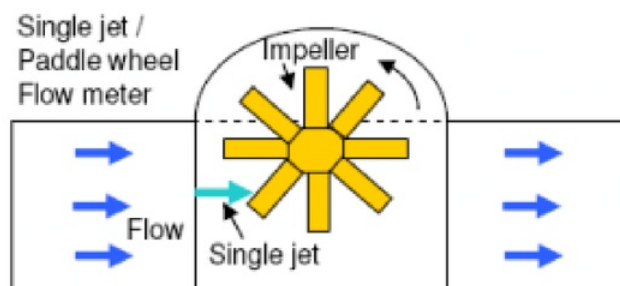
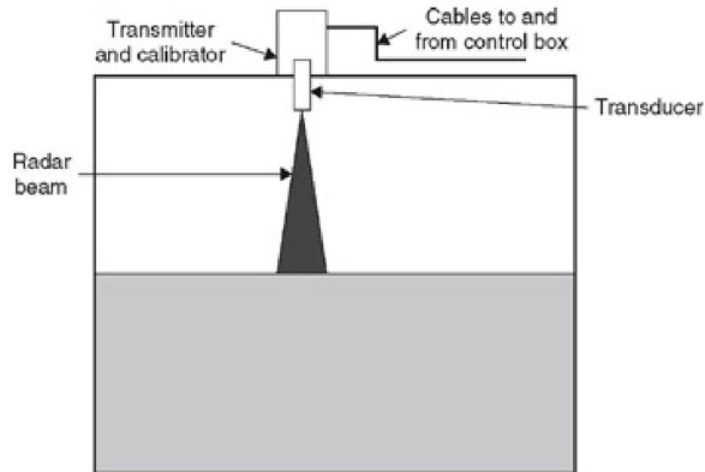


Figure 2 - Single Jet Flow Sensor (EnggCyclopedia)

While using transducers to measure flow measurement is commonplace, transducers can also be used to measure level in a process system. For example in figure 3 a transducer emits a radar signal that measures the distance between the transducer at the top of the tank, and the liquid level in the tank. This data is transferred to a control room where it is converted into a digital unit that the operator can understand.



If you read the brief carefully, only flow sensing is required in your research section.

Figure 3 - Level Measurement Transducer (House)

A fourth type of transducer is once again used in tank level and is called a pressure transducer. This transducer is placed at the bottom of a tank and as the level inside the tank increases, the transducer measures this pressure and relays it to a digital display. They can also be immersed in the liquid inside the tank where it measures the pressure. This can be seen in figure 4.

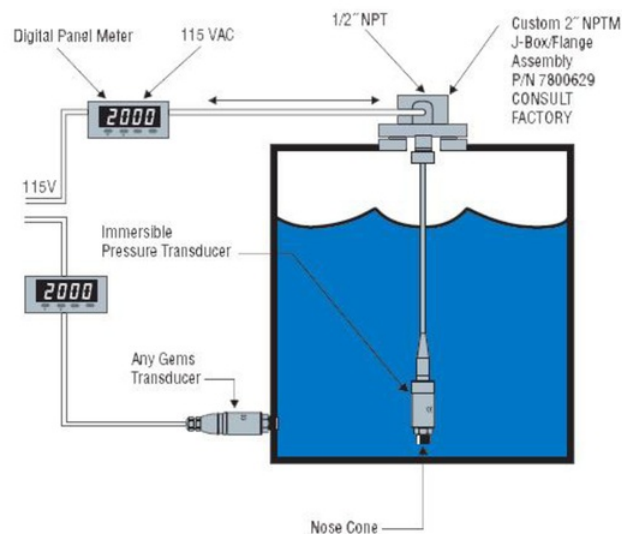


Figure 4 - Pressure Sensor (Gems Sensors & Controls)

1.2. Signalling Conditioning Methods

Signal conditioning is the process of adjusting a measured unit from a signal so that it is in a suitable state to be processed by the transmitter sending it to the display unit. This is backed up by Webster who states *"Signal conditioners are measuring system elements that start with an electric sensor output signal and then yield a signal suitable for transmission, display, or recording, or that better meet the requirements of a subsequent standard equipment or device."* (Webster, 2012) Webster also gives this example of where signal conditioning comes into a process system that can be seen in figure 5. It shows that the sensor picks up the signal, which is then conditioned before transmission to the display, alarms or controller.

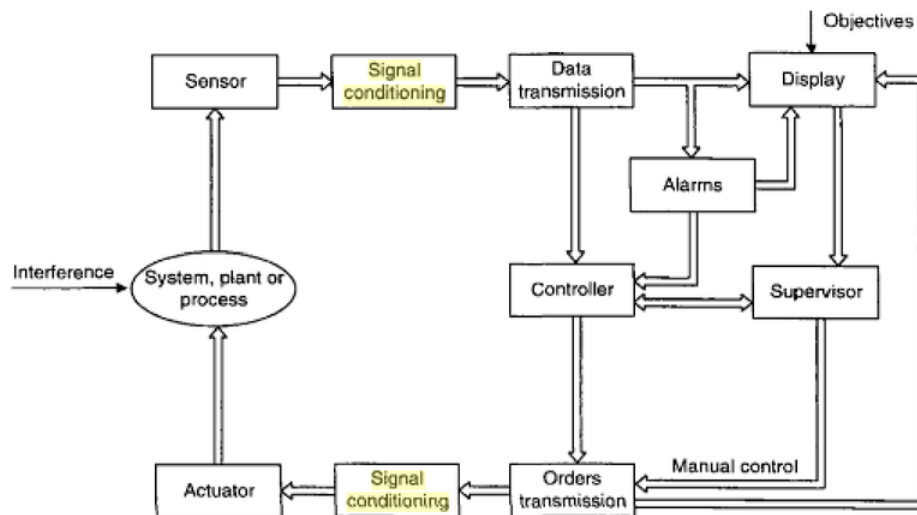


Figure 5 - Signal Conditioning in a Process (Webster, 2012)

1.2.1. Amplification

Amplification is a type of signal conditioning in which a voltage signal is amplified to a suitable level as the voltage picked up by the sensor is too low to be actively read and converted into a digital display by a DAQ. Normally devices that detect voltages usually use the range of between 0 – 10V, however if a small voltage was being given off from a device such as a thermocouple, the detected voltage would need to be amplified in order to reach the 0 – 10V range. Alternatively if the measured voltage exceeds the 10V range, a system of resistors would need to be incorporated in order to bring the voltage into range.

this should be related to flow sensing

(Windmill Software Ltd)

1.2.2. Excitation

Excitation is the manipulation of supply voltages to meet a certain RTD's requirements, for example the number of transducers used in a system or the type of transducer. Excitation can be the range requirements for voltage and current of a power supply, and getting the correct excitation conditions can have a huge effect on the performance of the system. Excitation in a system can be used to measure various elements of forces in a system such as temperature, level, brightness, vibration and pressure.

(Windmill Software Ltd)

1.2.3. Filtering

Filtering is the signalling condition process of reducing background interference via noise errors when transmitting a signal. The simplest method of resolving this problem is using a low pass filter when selecting the correct transducer for the process. This is a type of electronic filter that only allows frequencies on the lower band to pass through its pass band. It then in turn attenuates the higher frequencies in the signal. In transducers these types of low pass filters can also be referred to as an anti-aliasing filter due to it working with digitised signals of the low pass band.

(Windmill Software Ltd)

1.3. Bytronic Process Rig & Control Loop

The Bytronic Process Rig is a system designed by Bytronic Educational Technology is a bench top system used to simulate various process operations. It contains various parts such as a process module, control console and a built in power supply. This rig can be connected to a computer where various PLC programs can be downloaded to simulate and tune the system in order to get the best conditions of flow, temperature, level and pressure in a system. The system typically processes water through various steps in a cycle. The level of the system uses a 0-10v magnetostrictive sensor; pressure is measured using a 0-5bar-gage sensor and flow is measured using a turbine flow sensor.



Figure 6 - Bytronic Process Rig

An F to V converter chip is used.

A control loop needed to be researched in order to be used for the process rig system to determine flow using the systems turbine flow meter. Turbine flow meters operate by detecting pulse signals from the rotor to determine how many revolutions there have been. Depending on the pulses per minute the plc will be able to determine how many revolutions there is per minute, and in turn the flow rate per minute. The value calculated by the PLC will be given to the panel man (operator of the process) as an analogue value. As the system is set to pump a maximum of 2 litres a minute, the process will use the turbine to determine how much powerful the flow of water is, and give an analogue value of between 0-5v to give the operator a readable digital value. This 0-5v range can also be interpreted as 4-20ma ranges.

1.4 The Ziegler Nichols Method

The Ziegler Nichols method is a method used to tune a PID controller. It is the process of providing three workable values for:

- K_p – Path Gain
- K_i – Integrator Time Constant
- K_d – Derivative Time Constant

The process of tuning using the Ziegler Nichol method is quite simple. Initially the K_i and K_d terms of the PID controller are turned off leaving only K_p . K_p is then slowly increased from 0 until the output of the system demonstrates sustainable oscillations. This is where critical gain is achieved. Then individually values for K_p and K_i should be calculated, then K_p , K_i and K_d after using the following table shown in figure 7. As long as K_p gives an output of a stable oscillation, this process should work.

This is not the method we used!

Ziegler Nichols Method			
Control type	K_p	$\tau_N (=tau_N)$	$\tau_V (=tau_V)$
P	$0.5 * K_{p,crit}$	-	-
PI	$0.45 * K_{p,crit}$	$0.85 * T_{crit}$	-
PID	$0.6 * K_{p,crit}$	$0.5 * T_{crit}$	$0.13 * T_{crit}$

Figure 7 - Ziegler Nichols Method of Calculation Table (Kurser)

While the Ziegler Nichols method of tuning is a widely used tool, it is not consistent and will not always give the best results the user desires.

(Graf, 2013)

2. Flow Control System Modelling, Simulation and Testing

2.1. The Caldwell Model

For the first part of the practical of this assignment the Bytronic process rig needed to be in an open loop configuration and analysed so that a mathematical model could be constructed. This was done using MATLAB. The flow log can be seen in the annex of this assignment. It was loaded into a MATLAB formula and simulated to give a graph.

```
Assignment_1.m  X  +
1 - hold off
2 - %load data from file into array
3 - load flow_log_2.txt;
4 - flow_array = flow_log_2(:,1);
5 - plot(flow_array);
6 - grid on
```

Figure 8 - Matlab Flow Array Formula

For the Caldwell method the value of t_y needed to be derived from the simulated process reaction curve. This can be found at 73% of final value, which on the graph would be at along the Y axis. From the process reaction curve t_y was derived as being 112.6. This can be viewed in figure 9.

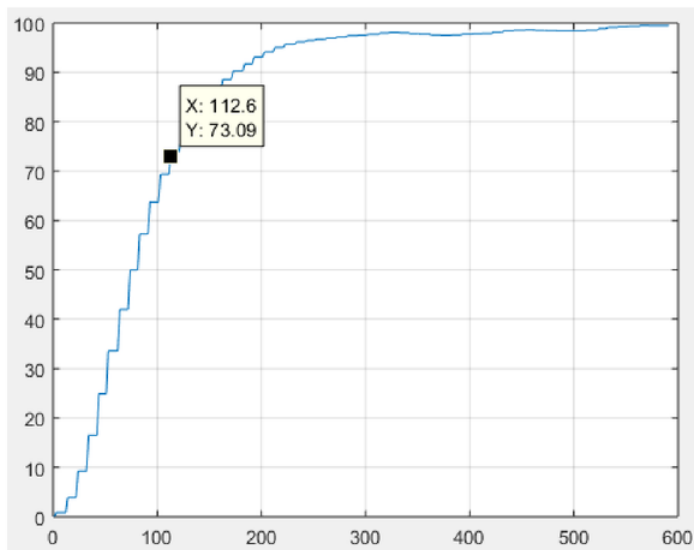


Figure 9 - Process Reaction Curve

The Caldwell calculations were then applied using this value. This can be seen below:

$$T1+T2 = t_y/1.32 = 112.6/1.32 = 85$$

$$T1+T2 = 85\text{ms} = 0.085\text{s}$$

$$T1+T2/2 = 42.5\text{ms} \quad \text{Use brackets please: } (T1 + T2)/2$$

The new value of 42.5 was then selected on the process reaction curve referencing the X axis this time and it gave a new value of 18.83. This value needed to be divided by 100 resulting in the final value of $X = 0.1883$. This can be seen in figure 10.

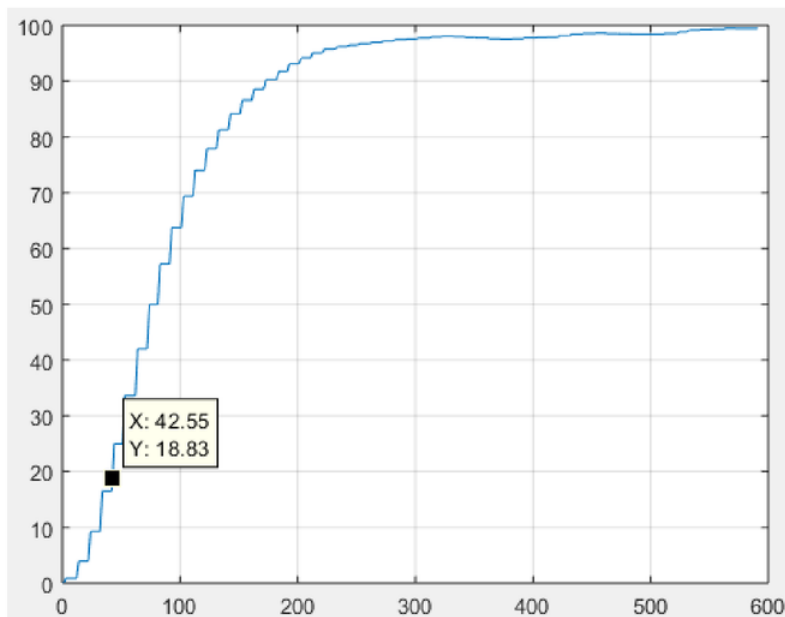


Figure 10 - Second Process Reaction Curve

As the Caldwell method levels out at 0.26, the best value we can have for $T_2/(T_1+T_2)$ is 0.5. *Use brackets please*

So taking the values of $T_1+T_2 = 0.085\text{ms}$, the following calculations were carried out:

$$T_1T_2 = 0.0425^2 = 1.80625 \times 10^{-3} = 0.00180625$$

The transfer function is $1/(T_1T_2s^2+(T_1+T_2)s+1)$, so the values can be added in to create a transfer function for Matlab.

Num1 = [1]

Den1 = [0.00180625 + 0.085 + 1] *There are no plus signs in these*

```
hold off
%load data from file into array
load flow_log_2.txt;
flow_array = flow_log_2(:,1);
plot(flow_array);%blue line
hold on
grid on
%check time column to get sample time
t=(0:0.001:length(flow_array)*0.001);
num1=[1];
den1=[0.00180625 0.085 1];
model_1 = step(98*num1, den1, t);
plot(model_1);
```

Figure 11 - Matlab Transfer Function

W
h
e
e
i
s
t
h
e
C
l
d
w
e
ll
c
u
r
v
e
t
o
e
x
p
l
a
i

*This is
correct*

2.2. Simulink Model

The next step for this assignment was using the block design to create a Simulink model using the Matlab transfer function that had been previously derived. The transfer function was given the s^2 & s variables. The step input for this system was given the final value of 50. This can be seen in figures 12 and 13.

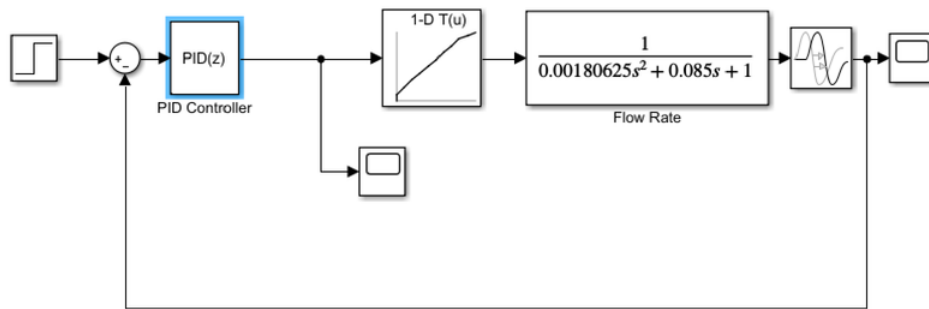


Figure 12 - Simulink Model

Step	
Output a step.	
Parameters	
Step time:	1
Initial value:	0
Final value:	50
Sample time:	0

Figure 13 - Step Input Values

The PID value in the Simulink model was then tuned and the block graph updated to the block response shown in figure 14. This shows the tuned oscillation response over 0.45 seconds.

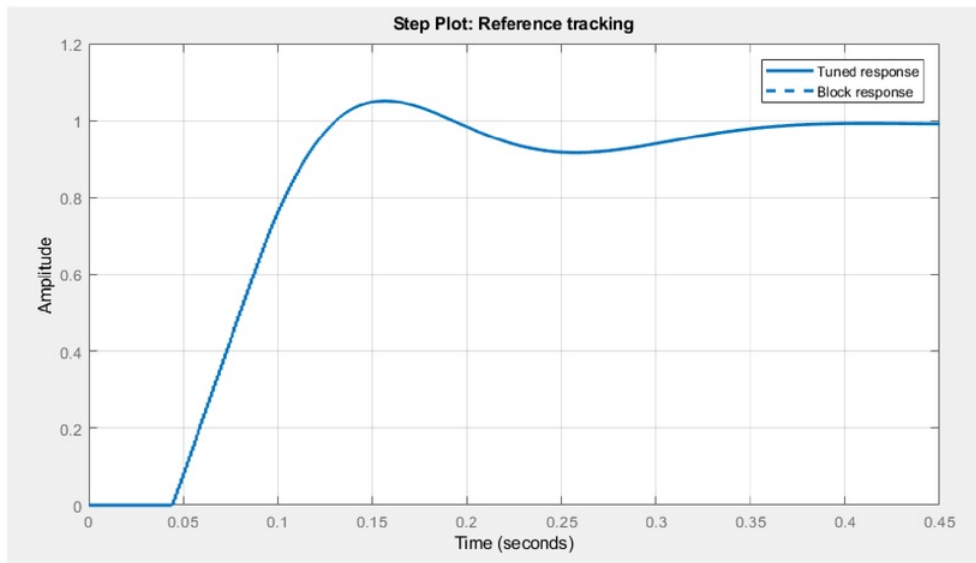


Figure 14 - Tuned Response 1

The tuned PID gave the following values:

Controller parameters

Source: ☐ [Compensator formula](#)

Proportional (P):

Integral (I):

Derivative (D):

☒ Use filtered derivative

Filter coefficient (N):

Select Tuning Method:

$$P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

Figure 15 - PID Values

The lookup table had the following values entered into.

This belongs to the explanation of the model

Number of table dimensions:

Data specification:

Table data:

Breakpoints specification:

Breakpoints 1:

Figure 16 - Lookup Table Values

The transfer function was then entered into the flow rate block of the model. This was entered in the similar format to the MATLAB formula. This can be seen in figure 17.

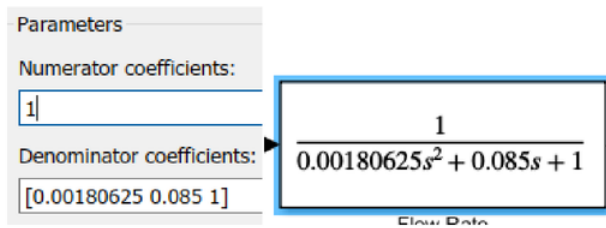


Figure 17 - Entering the Transfer Function

The time delay was measured using the process rig and the software Pico Scope 6. The time delay value was found to be 43.29ms. The graph is shown in figure 18, and the value derived is shown in figure 19.

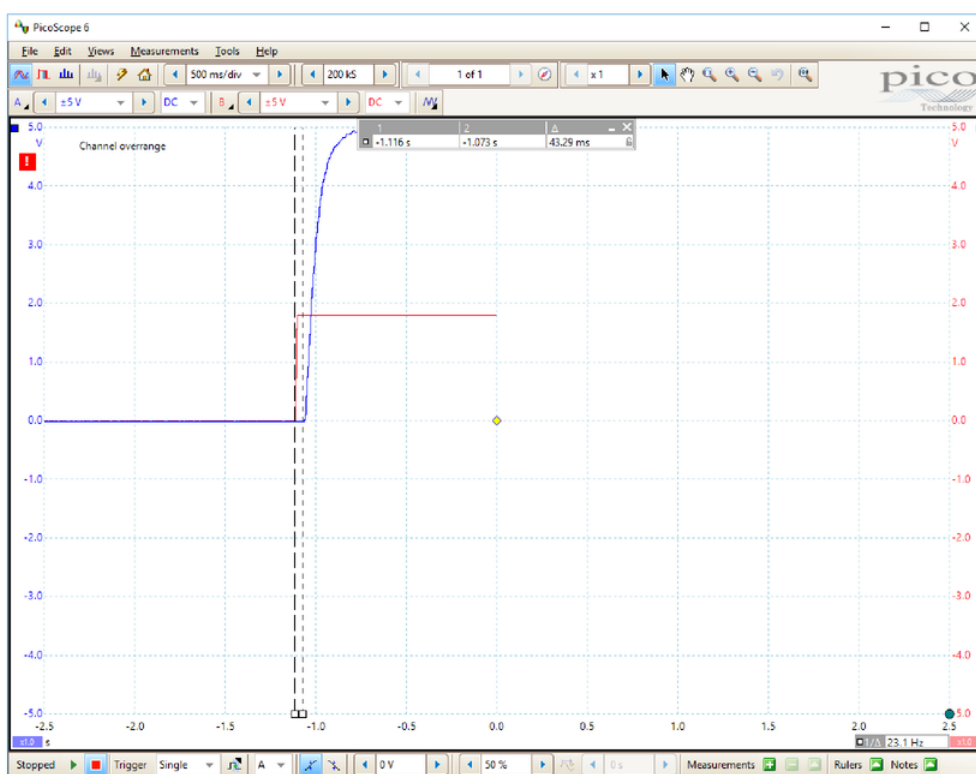


Figure 18 - Time Delay Pico Scope

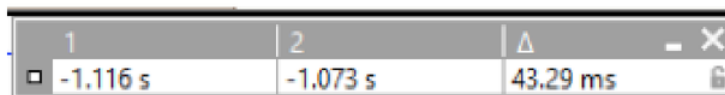


Figure 19 - Time Delay Value

The time delay is used to determine the systems delay response time. The value was entered into the transport delay as seen in figure 20.

Parameters	
Time delay:	0.04329
Initial output:	0
Initial buffer size:	1024

Figure 20 - Transport Delay

The final component in the diagram was the scope.

2.3. Tuning the PID

Now that the Simulink Model had been built and all values entered correctly, the PID controller needed to be tuned in order to give the best possible response for the system. This was done in three steps similar to the Ziegler Nichols method where the P controller value was tuned first, then the values for PI, then values for PID.

The P controller was first tuned and gave the proportional value of 0.158453818206392. This can be seen in figure 21.

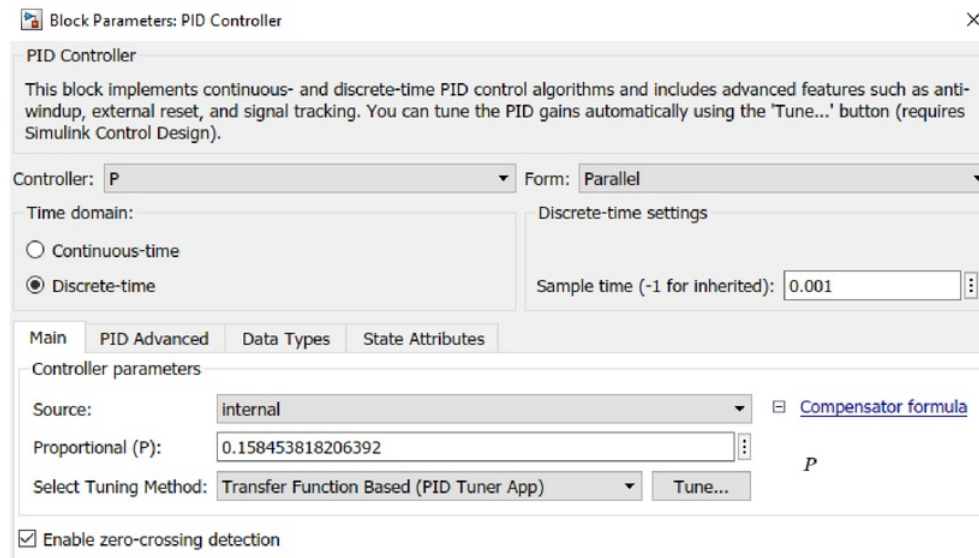


Figure 21 - Tuned P controller

It gave the transport delay response seen in figure 22. This shows a very stable response to the value as it rises quickly but does not oscillate at all. This needed to be tweaked further.

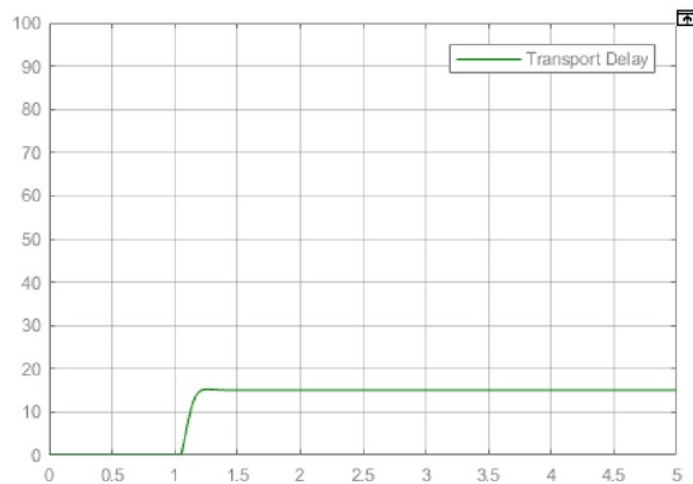


Figure 22 - P Controller Transport Delay

The second tuned response of the P controller had the P value raised to 0.958453818206392. This can be seen in figure 23.

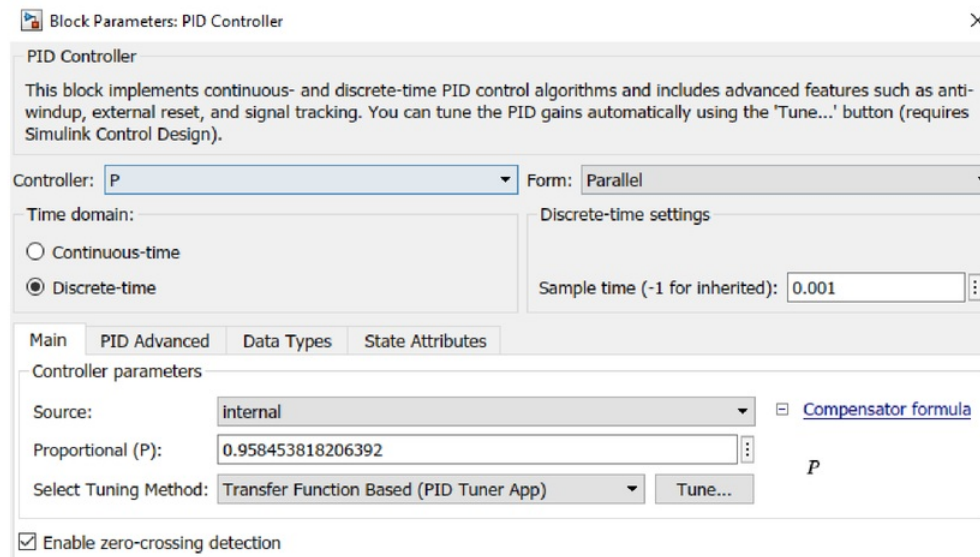


Figure 23 - Second Tuned P Controller

The tuned response for this value gave a transport delay that climbed higher and quicker, and also slightly oscillated before settling out at about 30. This still needed to be tuned further. This can be seen in figure 24.

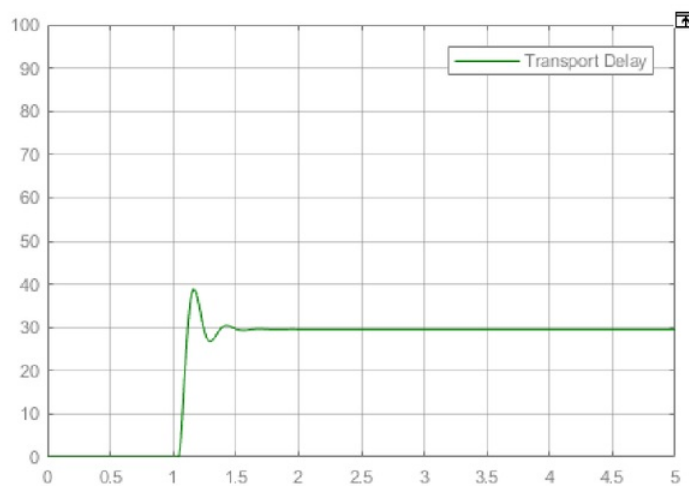


Figure 24 - P controller Transport Delay 2

The third tuned response for the P controller used a value of 1.558453818206392 which provided a good amount of oscillation. It levelled out at 35. This can be seen in figures 25 and 26 respectively.

Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **P** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): **0.001**

Main PID Advanced Data Types State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **1.558453818206392** **P**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

☒ Enable zero-crossing detection

Figure 25 - Third Tuned P Controller

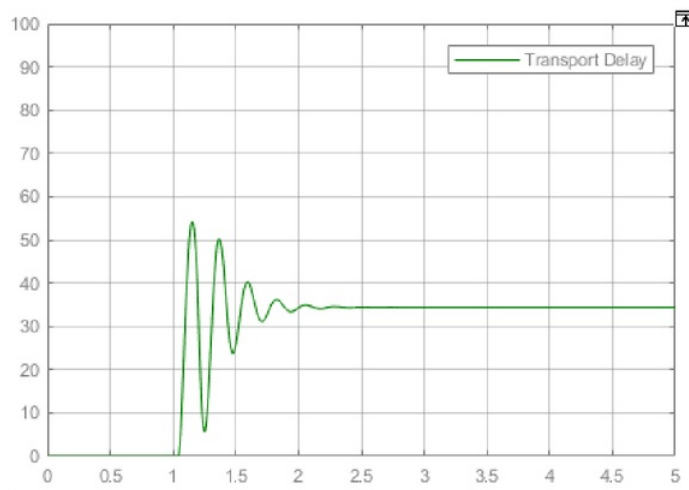


Figure 26 - P controller Transport Delay 3

Now that the P controller had been tuned, the I value needed to be activated and tuned as well as retuning the P value so that both could work together for the system. The Simulink model gave the two initial tuned values for proportional and integral shown in figure 27.

Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Integrator method: **Forward Euler**

Sample time (-1 for inherited): **0.001**

Main PID Advanced Data Types State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **0.0091484754827173**

Integral (I): **0.090177439996481**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

$$P + I \cdot T_s \frac{1}{z-1}$$

Initial conditions

Source: **internal**

Integrator: **0**

External reset: **none**

Figure 27 - Tuned PI Controller

It gave the initial Transport Delay response shown below in figure 28. This shows a very slow and straight rise which needed to be tweaked further.

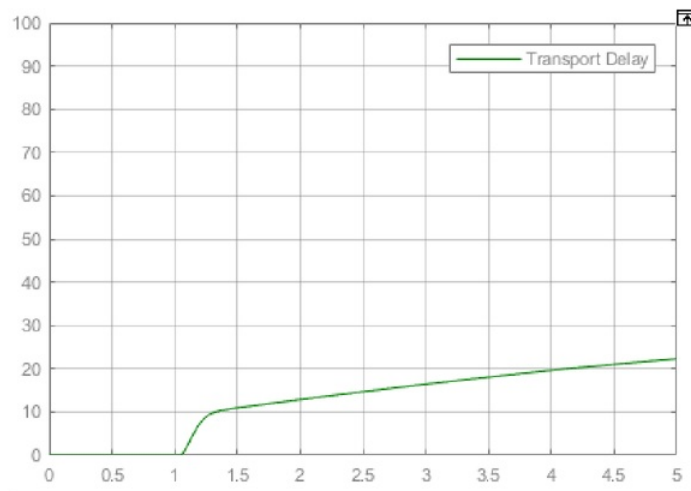


Figure 28 - PI Controller Transport Delay 1

The values were tweaked to for the second tuning of the PI controller. These new values for proportional and integral can be seen in figure 29 below.

Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Integrator method: **Forward Euler**

Sample time (-1 for inherited): **0.001**

Main PID Advanced Data Types State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **0.0414647253615273**

Integral (I): **0.50176224501055**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

Initial conditions

Source: **internal**

Integrator: **0**

$$P + I \cdot T_s \frac{1}{z-1}$$

Figure 29 - Tuned PI Controller 2

It gave a similar response to the first one, however this response reached a higher value of 45 and gave a much nicer curve when rising to the 45 value.

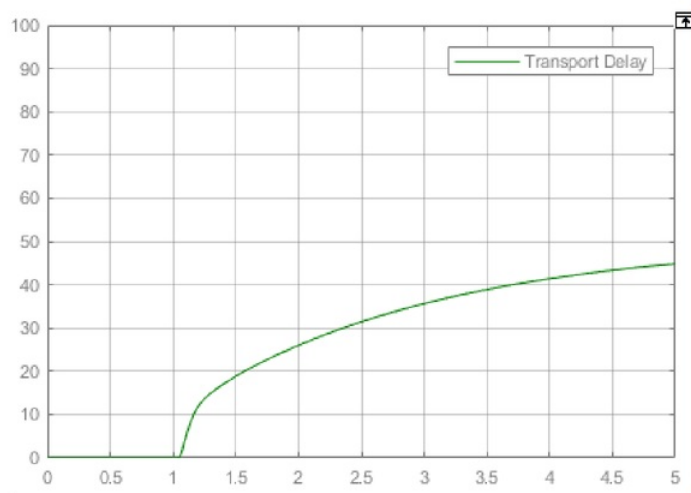


Figure 30 - PI Controller Transport Delay 2

The values were tweaked a third time to give the values for proportional and integral shown in figure 31.

Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Integrator method: **Forward Euler**

Sample time (-1 for inherited): **0.001**

Main PID Advanced Data Types State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **0.0914847548271738**

Integral (I): **0.901774399964819**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

$$P + I \cdot T_s \frac{1}{z - 1}$$

Figure 31 - Tuned PI Controller 3

This gave a much nicer curved response for the transport delay and gave a final raised value of just below 50 after 5 seconds. This can be seen in figure 32.

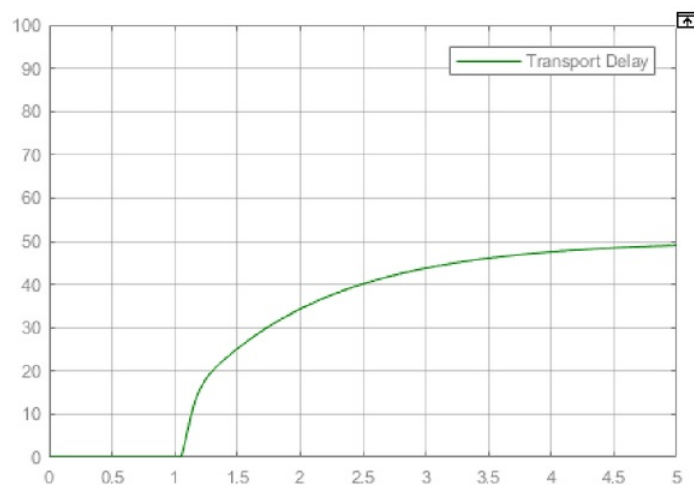


Figure 32 - PI Controller Transport Delay 3

Now the controller needed to have its derivative value activated and all three values of proportional, integral and derivative needed to be retuned for a final time. Figure 33 shows the tuned values for the PID controller.

Source:	internal
Proportional (P):	0.124773826522681
Integral (I):	1.10440725028668
Derivative (D):	0.002409553843206

Figure 33 - Tuned PID Controller 1

These values gave the response for the transport delay shown in figure 34. It shows a gentle curve rising to the value of 50.

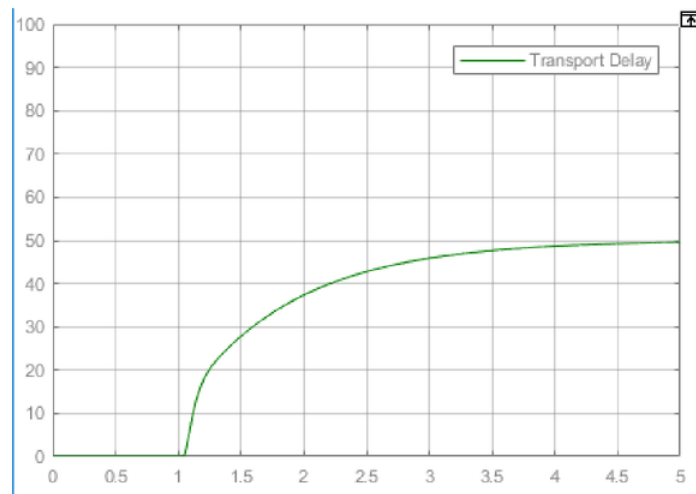


Figure 34 - PID Controller Transport Delay 1

These values were adjusted to provide a better response. Figure 35 shows the newly adjusted values for proportional, integral and derivative.

Source:	internal
Proportional (P):	0.12
Integral (I):	1.9
Derivative (D):	0.00187

Figure 35 - Tuned PID Controller 2

This gave a better curved response for the transport delay and still gave the final value of 50 show in figure 36.

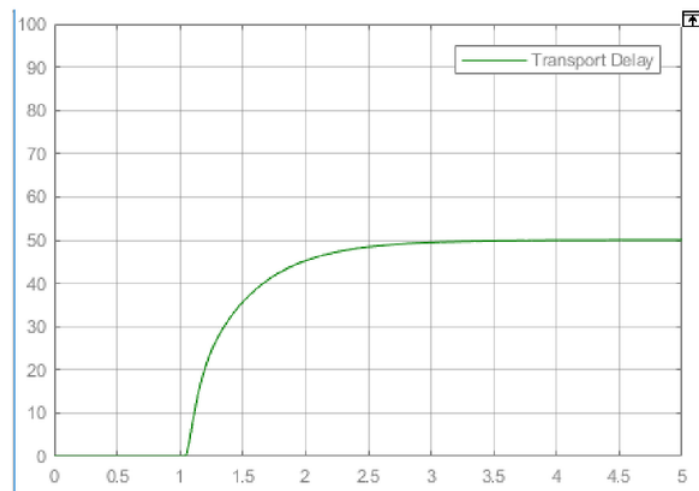


Figure 36 - PID Controller Transport Delay 2

The values for the PID were adjusted further and can be seen in figure 37.

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Parallel**

Time domain:

☐ Continuous-time

☒ Discrete-time

Discrete-time settings

Integrator method: **Forward Euler**

Filter method: **Forward Euler**

Sample time (-1 for inherited): **0.001**

Main PID Advanced Data Types State Attributes

Controller parameters

Source: **internal** [Compensator formula](#)

Proportional (P): **0.2**

Integral (I): **3.5**

Derivative (D): **0.01**

☒ Use filtered derivative

Filter coefficient (N): **875.454423089363**

Select Tuning Method: **Transfer Function Based (PID Tuner App)** **Tune...**

$$P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

Figure 37 - Tuned PID Controller 3

The final response given was good as it had a very quick rise and settled at 50 after approximately 1.7 seconds. This can be seen in figure 38.

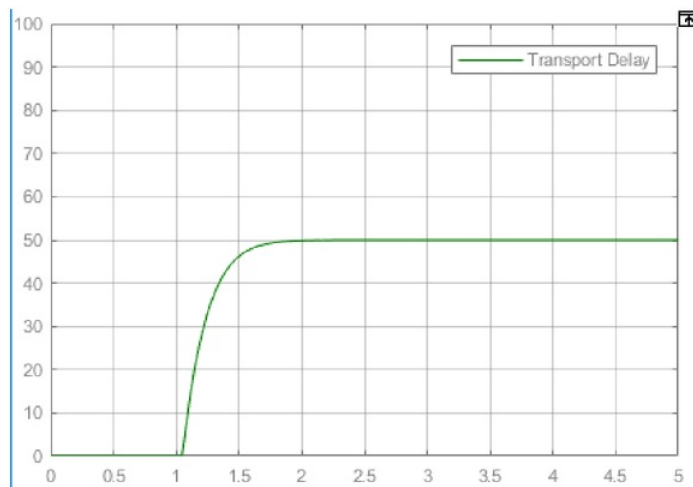


Figure 38 - PID Controller Transport Delay 3

2.4. Practically Tuning the PID Controller

The values tuned by the Simulink model for the PID controller could now be used in practise to tune real world values using the Pico Scope 6 software and the PLC flow controller with the Bytronic Rig.. This was done by using a set point of 1 litres/minute for the PID mode and the response time was measured from the start point, to its settling point. The values were then tweaked in order to try and get a better response time from the real life model. The tweaked values for the PID controller were entered on the HMI screen.

The first test used the following values for PID:

$P = 0.2$

$I = 3.5$

$D = 0.01$

These were entered on the HMI screen as shown in figure 39.

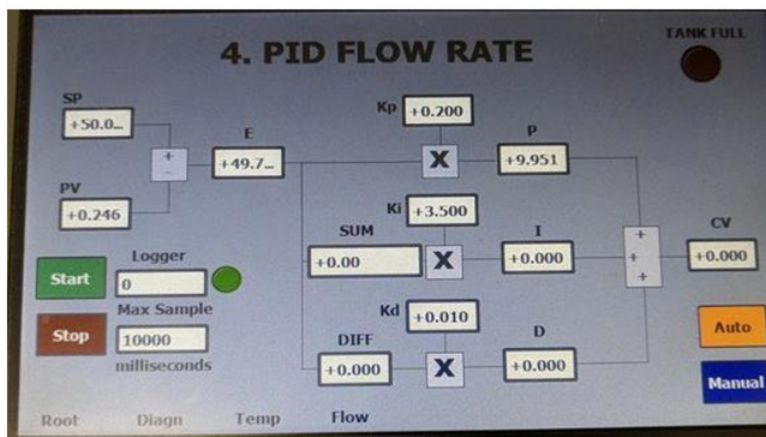


Figure 39 - HMI Screen 1

The process rig was then started and the Pico Scope program recorded the response time. It gave a response time of 532ms. This can be seen in figure 40 and the Pico Scope response can be seen in figure 41.

1	2	Δ	- X
-2.083 s	-1.551 s	532.0 ms	
2.411 V	2.623 V	212.1 mV	

Figure 40 - Response time 1

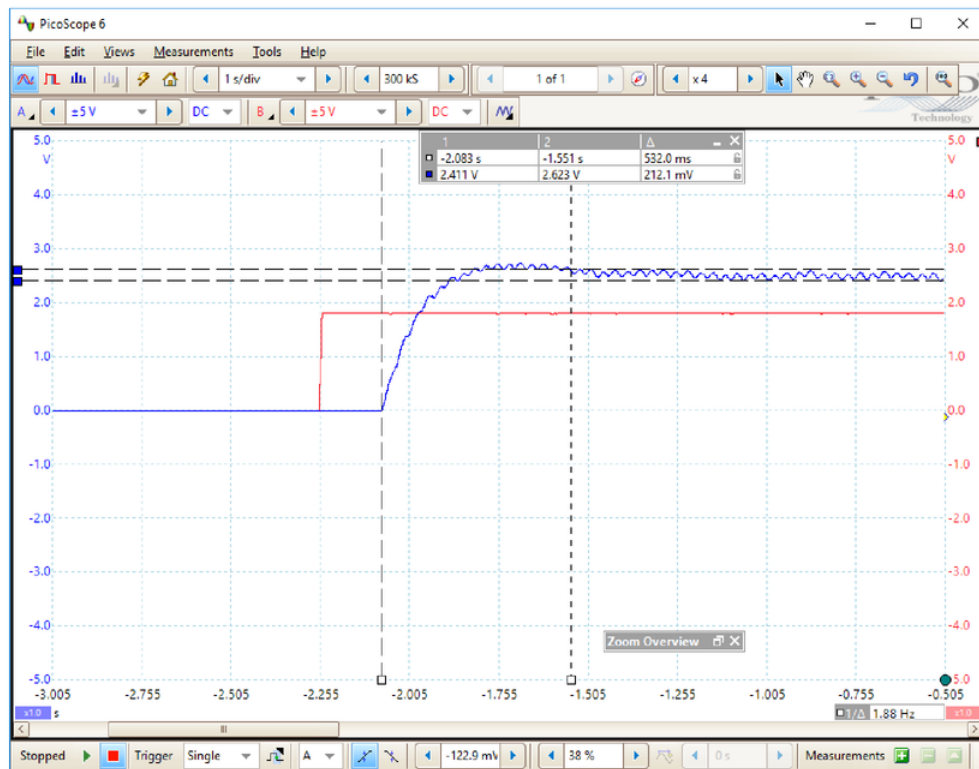


Figure 41 – Pico Scope Response 1

For the second test the values for PID were tweaked slightly to as seen below:

$P = 0.3$

$I = 3.8$

$D = 0.02$

These were then entered onto the HMI screen as shown in figure 42

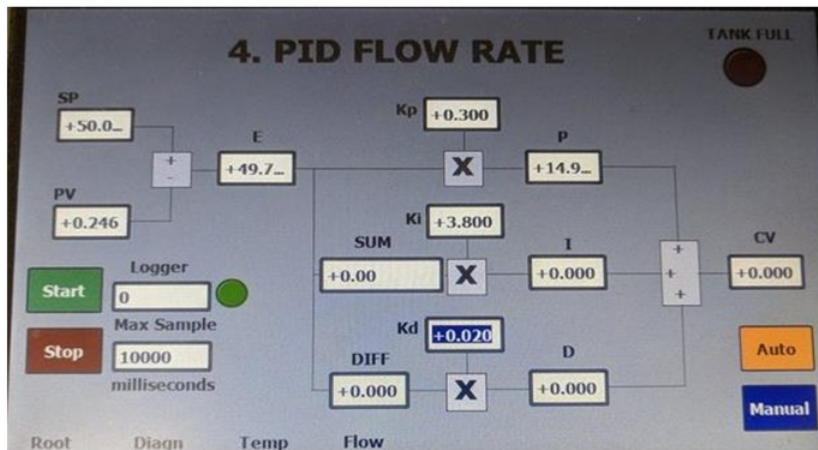


Figure 42 - HMI Screen 2

This gave a better and more improved response time of 312.1ms as shown in figure 43. The Pico Scope response can be seen in figure 44.

1	2	Δ	
☐ -2.309 s	-1.997 s	312.1 ms	
☒ 2.411 V	2.623 V	212.1 mV	

Figure 43 - Response Time 2

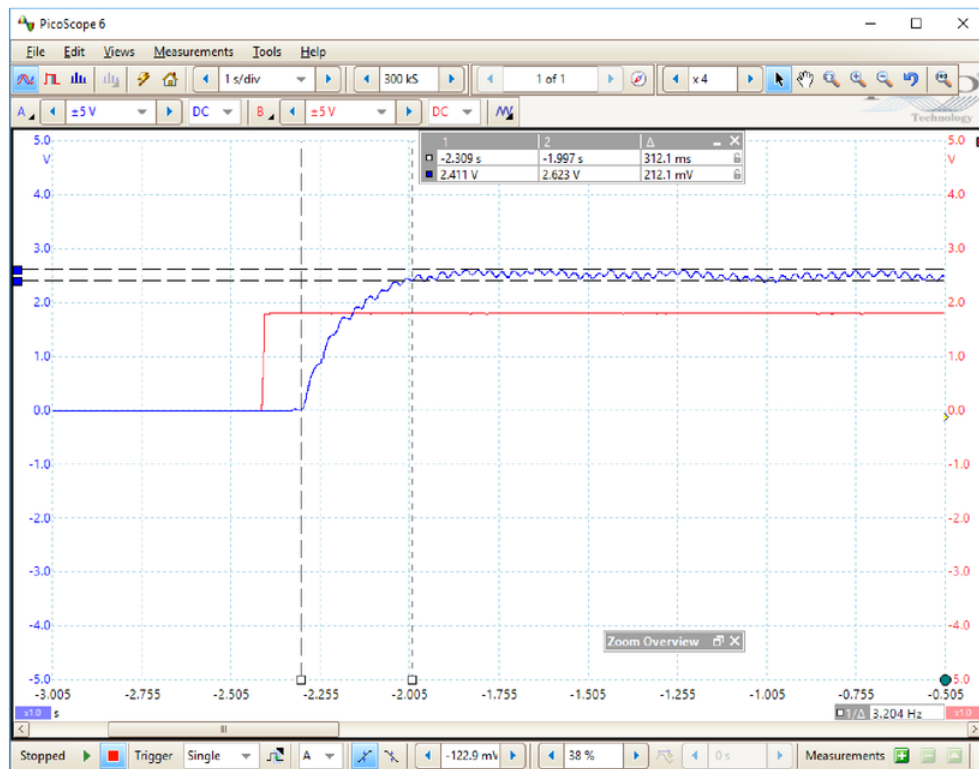


Figure 44 - Pico Scope Response 2

The third test had new adjusted values again in order to improve the response time but this did not occur. The new values can be seen below:

$P = 0.35$

$I = 4$

$D = 0.06$

These values were then entered onto the HMI screen and the Rig was activated once again. The response time given by these values resulted in 603.4ms. This was worse than the previous result and therefore the values were not as good as the PID controller values used in the second test.

1	2	Δ	-	X
☐ -2.255 s	-1.651 s	603.4 ms		
☒ 2.411 V	2.623 V	212.1 mV		

Figure 45 - Response Time 3

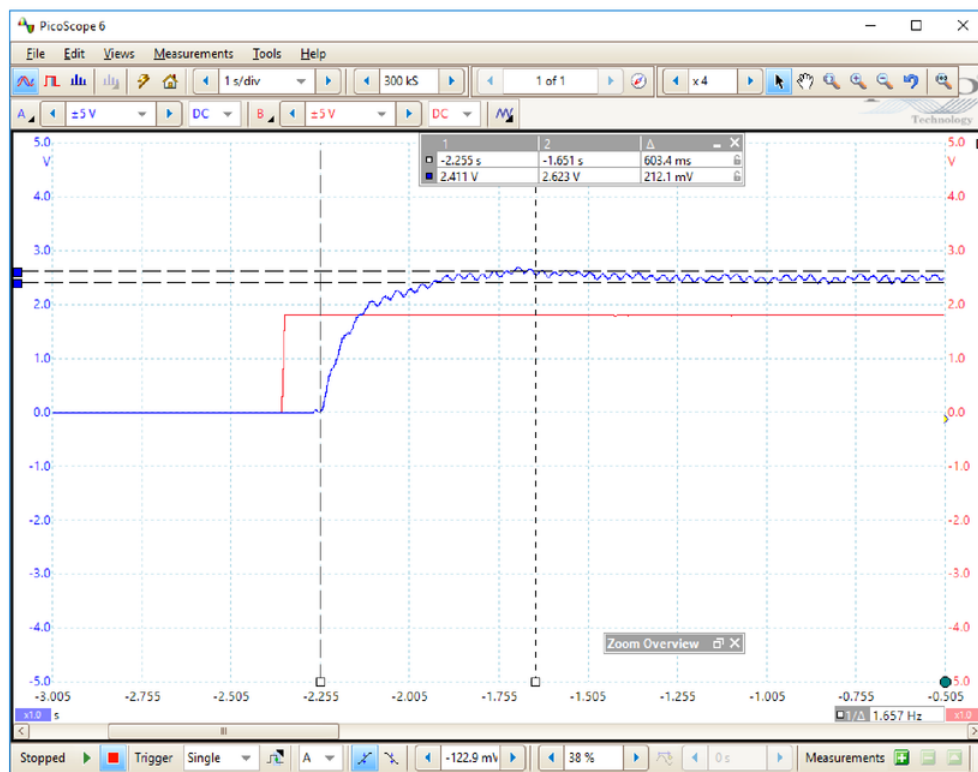


Figure 46 - Pico Scope Response 3

2.5. Applying the Ziegler Nichols Method

The final part of this assignment was to apply the Ziegler Nichols method to the PID controller. For this the open loop process reaction curve from the system was analysed and gave the following values for T, L and R:

$$T = 100\text{ms}$$

$$L = 45\text{ms}$$

$$R = 100/T = 100/0.1 = 1000$$

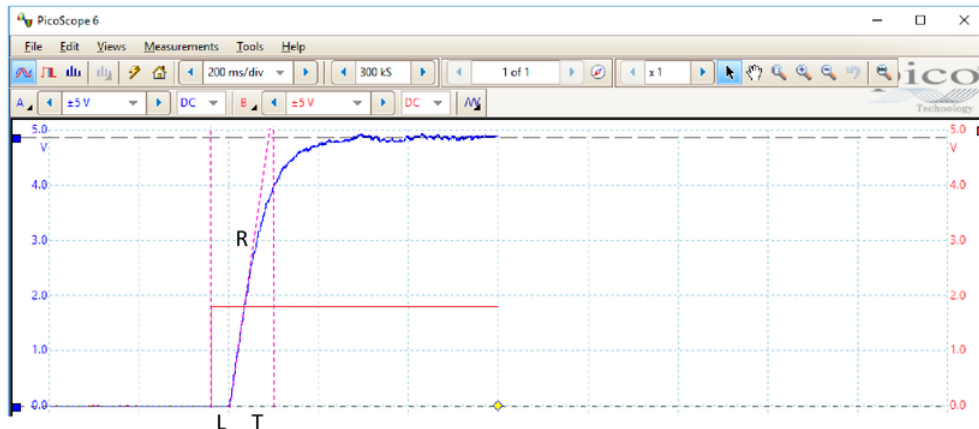


Figure 47 - Process Reaction Curve

The Ziegler Nichols calculations were performed using these values:

CONTROL MODE	K_p	K_i	K_d
P	$\Delta c / RL$		
P I	$0.9 \Delta c / RL$	$K_p / 3.33 L$	
P I D	$1.2 \Delta c / RL$	$K_p / 2 L$	$0.5 K_p L$

P controller - $K_P = 100 / (1000 \times 0.045) = 2.2$

PI controller – $K_P = 0.9 \times 2.2 = 1.98$, $K_I = 1.98 / (3.33 \times 0.045) = 13.2$

PID controller – $K_P = 1.2 \times 2.2 = 2.64$, $K_I = 2.64 / (2 \times 0.045) = 29.3$, $K_D = 0.5 \times 2.64 \times 0.045 = 0.0594$

The system lag means these values are not accurate values, and that the model is more accurate than the Ziegler Nichols effect.

3. Conclusions & Recommendations

3.1. The Caldwell Method

The first part of the practical experiment was producing a transfer function using the Caldwell method. This method was very easy to use but gave results that were nowhere near what was ???????? needed for the final practical tuning of the PID controller. It gave a transfer function that was very long and specific, and the final values used just in the Simulink tuning of the PID, were nowhere near??????? the same values that were initially inputted from the Caldwell Method.

3.2. Simulink Tuning Method

Using the Simulink Tuning method was a long process to get the final values. The final values tuned were:

$$P = 0.2$$

$$I = 3.5$$

$$D = 0.01$$

These values were good as when put into practise on the practical PLC with the Bytronic Process Rig they gave a response time of 532ms, only 200ms slower than the best values tuned in this experiment. This shows that the Simulink Tuning method is effective to an extent in tuning a PID controller without any practical equipment and using just software.

3.3. Practical Tuning Method

The practical tuning method for the PID controller was an interesting experiment. Values were able to be tweaked for P, I and D to get a final response time of 312ms, a very quick time. However through the third test it became apparent that increasing each value for P, I and D only by a small amount, gave a far worse response time of 603ms, worse than the Simulink tuned values. This shows that it is possible in practise to get possibly worse values by altering values using the slightest of margins.

3.4. Ziegler Nichols Method

The Ziegler Nichols method was used in order to determine if the traditional method of deriving values for KP, KI and KD would derive similar values to the final values used to get the best response time from the process rig. Comparing the values below shows that the Ziegler Nichols method of deriving values was nowhere near as accurate as the practical method.

Ziegler Nichols KP = 2.64 Simulated = 0.3

Ziegler Nichols KI = 29.3 Simulated = 3.8

Ziegler Nichols KD = 0.0594 Simulated = 0.02

*Thoroughly
shown
that the
Ziegler
Nichols
method
is not
as
accurate
as
the
practical
method.*

4. Citation List

Bibliography

- Caldwell, J. (1999). *Mathematical Modelling*. Springer Netherlands.
- EnggCyclopedia. (n.d.). *Mechanical flow measurement devices*. From EnggCyclopedia: <http://www.enggcyclopedia.com/2011/07/mechanical-flow-measurement-devices/>
- Gems Sensors & Controls. (n.d.). *Pressure Transducer Operating Principle*. From Gems: Sensors & Controls: <https://www.gemssensors.com/customersupport/literature-pdfs/operating-principle-installation-and-maintenance/cvd-pressure-transducer-operating-principle>
- Graf, J. (2013). *PID Control: Ziegler-Nichols Tuning*. CreateSpace Independent Publishing Platform.
- House, D. J. (n.d.). *Tank Measurement Radar System*. From Engineering 360: <https://www.globalspec.com/reference/22655/203279/tank-measurement-radar-system>
- Kurser. (n.d.). *Ziegler Nichols Method*. From Kurser.iha.dk: http://kurser.iha.dk/m/mtpri1/control/3_PID/f_ziegler/ziegler_nichols_method.html
- PCE Instruments UK Ltd. (n.d.). *Flow Transducers*. From <http://www.industrial-needs.com/systems/flow-transducers.htm>
- Viswanath, B. (2016, 07 15). *Measuring or Working principle of Ultrasonic flow meter*. From <https://i.ytimg.com/vi/Mkjlw-iX6Af4/maxresdefault.jpg>
- Webster, J. G. (2012). *Sensors and Signal Conditioning*. John Wiley & Sons.
- Windmill Software Ltd. (n.d.). *Signal Conditioning*. Retrieved 01 29, 2018 from Windmill Software: <http://www.windmill.co.uk/signal-conditioning.html>
- Xhale. (2017, 03 17). *Ultrasonic Flow Meter*. From <https://forum.arduino.cc/index.php?topic=463316.0>

5. Appendices

Flow Log:

0.0 0.4311594 0.8623188 0.8623188 0.8623188 0.8623188 0.8623188 0.8623188 0.8623188
0.8623188 0.8623188 0.8623188 2.376812 3.891304 3.891304 3.891304 3.891304 3.891304
3.891304 3.891304 3.891304 3.891304 6.565217 9.23913 9.23913 9.23913 9.23913 9.23913
9.23913 9.23913 9.23913 9.23913 12.86594 16.49275 16.49275 16.49275 16.49275 16.49275
16.49275 16.49275 16.49275 16.49275 20.71377 24.93478 24.93478 24.93478 24.93478 24.93478
24.93478 24.93478 24.93478 29.27536 33.61594 33.61594 33.61594 33.61594 33.61594 33.61594
33.61594 33.61594 33.61594 33.61594 37.82246 42.02898 42.02898 42.02898 42.02898 42.02898
42.02898 42.02898 42.02898 42.02898 45.98913 49.94928 49.94928 49.94928 49.94928 49.94928
49.94928 49.94928 49.94928 53.59783 57.24638 57.24638 57.24638 57.24638 57.24638 57.24638
57.24638 57.24638 57.24638 60.49275 63.73913 63.73913 63.73913 63.73913 63.73913 63.73913
63.73913 63.73913 63.73913 66.53986 69.34058 69.34058 69.34058 69.34058 69.34058 69.34058
69.34058 69.34058 69.34058 71.66666 73.99275 73.99275 73.99275 73.99275 73.99275 73.99275
73.99275 73.99275 73.99275 75.96014 77.92754 77.92754 77.92754 77.92754 77.92754 77.92754
77.92754 77.92754 77.92754 79.59058 81.25362 81.25362 81.25362 81.25362 81.25362 81.25362
81.25362 81.25362 81.25362 82.7029 84.15218 84.15218 84.15218 84.15218 84.15218 84.15218
84.15218 84.15218 84.15218 85.36957 86.58696 86.58696 86.58696 86.58696 86.58696 86.58696
86.58696 86.58696 86.58696 87.56884 88.55073 88.55073 88.55073 88.55073 88.55073 88.55073
88.55073 88.55073 88.55073 89.41304 90.27536 90.27536 90.27536 90.27536 90.27536 90.27536
90.27536 90.27536 90.27536 91.01811 91.76087 91.76087 91.76087 91.76087 91.76087 91.76087
91.76087 91.76087 91.76087 92.43841 93.11594 93.11594 93.11594 93.11594 93.11594 93.11594
93.11594 93.11594 93.11594 93.6413 94.16666 94.16666 94.16666 94.16666 94.16666 94.16666
94.16666 94.16666 94.16666 94.61232 95.05797 95.05797 95.05797 95.05797 95.05797 95.05797
95.05797 95.05797 95.05797 95.4058 95.75362 95.75362 95.75362 95.75362 95.75362 95.75362
95.75362 95.75362 95.75362 95.98913 96.22464 96.22464 96.22464 96.22464 96.22464 96.22464
96.22464 96.22464 96.22464 96.3442 96.46377 96.46377 96.46377 96.46377 96.46377 96.46377
96.46377 96.46377 96.46377 96.58334 96.7029 96.7029 96.7029 96.7029 96.7029 96.7029
96.7029 96.7029 96.7029 96.82609 96.94927 96.94927 96.94927 96.94927 96.94927 96.94927
96.94927 96.94927 96.94927 97.07246 97.19566 97.19566 97.19566 97.19566 97.19566 97.19566
97.19566 97.19566 97.19566 97.31884 97.44203 97.44203 97.44203 97.44203 97.44203 97.44203
97.44203 97.44203 97.44203 97.50362 97.56522 97.56522 97.56522 97.56522 97.56522 97.56522
97.56522 97.56522 97.56522 97.6558 97.74638 97.74638 97.74638 97.74638 97.74638 97.74638
97.74638 97.74638 97.74638 97.85507 97.96377 97.96377 97.96377 97.96377 97.96377 97.96377
97.96377 97.96377 97.96377 98.01811 98.07246 98.07246 98.07246 98.07246 98.07246 98.07246
98.07246 98.07246 98.07246 98.03986 98.00725 98.00725 98.00725 98.00725 98.00725 98.00725
98.00725 98.00725 98.00725 97.92754 97.84782 97.84782 97.84782 97.84782 97.84782 97.84782
97.84782 97.84782 97.84782 97.80797 97.76811 97.76811 97.76811 97.76811 97.76811 97.76811
97.76811 97.76811 97.76811 97.68478 97.60145 97.60145 97.60145 97.60145 97.60145 97.60145
97.60145 97.60145 97.60145 97.56159 97.52174 97.52174 97.52174 97.52174 97.52174 97.52174
97.52174 97.52174 97.52174 97.56522 97.6087 97.6087 97.6087 97.6087 97.6087 97.6087
97.6087 97.6087 97.6087 97.68841 97.76811 97.76811 97.76811 97.76811 97.76811 97.76811
97.76811 97.76811 97.76811 97.81159 97.85507 97.85507 97.85507 97.85507 97.85507 97.85507
97.85507 97.85507 97.85507 97.87319 97.8913 97.8913 97.8913 97.8913 97.8913 97.8913
97.8913 97.8913 97.8913 98.02536 98.15942 98.15942 98.15942 98.15942 98.15942 98.15942
98.15942 98.15942 98.15942 98.28986 98.42029 98.42029 98.42029 98.42029 98.42029 98.42029
98.42029 98.42029 98.42029 98.48189 98.54348 98.54348 98.54348 98.54348 98.54348 98.54348
98.54348 98.54348 98.54348 98.57609 98.6087 98.6087 98.6087 98.6087 98.6087 98.6087
98.6087 98.6087 98.6087 98.52174 98.52174 98.52174 98.52174 98.52174 98.52174 98.52174
98.52174 98.52174 98.52174 98.49638 98.47102 98.47102 98.47102 98.47102 98.47102 98.47102
98.47102 98.47102 98.47102 98.44927 98.44927 98.44927 98.44927 98.44927 98.44927 98.44927
98.44927 98.44927 98.44927 98.44203 98.44203 98.44203 98.44203 98.44203 98.44203 98.44203
98.44203 98.44203 98.44203 98.43478 98.43478 98.43478 98.43478 98.43478 98.43478 98.43478
98.43478 98.43478 98.43478 98.55073 98.55073 98.55073 98.55073 98.55073 98.55073 98.55073
98.55073 98.55073 98.55073 98.70652 98.86232 98.86232 98.86232 98.86232 98.86232 98.86232
98.86232 98.86232 98.86232 99.14493 99.14493 99.14493 99.14493 99.14493 99.14493 99.14493
99.14493 99.14493 99.14493 99.28261 99.28261 99.28261 99.28261 99.28261 99.28261 99.28261
99.28261 99.28261 99.28261 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782
99.34782 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782 99.34782

99.42754 99.50725 99.50725 99.50725 99.50725 99.50725 99.50725 99.50725 99.50725 99.4855
99.46377 99.46377 99.46377 99.46377 99.46377 99.46377 99.46377 99.46377 99.46377 99.4529
99.44203 99.44203 99.44203 99.44203 99.44203 99.44203 99.44203 99.44203 99.44203

Scott Dedoncker Control & Instrumentation Ass 1 FINAL SUB

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

63/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

Text Comment. If you read the brief carefully, only flow sensing is required in your research section.

PAGE 6

Text Comment. this should be related to flow sensing

PAGE 7

PAGE 8

Text Comment. An F to V converter chip is used.

PAGE 9

Text Comment. This is not the method we used!

PAGE 10

Text Comment. Use brackets please: $(T1 + T2)/2$

PAGE 11

Text Comment. Where is the Caldwell curve to explain this?

Text Comment. Use brackets please $T2/(T1 + T2)$

Text Comment. There are no plus signs in these brackets as they are a series of coefficients

Text Comment. This is correct

PAGE 12

PAGE 13

Text Comment. This belongs to the explanation of the model

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

Text Comment. ????????

Text Comment. ????????

Text Comment. Though different, you should still have applied these values and obtained

the results.

PAGE 32

Text Comment. You must select the Harvard option when automatically referencing using MS Word.

PAGE 33

PAGE 34

GRADING FORM: CSG-RUBIC

SCOTT DEDONCKER

63

RESEARCH-INFORMED LITERATURE



Section 1: Your research - Some helpful research carried out with a small range of flow sensing methods. However, you have not read the brief carefully enough. You should have restricted your research to flow sensing. You have not explained the Caldwell method and the Ziegler-Nichols method researched is incorrect. (8/20) Section 4: Your use of Harvard for citing and listing - Good range of sources are cited correctly but for the listing, you must remember to switch on the Harvard option in MS Word. (3/5)

11

KNOWLEDGE AND UNDERSTANDING



Section 2: Explanation of the control system and model - Good explanation of the development of the model given which covers the transport delay and the LUT. (11/15)

11

ANALYSIS



Sections 2 and 5: Analysis of model, simulation and practical testing - Unfortunately, there is no mathematical analysis of the model present in the report. Good evidence of tuning the simulation of the control system for the model with results applied to the practical rig. Further analysis of the results could have been carried out. (10/20) Section 3: Conclusions and recommendations - Some helpful conclusions but not all are correct. (6/10)

16

PRACTICAL APPLICATION AND DEP.



Sections 2: Development of the model, use of Simulink – Good development of the model and use of Simulink. (10/10) Sections 2: Practical tuning – Good use of model tuning values. However, though Ziegler Nichols method was correctly applied, values were not implemented. (3/5) Section 2: Presentation of results - Very good. (5/5)

18

SKILLS FOR PROFESSIONAL PRAC.



All sections - Generally well structured, presented and written (7/10)

7