Process Control Remote-Laboratory for Control Education

Jonathan Zeelenberg, Christopher Herring, Levi Richardson, Sebastian Henn and Tameem Mithaiwala

***Abstract*—The RT578 is a process control machine designed for educational purposes. The School of Chemical and Mechanical Engineering at The University of Western Australia recently commissioned an RT578 with the intention of connecting the machine to the Labs Online system, which provides remote control and data acquisition of a range of educational equipment. Remote access to the machine is need to allow large numbers of students to gain meaningful experience using with the machine. This project involved adapting the RT578 to allow remote operation, development of suitable experimental protocols, and development of a simulator of the machine. Remote acquisition of the sensor data and control of water level were achieved. Recommendations were made for additional modifications required for the completion of the project.**

***Index Terms*—Process control, remote access laboratories, web-based control education, Gunt RT578**

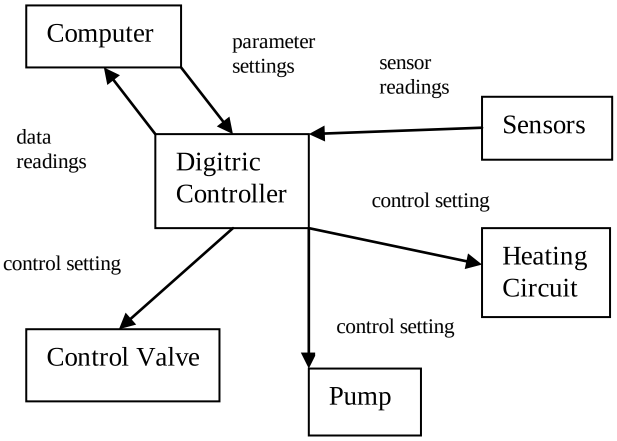


Fig. 1. Function block diagram of RT578 Control Rig

I. Introduction

THIS is the group report for the Remote Labs group of MCTX 3420. This report is a detailed report of our project dealing with the automation and modifications of the RT578 Process Control Rig. Our project was to investigate the most cost effective and efficient method of running laboratories on the machine for future courses. We started by investigating the TeleLabs system which the University has already running several different experiments online such as the iron experiment and the ABB robot. We then started to investigate the options that were possible and that we could use to make the RT578 available for online use. After several different options we came to the decision that the best option available to us was to modify the machine in such a way that the controller on the machine was bypassed by a controller we would install that could control every element of the control rig that we needed. With the time available we were not able to fully implement our plan however we were able to get enough experimental data that showed that our premise was possible and simple to implement.

II. Project Development: Levi Richardson

When this project first started our task was to understand how the current TeleLabs system and its integration with the current lab machines and the National Instruments Fieldpoint. Since the Fieldpoint was bought by the university the standard for field buses has changed and the current one is flaky and unreliable. The idea was to investigate how to replace and upgrade the field bus and link it with all the lab equipment that is currently running on TeleLabs.

Professor J. Trevelyan then told us that our new aim would be to integrate a new process control rig, the RT578, into the TeleLabs system and determining the most cost effective method of running labs on the new RT578. This was decided as the level control RT578 was new and could be used for the new courses that are being implemented at UWA in 2012.

The RT578 is a process control rig designed to allow users to run experiments with different control systems and parameters on 4 different processes and measure their effects and outcomes.

To gain a better understanding of the RT578, its setup and the desired labs we met with Dr Roshun Paurobally as he would be the main user of the RT578 and the labs. From Dr Paurobally we got the first ideas for what the labs need to achieve and how that needs to be done.

Once we got access to the RT578 and performed several basic experiments on it we started to identify several key issues with the machine. Our initial hopes were that we could simply implement a TeleLabs interface such as is used on current machines such as the ABB robot and iron control experiment. However there were several problems which prevented this. The first of these problems was the valves. To run a certain experiment on the machine several valves must be opened or closed in a specific manner which determines which experiment is run as they determine where the water will flow. These valves are manual and so must be changed by hand if another experiment is to be run which makes running online labs difficult without modifying the machine. Another problem we found was that the computer interface with the machine was not as comprehensive as we needed it to be. Currently all that the computer does is read the data and plot it with limited control over the desired level for the controlled parameter. For our purpose we need greater control over the RT578 rig for the user. To properly interface the rig with TeleLabs there needs to be the ability to change the variables on the rig so that different experiments can be run. Another problem was that the pump and heating circuit require a button to be pressed to switch them on. This means that the RT578 can only be used when someone presses the button and so makes automating and adding the RT578 to TeleLabs difficult.

With these problems in mind we came up with 4 different options of what we could achieve.

The first option was to not change or modify the RT578 rig in any way and just run labs as they would normally be run. To see if this was a viable and cost effective option we needed to get an approximate cost of running the labs. From discussions with Prof. Trevelyan we got an approximate value of the number of students who could be expected to be using the lab and the rough cost of running a lab. With approximately 600 students and a maximum of 4 students allowed per lab this gives us a number of 150 laboratories that need to be run. With 2 hours to run each laboratory and a minimum cost of a lab demonstrator about $40 this gives us a cost of running the laboratories as about $12000.

Our second option was to modify the computer connection with the RT578 but leave the rest of the rig alone. The rig could then be switched on and setup in a certain configuration for a certain laboratory and left to run for several weeks while the students did the laboratory through TeleLabs. This option allows for greater use of the rig as it can be accessed all day and a lab demonstrator isn’t required however leaving the rig on constantly could lead to high maintenance costs.

The third option we came up with was to add a second controller which could be used to automate the RT578 then linked to the current Digitric controller through LabVIEW and so when a student logs on through TeleLabs the new controller would automatically set the rig to the correct settings for the lab that the student wishes to run. This option would allow a very flexible setup without removing the current controller or modifying the rig very much. One of the main problems with this option is that linking the 2 controllers could prove very problematic and complicated and could be costly. It would also require modifying the internal connections of the rig which would not be preferential.

Our last option was to replace the current Digitric with our own controller which we could then connect to all the sensors and use to automate the RT578 for online use. While this option requires a better understanding of the circuitry and safe levels of the RT578 it does make it easier to control through LabVIEW and connect to TeleLabs. This option is very similar to the previous option however it takes the inputs from the current controller and routes them to our controller instead.

The new controller in options three and four would be with a copy of the new field bus that the university has bought to replace the old Fieldpoint. The new controller will be explained in greater detail further on.

After a group discussion we decided to focus on option 4 as we believed that this was the best and easiest option to implement on the RT578. We discounted the first option as not only would it be very expensive but the time it would take would be too long. The costs would also be on going where as the others would mostly be a one off expense with lower ongoing maintenance costs. The second option we discarded as we were unsure whether it would be possible to link the current computer interface with TeleLabs. Another reason why we did not choose it was that the experimental options were very limited as the RT578 had to be manually reset every time a different experiment needed to be run. With the third and fourth options similar we decided to go with the fourth for several reasons. One was the fact that to automate the RT578 the pump, control valve and heating circuit they would have to be connect to the new controller and so it would make sense that the sensor outputs would also be connected to the controller. This would simplify the LabVIEW implementation and allow greater control for the user over the RT578. This would allow use to make it so that when someone logged onto TeleLabs and wanted to do an experiment they could choose which experiment they wanted to run and the RT578 would automatically be orientated to this experiment.

With our option chosen we started to implement it on the RT578. This will now be explained in greater detail.
For a given cost of $150 per hour and with a given time of approximately 5 hours per week dedicated to the project for 13 weeks this gives a total cost of $9750 for my labour.

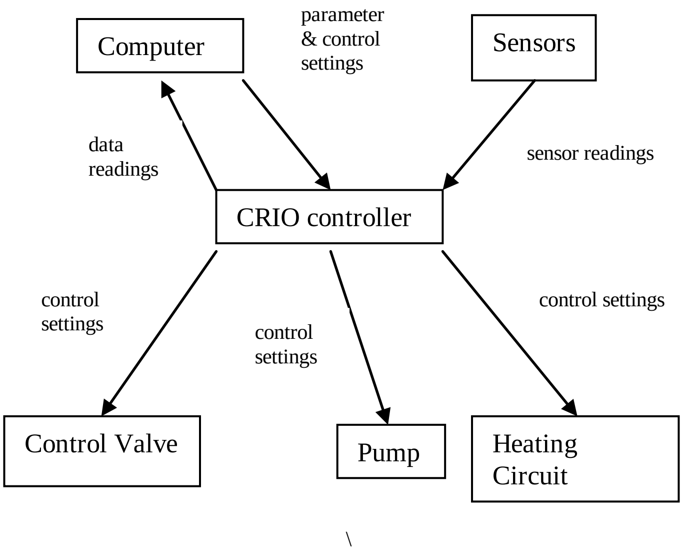


Fig. 2. Function Block Diagram of RT578 with Modifications

III. RT578 – cRIO Interface: Jon Zeelenberg

In order to control the RT578 remotely using a cRIO, it was necessary to design the interface between the two. Initially, an effort was made to understand the main components of the system and how they operated.

The RT578 contained a main storage tank and a level control tank. In level control mode, water was pumped by a variable speed pump, though a control valve, and into the base of the level control tank. Water flowed out of the level control tank, back into the storage tank, at a constant rate which was pre-configured using a manually controlled proportional valve. The steady state water level in the level control tank was therefore determined by the setting of the outflow valve, and the rate of water inflow, which could be controlled by controlling 1) the pump motor speed and 2) the position of the control valve.

At the top of the level control tank, an outflow valve allowed air to escape as the water level rose. In pressure control mode, this valve was closed, allowing the air pressure inside the tank to rise with the water level.

The third actuator was the heating circuit which consisted of a heating element, a small water pump, and a heat exchanger. In temperature control mode, this was used to control the temperature of the process water by diverting water through the heat exchanger and bypassing the level control tank.

Process information was detected using sensors for four variables: pressure, water level, water flow rate, and temperature. Each of these will be discussed separately.

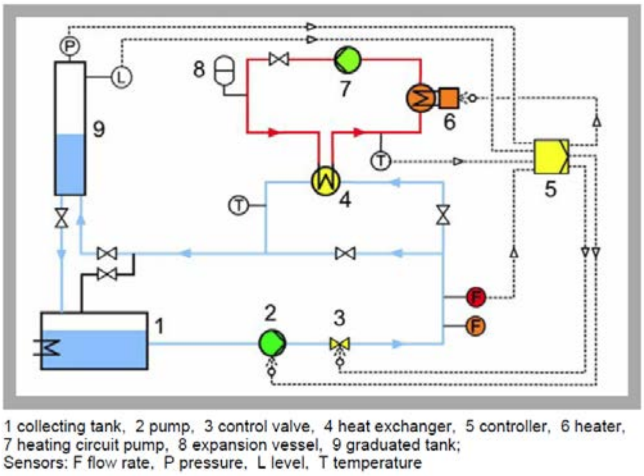

P
L
8
7
6
9
T
W
4
T
5
1
2
3
F
F
1 collecting tank, 2 pump, 3 control valve, 4 heat exchanger, 5 controller, 6 heater,
7 heating circuit pump, 8 expansion vessel, 9 graduated tank;
Sensors: F flow rate, P pressure, L level, T temperature

Fig. 3. Simplified P&ID diagram of the RT578. Adapted from (GUNT 2005).

A. *RT578 Main Components*

1) *Sensors*

a) *Level Sensor*

The Level sensor was an "Elobau 212KKB 02" which was a capacitive type sensor, which measured the water level with a long probe extending down into the tank. The output was a 0-10v analogue output where 0V was low and 10V was high.

b) *Pressure*

The pressure sensor was a "Huba Control Pressure transmitter - Type 691". This was a relative pressure sensor with an operating range of 0-6 bar, and an analogue voltage output of 0-10V

c) *Temperature*

The temperature sensor was located on the machine but had no markings, but appeared to be a resistance type temperature sensor. This output was converted to a 0-10V signal.

d) *Flow Rate*

The flow meter was an "Endress + Hauser Promag 10" which is an electromagnetic type flow meter. It had a pulse output which was converted to a 0-10V signal by further processing.

2) *Actuators*

a) *Pump*

The pump was a centrifugal pump driven by a "Grundfos CR3-10" 750W 3-phase induction motor. The speed was controlled by a "Toshiba VF-S11" inverter.

The pump speed was controlled by a 0-10V input to the inverter, where 0V was the minimum speed setting and 10V was the maximum speed.

b) *Control Valve*

The control valve was a "Samson 3277" pneumatic actuator which used a 0-10V input for control. The valve was fully closed with 0V input, and fully open at 10V.

c) *Heater Circuit*

The heating circuit consisted of a 2kW electric heating element and a "Grundfos UPS 25-40" hot water pump. Both of these devices were controlled via a series of relays -thermostatically- to maintain the temperature in the heating circuit

3) *Controller*

a) *Connections to main components.*

The RT578 was not provided with a circuit diagram, but by investigating the connections inside the electrical cabinet, the schematic in Figure 4 was drawn. The existing controller was only capable of measuring from one sensor at a time, which was connected to the controller using jumpers on the control panel. The controller also had two outputs, a "control output" and a "cascade control output" which were both 0-10V signals.

In order to control the flow rate, water level, or pressure, the controller could be set to control either the pump speed or the control valve. This was done by adjusting two three position switches to either “control” or “cascade”, which connected the controller output to each actuator circuit.

Alternatively, the switch could be set to manual to control the pump speed or control valve by adjusting a potentiometer on the cabinet.

A similar switch controlled the heating circuit, but this was ignored at this stage of the project for simplicity.

Two sets of terminals allowed the inputs of the controller, and outputs being used to control each device to be measured using a multimeter.

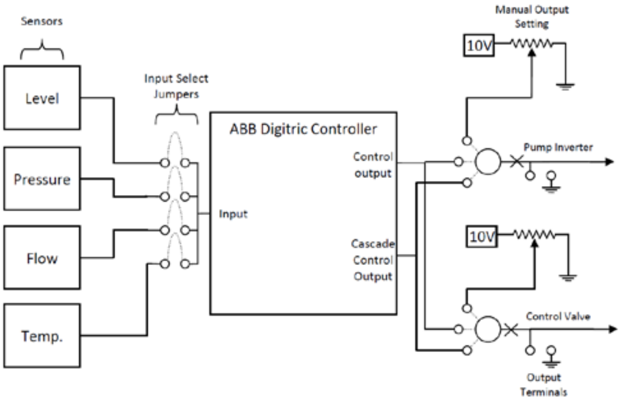


Fig. 4. Simplified circuit schematic of the RT578 controller connections. The heater control circuit is not shown for simplicity.

B. *cRIO/RT578 Interface*

Controlling the RT578 from the cRIO was achieved by making a number of connections between the two and only making minor modifications to the RT578.

a) Connections – Wiring between the cRIO and the RT578.

The terminals on the front of the RT578 control panel provided easy access to the sensor signals. The input terminals were connected to using banana terminals which were wired to the input card on the cRIO. The outputs from the cRIO were connected to the output terminals on the RT578.

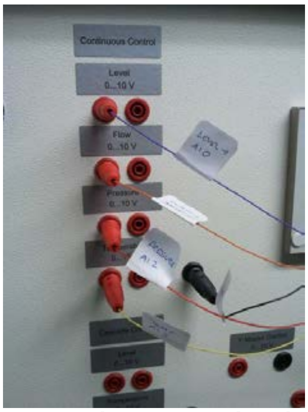


Fig. 5 Connections to the input terminals on the RT578.

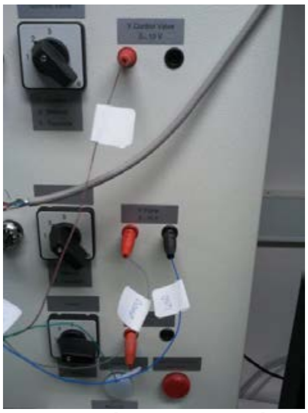


Fig. 6. Connections to the output terminals on the cRIO.

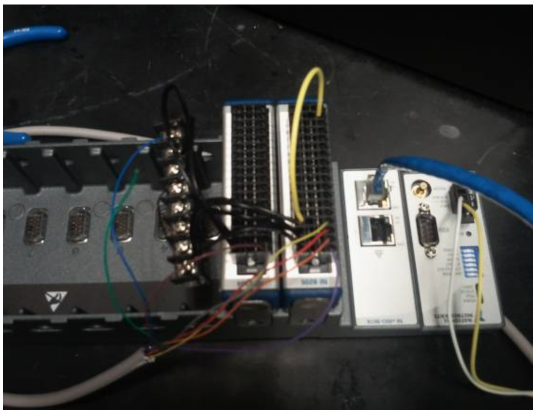


Fig. 7. Connections to the cRIO I/O cards.

b) Modifications made to the RT578

In order to drive the voltage settings on the RT578 used to drive the pump speed and the control valve via the terminals on the control panel, it was necessary to make some minor modifications to the machine. The output terminals were intended to be used to *measure* the signals being sent to the actuators, so driving these voltages externally may cause damage to the controller or other components.

The simplest solution was to disconnect each output from its three position switch. Figure 4 shows the two connections broken to make the modifications –marked with an ‘X’.

After making these connections and alterations, it was demonstrated that the RT578 could be controlled remotely using the cRIO.

C. *Control Considerations*

1) *Pump Protection.*

According to the pump installation manual (Grundfos), the minimum flow rate for the pump is 10% of the rated maximum flow rate of 3m^3/h, and the pump should never be operated against a closed outlet valve. This minimum flow rate assumes that the pump is operating at 50Hz. No information could be found on the relationship between input power frequency and the minimum flow requirements. The risk of low flow rate is that it may cause the pump motor to overheat.

The minimum and maximum speed settings for the pump had programmed into the inverter at the factory, so that an input of 0V or 10V cannot result in damage to the pump. However, the outflow from the pump passed through the control valve and it was possible to close off the valve completely. Therefore, an extra layer of pump protection was necessary.

A bypass valve allowed water to be bypassed around the control valve. A simple way to provide protection to the pump would be to fix this valve open to a position where the minimum pump flow requirement would be satisfied under the conditions of a maximum pump motor speed setting and the control valve fully closed. This would satisfy pump protection, but may greatly reduce the effect of the control valve on flow rate, particularly at low flow rates when the resistance of the bypass path is minimal.

A control solution would involve protecting the pump from a low flow condition by monitoring the system flow rate, and maintaining the control valve position accordingly, or simply by setting a minimum control valve position. Using software to do this may be a particular problem in this case, since one of the goals of the machine is to allow students to reprogram the controller. This may result in software errors and cause damage to the pump. This safety mechanism may also fail if the control valve itself fails, or if any of the valve control circuitry fails (e.g. if the cable becomes disconnected) which also seems plausible with student involvement.

The best solution would be to combine a control solution with additional hardware protection. Replacing the bypass valve with a pressure-actuated relief valve would provide protection to the pump in the event of low flow. This solution would protect the pump at the rated speed (50Hz power supply), but may not at lower speeds. It is possible that at lower speeds, the pressure generated may be too low to activate the relief valve, even though the flow rate may be low enough to cause the motor to overheat.

Another hardware-implemented protection mechanism may be to drive an actuated bypass valve using a temperature sensor placed on the motor casing. This would allow an increased flow rate in the event that the pump gets too hot.

2) *Control Tank Overflow Protection*

In level control experiments, the control tank overflow return path is left open to allow air to be vented from the top of the tank and to allow water to flow back into the main tank when it overflows.

There was also a pressure relief valve on the top of the tank which was used to relieve built up pressure during the pressure control experiments. If this was left open during the a level control experiment, the tank overflow would result in water being ejected from this valve (and into the laboratory) since it is mounted below the tank overflow outlet port. It was therefore reasoned that tank overflow should be prevented if possible.

To avoid this possibility, it was determined that the water tank outflow valve should be set (manually) to approximately 35 degrees, and the control valve bypass valve should be closed. In this configuration, it was impossible to overflow the tank, even with the pump set to maximum speed and the control valve fully open.

The safe operation conditions were determined by setting the control valve fully open, and the pump on full speed, then gradually closing the tank outflow valve until a level of approximately 55cm (tank approximately ¾ full) was maintained at steady state.

Although the above settings resulted in a safe mode of operation, they may represent a constraint on experimentation. A small reduction in pump speed, or a small amount of valve closure results in the water level dropping to near zero.

D. *Costing*

TABLE I
COSTS OF THIS PART OF THE PROJECT

Engineering costs	Unit Cost	Qty	total Cost
Engineering Time	$150.00	50	$7,500.00
Workshop costs			
Technician Time	$80.00	2	$160.00
Component Costs			
Misc. Components	$25.00	1	$25.00
cRIO (Estimated)	$3,366.00	1	$3,366.00
Total			$11,051.00

The total costs of this part of the project are summarized in Table 1. The cost of the cRIO was not the cost of the actual unit used since it was borrowed, and had more capability than required. The cost for the cRIO was estimated based on the actual requirements for this project.

E. *Future Work*

Some control elements of the RT578 require manual control and cannot be automated until control actuators are added.

1) *Pump and Heater Circuit Switch*

The pump and the heater circuit must be started by pressing a button on the control panel. These switches drive a relays which provide power to the pump and heater circuit. In order to fully automate the RT578, these relays need to be controlled remotely by the cRIO. Currently, the pump and heater circuit must be left running in order to control the machine remotely.

2) *Actuated Configuration Valves*

The machine has a number of configuration valves which allow the user to change the kind of experiment to perform (e.g. level, pressure or temperature control). The valves are used to divert water through the heater circuit, bypass the control tank or bypass the control valve. There are also two proportional valves used to alter the control tank outflow rate, and also the proportion of process water flowing through the heat exchanger. In order to fully automate the machine, and be able to switch between configurations and experiment types, these valves must be actuated and controlled remotely by the cRIO.

3) Heater Circuit Control

For simplicity, the heating circuit control was not implemented and has been left as future work.

IV. RT578 – Hardware Client: Chris Herring

A. OVERVIEW

The RT578 control rig may be controlled directly from the National Instruments cRIO. As the sensors may all be accurately read using the NI9205 ±10v analogue input module, and all outputs (the pump VSD, pneumatic valve and, subject to future work, the heating element) may similarly be driven using the NI9264 ±10v analogue output module. Each of these inputs, or a combination of multiple inputs may be set as the feedback variable(s) in a control loop. Whilst any of the actuators, or multiple, may be set as the control variable(s) in order to obtain some desired control system.

The cRIO FPGA device may be programmed with a master VI. This master VI will make available all input/output parameters for the client VI's. Depending on the current physical configuration the RT578 control rig (ie position of valves, etc), some of these parameters should be restricted in order to ensure safe operation.

In order to define the current state of the control rig such that the master VI will be aware of the current physical configuration the system, two options are made available;

1) Manually set the state via the local master LabVIEW VI, as valves and switches are manually actuated by the lab supervisor;
2) Replace valves and switches with electronically actuated devices which, as a certain client lab specific VI is loaded, will change the state of the master VI and automatically set valves and switches accordingly.

The first option will require an operator to physically set up the RT578 rig for each experiment. Clients should not be able to load a lab specific VI for which the control rig is not set up for as it may damage the rig. This should be implemented both via TeleLabs (to stop the VI being loaded) and also via the master VI (to prevent any parameters being set if the machine is not in the appropriate state and the incorrect VI is allowed to load).

The second option utilizes a digital output cRIO module, all valves and manual switches will be replaced with electronically actuated devices which may be controlled by a digital 'on/off' signal from the cRIO. This design allows any lab to be run at any time as when a lab specific VI is loaded, the master VI will change to the appropriate state, setting digital outputs to actuate valves and switches into the required configuration.

These two options may of course be used in conjunction with one another, depending on requirements course structure(s). The first option is allowing several labs to be run at a time, but not all. An example of this is if two units utilize the control rig simultaneously, however due to the order of course content, only one lab is required to be run per unit in an allocated time period. This situation would require only two labs out of many to be allowed to run. Automatic actuation is desired; however a restriction on the allowed states of the control rig must be manually set. This example may be extended for an arbitrary number of units, simultaneous laboratories and restricted states.

B. CONTROL TECHNIQUES

A key advantage of using a LabVIEW based hardware client is the large range of options available for control techniques. The hardware client currently has installed the *PID and Fuzzy Logic Toolkit* for LabVIEW which allows P/PI/PID algorithms to be easily implemented into a control system, with control over all controller gains (National Instruments, 2011). Toolkits and third party software also exists which can be used to implement these along with other higher level control techniques if required. In the absence/unavailability of a pre-designed piece of software for a particular control technique, a controller may be designed within the LabVIEW environment. This creates the opportunity for the control rig to be used for a broad range of purposes, from education at all levels to experimental and research.

C. PROOF OF CONCEPT

In order to demonstrate basic operation of this system, two experiments were designed;

a. Basic IO VI
b. Basic level control VI

The Basic IO VI was designed to investigate the operation and accuracy of the ±10v analogue input/output modules (NI9205/NI9264 respectively). This VI consists of two floating point number fields with accuracy of two significant figures. One of these number fields is a user variable value wired to a FPGA node which is set to channel 0 of the analogue output module. Whilst the other is an indicator wired to a FPGA node set to channel 0 of the analogue input module. The output value was set in incrementing steps from 0v to +10v, then in decrementing steps from +10v to 0v and output voltages were measured. Similarly, a bench top power supply was used to apply a voltage to the input module, allowing a reading to be obtained from the indicator field of the LabVIEW VI. The voltage range and order of readings for the input test was the same as for the output test.

All external voltage reading (bench top power supply and analogue output module) were taken using a *Fluke 87* multi meter. The relevant specifications for which can be seen in Table II below (Fluke 87 Datasheet).

TABLE II
FLUKE 87 DC VOLTAGE SPECIFICATIONS

Range: 0-40 V; Temperature: 18-26°C	
Resolution	**Accuracy**
0.01V	±1% + 1

The results of the output experiment may be seen in Table III below:

TABLE III
ANALOGUE OUTPUT MODULE TESTS

Set Voltage (V)	Measured Voltage (V)	Measured Voltage Accuracy (V)	Error (V)
0.10	0.10	0.01	0.00±0.01
0.80	0.81	0.02	-0.01±0.02
1.21	1.21	0.02	0.00±0.02
1.60	1.60	0.03	0.00±0.03
2.22	2.21	0.03	0.01±0.03
2.80	2.81	0.04	-0.01±0.04
3.40	3.40	0.04	0.00±0.04
4.00	4.02	0.05	-0.02±0.05
4.64	4.63	0.06	0.01±0.06
5.48	5.49	0.06	-0.01±0.06
6.02	6.02	0.07	0.00±0.07
7.55	7.52	0.09	0.03±0.09
8.26	8.24	0.09	0.02±0.09
9.00	9.01	0.10	-0.01±0.10
10.00	9.99	0.11	0.01±0.11
9.50	9.52	0.11	-0.02±0.11
8.62	8.61	0.10	0.01±0.10
8.02	7.99	0.09	0.03±0.09
7.45	7.45	0.08	0.00±0.08
6.22	6.21	0.07	0.01±0.07
5.64	5.64	0.07	0.00±0.07
4.88	4.89	0.06	-0.01±0.06
3.89	3.90	0.05	-0.01±0.05
3.04	3.03	0.04	0.01±0.04
2.15	2.15	0.03	0.00±0.03
1.50	1.51	0.03	-0.01±0.03
0.06	0.06	0.01	0.00±0.01
0.00	0.01	0.01	-0.01±0.01

These results indicate that digital output module is accurate to within the range of the available measuring equipment. This is sufficient accuracy for controlling the control rig actuators. It is observed during the experiment that the response to a change in output value is extremely quick and no voltage overshoot is seen.

Table IV shows the results of the input experiment, input voltages were set via a bench top power supply and read from the *Fluke* multi meter, then compared to the values read by LAVBiew via the analogue input module.

TABLE IV
ANALOGUE INPUT MODULE TESTS

Set Voltage (V)	Measured Voltage (V)	Set Voltage Accuracy (V)	Error (V)
0.01	0.00	0.01	0.01
0.21	0.20	0.01	0.01
0.61	0.60	0.02	0.01
0.19	0.21	0.01	-0.02
1.88	1.89	0.03	-0.01
2.64	2.64	0.04	0.00
3.51	3.50	0.05	0.01
4.50	4.49	0.06	0.01
5.49	5.51	0.06	-0.02
6.02	5.99	0.07	0.03
7.02	7.04	0.08	-0.02
7.99	8.01	0.09	-0.02
8.23	8.22	0.09	0.01
9.16	9.15	0.10	0.01
10.02	9.99	0.11	0.03
9.01	8.99	0.10	0.02
8.11	8.10	0.09	0.01
7.20	7.18	0.08	0.02
5.99	6.00	0.07	-0.01
4.01	3.99	0.05	0.02
3.21	3.21	0.04	0.00
2.03	2.02	0.03	0.01
1.00	1.00	0.02	0.00
0.02	0.10	0.01	-0.08

The conclusions reached from this input test are identical to those of the output test, the accuracy of the NI9205 input module is within the tolerance of the available measuring equipment. This is sufficient for reading the sensor output values.

The basic level control VI was designed to demonstrate the control capabilities of LabVIEW via the cRIO. A simple SISO PID controller was built using LabVIEW. After wiring of the machine inputs and outputs to their respective cRIO modules had been completed, this controller was used to control the height of the liquid in the tank. Three FPGA nodes were used in this VI;

The feedback variable used was the level sensor, located on analogue input AI0.

The control variable was the pneumatic valve, located on the analogue output AO1

And finally in order to activate the pump, a constant 10v output was applied to analogue output channel AO0, which set the pump to full speed allowing the valve to have full control.

The VI contains a cluster of input fields which allow the user to modify the proportional, integral and differential constants of the controller, as well as an input field allowing the user to set a set point between 0 and 10. Also included are fields displaying both graphically and numerically the current value of the feedback variable (sensor voltage) and the output of the PID controller (the control variable).

This simple LabVIEW program successfully implemented a level control system using the RT578 control rig. Seen below are screenshots of the VI as the set point was increased and output is at maximum in order to increase flow, and where set point was decreased and output is small in order to reduce flow, lowering level (Figures 8 and 9 respectively)

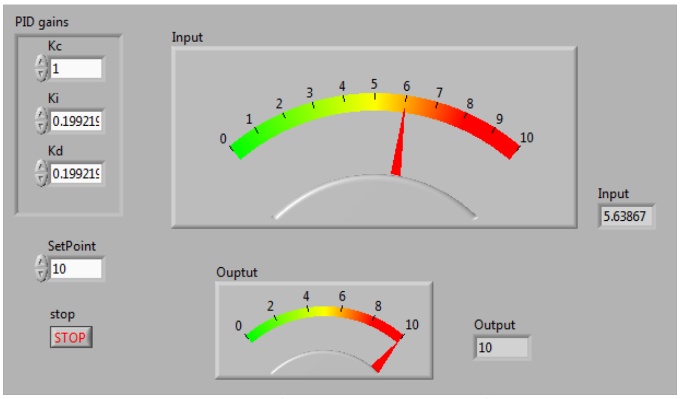

PID gains
Kc 1
Ki 0.19921
Kd 0.19921
SetPoint 10
stop STOP
Input
Input 5.63867
Ouptut
Output 10

Fig. 8. Set point increased, PID output is increased to compensate.

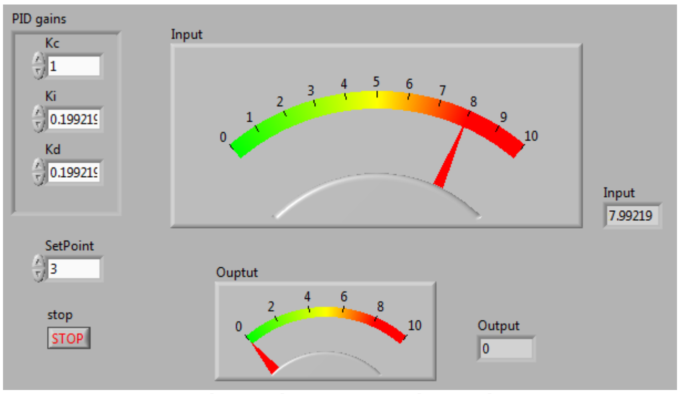

PID gains
Kc 1
Ki 0.19921
Kd 0.19921
SetPoint 3
stop STOP
Input
Input 7.99219
Ouptut
Output 0

Fig. 9. Set point decreased, PID output is decreased to compensate.

Although the two aforementioned basic VI's serve no practical purpose for remote control applications, they serve as a proof that the physical modifications made to the control rig combined with the use of a LabVIEW based hardware client do indeed satisfy the requirements for implementing more complex control systems. The feedback and control variables used in these examples may easily be changed to suite the required experiment using software alone, with no physical modifications. As such it is possible to implement multiple control systems and alternate between these as required. More complex systems, including MIMO (multiple input/multiple output) designs may also be implemented.

The FPGA project containing these VI's may be found in the compressed archive "*Project.rar*" submitted with the electronic copy of this report.

D. cRIO CURRENT SETUP

The cRIO is currently set up with the following modules:

TABLE V
CRIO SET UP

Expansion Slot	Module
1	NI2905 Analogue Input ±10v
2	NI2964 Analogue Output ±10v
OTHER	EMPTY

A static IP address has been setup on the cRIO, and university network as:

STATIC IP ADDRESS
130.95.52.81

V. EXPERIMENTS (SEBASTIAN)

The GUNT RT578 is an industrial standard variable controller that is designed to be used in an educational environment to teach various methods of control in process engineering courses. Students will use the equipment to gain an understanding about how each of the control parameters can be tuned in order to improve the response of the control variables. As there will be many students that will require access to the machine to perform the experiments, it would be more efficient to allow remote control through the TeleLabs system. This way, the students are able to conduct the labs in their own time and from home. Remote access would also rescind the need to employ a laboratory demonstrator (university policy currently states that there must be at least one demonstrator present for every 4 students)

A. PID Control

PID control is a method by which variables are adjusted until they reach a desired level. The difference between the actual and desired value of the variable is calculated (the error), and through the use of feedback, the controller adjusts the variable according to the PID parameters. PID control is split into 3 components, proportional, integral, and derivative.

The proportional component (set by the gain variable K_p) changes the output variable proportional to the error value. A high gain can result in a large overshoot, which in some cases leads to an instability. A low gain will result in a smaller response to a given error, and so the response time will be slower. In general, it is best to use the proportional term as the major driving force of the PID controller (*University of Michigan*, 1997).

A proportional controller alone will not always cause the system settle on the target value. To mitigate steady state errors, an integral term is included. The gain K_i is multiplied by the integral of the error (the magnitude of the instantaneous error multiplied by the duration) and improves the response time towards the desired value while eliminating any steady state errors. If the gain is too high, however, the system will overshoot the desired value.

The derivative component looks at how quickly the error is changing over time (the slope of the error multiplied by the derivative gain K_d) and is included to reduce the magnitude of the overshoot resulting from the integral component. The derivative component acts as a dampener, and slows the transient response of the system.

B. *Ziegler Nichols Tuning*

There are a number of tuning techniques that aim to provide a somewhat systematic way in choosing the K_p, K_i, and K_d parameters of a PID controller. John G. Ziegler and Nathaniel B. Nichols developed two methods for tuning a PID controller that will be explored in the laboratory.

1) *Z-N Oscillation Method*

Valid only for open-loop stable control systems, the oscillation method begins by setting the proportional gain K_p to a low value, and setting the other parameters (K_i and K_d) to zero. K_p is then increased until the critical gain has been found, at which point the output will oscillate with constant amplitude. The value of the critical gain K_{cr} and corresponding period P_{cr} are recorded and used to determine the P, I and D parameters (*National Instruments*, 2011):

TABLE VI
DETERMINATION OF PID PARAMETERS

Control Type	K_p	K_i	K_d
P	$0.5\ K_{cr}$	-	-
PI	$0.45\ K_{cr}$	$1.2\ K_{cr}/P_{cr}$	-
PID	$0.6\ K_{cr}$	$2\ K_{cr}/P_{cr}$	$0.125\ K_{cr}/P_{cr}$

This method is designed to smooth out the response curve as much as possible and typically leads to the best disturbance rejection performance.

2) *Z-N Reaction Method*

The process reaction curve method examines the behaviour of an introduced disturbance to the system once it has reached steady-state. This disturbance must be measured without external control, so for this method to work, the system must be configured in open-loop.

The controlled variable is changed by *x %*, resulting in to a response similar to the graph below (*Shaw J.* 2002) (Fig. 10).

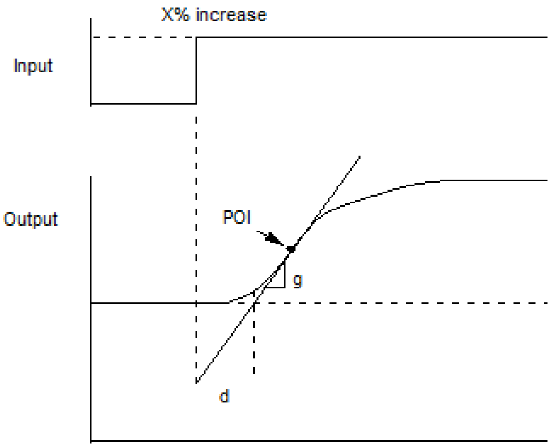


Fig. 10. Function Block Diagram of RT578 with Modifications.

From the response curve, the point of inflection (*POI*) is identified, and a tangential line is drawn from this point (with gradient *g*). The time delay *d* between the input change and the point at which the tangent line reaches the initial steady-state value is also recorded.

The PID parameters are then calculated from the values of *x*, *g* and *d*.

TABLE VII
DETERMINATION OF PID PARAMETERS

Control Type	K_p	K_i	K_d
P	$x / (dg)$	-	-
PI	$0.9\ x / (dg)$	$0.3 / d$	-
PID	$1.2\ x / (dg)$	$0.5 / d$	$0.5\ d$

C. *RT-578 Control*

The RT-578 has a selection of jumpers located on the front of the circuit box that controls the variable that is to be adjusted (temperature, pressure, liquid level or flow rate). Under normal circumstances, manual access is required to switch the jumpers to the corresponding parameter, but the machine has been modified so that all four control variables can be directly measured and adjusted with a secondary controller (cRIO), which is connected to the TeleLabs system.

The machine manages these variables through PID control. The user is able to define the PID parameters and the variable set-point through the same TeleLabs system, and can then observe the transient response as the machine adjusts itself accordingly. Experiments with the RT-578 are designed to give the user some understanding about how these parameters affect the response, and explore some systematic ways in which the PID parameters can be tuned in order for the machine to meet certain response criteria.

The RT-578 has at least 10 manually operated valves that control the flow of the liquid. This makes remote operation difficult, and these valves will eventually be replaced with electronic actuators so that the user can remotely control the fluid flow. However, it is possible to manually set the valves for a particular experiment at the start of the given period, and simply change it over for the next experiment once the students have all completed the first.

Finally, the cRIO has been connected to the pump and the control valve, allowing the user to begin and stop the experiment through the TeleLabs system and allowing the possibility of circumventing the internal PID controller entirely.

D. *Sample experiment: Level Control*

In this experiment the pump directly fills the main container in the presence of a small constant outflow. The valves are adjusted so the liquid is forced through the control valve (III) by the pump (II), into the process tank (X) whilst avoiding the heat exchanger (VI) (Figure 3). This is achieved by setting the valves in the following configuration (*GUNT Training Manual*, 2009):

TABLE VIII
VALVE CONFIGURATIONS

HV1	HV3	HV4	HV5	HV6
Closed	Closed	Open	Open	Closed

HV7	BV	XIV	XV	XVI
Open	Open	Open 0°	0° - 90°	Closed 90°

The valve XV dictates the outflow rate, and can be adjusted: setting it to 0° indicates that the valve is open and that the liquid will escape from the process tank at a maximum rate, while 90° means that the valve is closed and all of the liquid will remain in the tank.

1) Set valve XV to approximately 20° and K_p to 0.7, so that initially there is a large output flow-rate. K_i and K_d are set to zero.

2) Set the level variable to 50% and note transient response. The pump will be working quite hard to compensate for the large output flow-rate, and the system will settle with a large steady-state error.

3) After letting the tank drain completely, adjust valve XV to approximately 40° and repeat step 2. Compare the new response and note in particular the difference in steady-state error. Set the level back to 0% and let the tank completely drain once more.

4) Using the Z-N Oscillation method, repeat the experiment with increasing K_p values until the critical gain is found and use this to calculate the parameters for P, PI and PID control. Compare the output response curve for each type of control.

5) Change the control to open-loop and find the PID parameters using the Z-N Reaction Method. Modify the value of the control variable and measure the output response. Graphically, find *g* and *d*, and use this to calculate the parameters for each type of control, and compare the output response curves.

E. *Extension*

In order to really gain the maximum understanding about PID systems and their applications to mechatronic control systems, students will be encouraged to develop their own simulations for this system and program their own PID controllers to be used for remote software access. With the exception of the individual valves HV1 to XVI and the heating circuit, all of the inputs (the main pump and the control valve) and the key machine parameters (temperature, pressure, level and flow-rate sensors) can be manipulated remotely. This means it is in fact possible for students to create a LabVIEW program that could control the RT-578 whilst completely bypassing the Digitric Controller.

This way, students will understand the value and core concept of PID control, and in addition to learning methods of optimising the gain parameters, it provides them with a hands-on opportunity to apply knowledge gained from lectures in an efficiently run lab system. Although complete remote access is still not perfected, the basic requirements have been laid out and it is already possible to conduct all experiments discussed in this section.

F. *Labour Costs*

For a given cost of $150 per hour and with a given time of approximately 4 hours per week dedicated to the project for 12 weeks this gives a total cost of $7200 for my labour. There were no other associated costs with my contribution.

VI. Simulation: Tameem Mithaiwala

A. Introduction

A simulation of the level control experiment of the laboratory was implemented in LabVIEW. The purpose is to enable the user to spend their time efficiently while waiting in the queue for accessing the Telelabs experiment. It also familiarizes the user with the experiment before performing it on the equipment through the remote client. In this way, students can gain useful feedback before commencing the graded assessment. Also, the integration of remote-labs and simulation experience will prove to benefit the students as it will enable them to enforce their theoretical knowledge more extensively than when either of them is conducted individually (*J. Ma and J. V. Nickerson* 2006).

The model has been designed for level control of water in the tank, but can easily be modified to implement the other control parameters i.e. pressure, temperature and flow rate control.

B. Mathematical Model

Various mathematical models have been used to simulate level control of liquid in a tank. Pourabally (2010) based his system dynamics on the assumption that there exists a sensor measuring the water level at any given time. However, in the simulation this has to be mathematically determined and hence a model defining the height of the system at any instant time has been used. The following assumptions were made prior to designing the system:

1. The water tank is cylindrical; its area is calculated using:

$$A = \pi r^2 \tag{1}$$

2. Liquid Density remains constant.
3. The pressure at the bottom of the tank is given by:

$$P = \rho g h \tag{2}$$

The height, at any time t, is given by a differential equation - Equation (2) - which has been derived in Appendix A.

$$\frac{dh}{dt} = \frac{1}{A}\left(K_{in}\, v(t) - K_{out}\sqrt{\rho g h\,(t)}\, \right) \tag{3}$$

The height is calculated (using the simulation) by integrating it from 0 to any time t. The initial condition i.e. the height of water in the tank at time 0 is represented by h_0 in the model and is specified by the user. A block diagram of the mathematical model is shown in Figure 11. LabVIEW is then used to simulate this mathematical model.

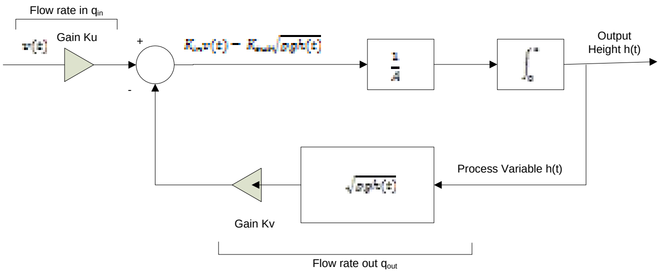


Fig. 11. Mathematical Block Diagram for Determination of Height.

C. PID

PID is used to control the water level in the tank. A PID controller is a control loop feedback mechanism used to achieve steady state and transient performance of general control systems (*Paurobally*, 2010). It is a summation of Proportional, Integral and Derivative parameters and is modeled as depicted in the block diagram in Figure 12. For a more detailed discussion, refer to Section V.

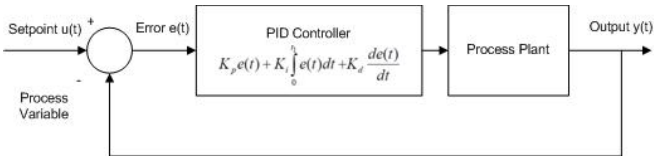


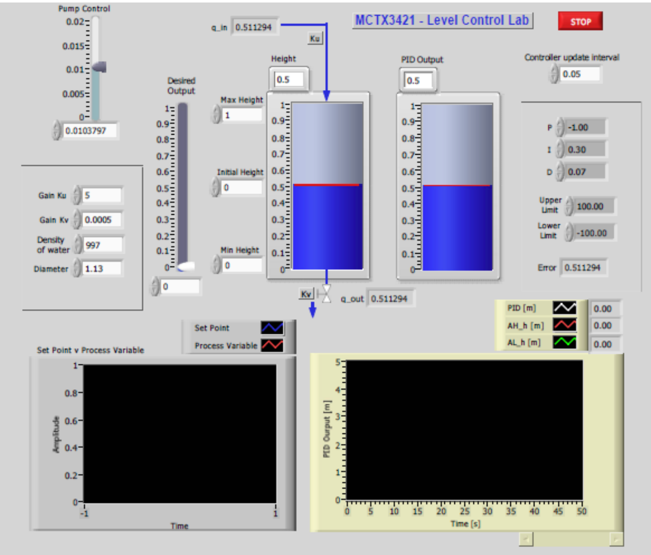
Fig. 12. Block Diagram of a PID Controller

D. Simulation Model

The simulator is designed to mimic the RT-578 controlling the liquid level in a water tank. The mathematical model of the system (Equation (2)) is implemented using simple arithmetic operation available in LabVIEW. Though not implemented in this program, it is possible for the student to save data received through performing the simulation.

The user interface for the simulator is shown in Figure 13. When running the simulation, the user has control over a wide range of parameters which incorporates flexible learning techniques. These include:

1. Input flow rate: Adjusted by changing the input signal for the pump v(t).
2. Setpoint: The desired water level in the tank


MCTX3421 - Level Control Lab

Pump Control 0.0103797
q_in 0.511294
Height 0.5
PID Output 0.5
Controller update interval 0.05
Desired Output 0
Max Height 1
Initial Height 0
Min Height 0
Gain Ku 5
Gain Kv 0.0005
Density of water 997
Diameter 1.13
P -1.00
I 0.30
D 0.07
Upper Limit 100.00
Lower Limit -100.00
Error 0.511294
q_out 0.511294
PID [m] 0.00
AH_h [m] 0.00
AL_h [m] 0.00

Fig. 13. Front Panel (user interface) of the Simulator.

3. PID Parameters: Values for K_p, T_i, T_d and the setpoint limits.
4. Values for the mathematical model.

The resultant output height without control is shown in Tank 1, while the PID Output at each step is displayed in Tank 2. There are two graphs providing the student with relevant information about the outputs. The first graph compares the setpoint with the process variable while the second graph plots the PID output against time.

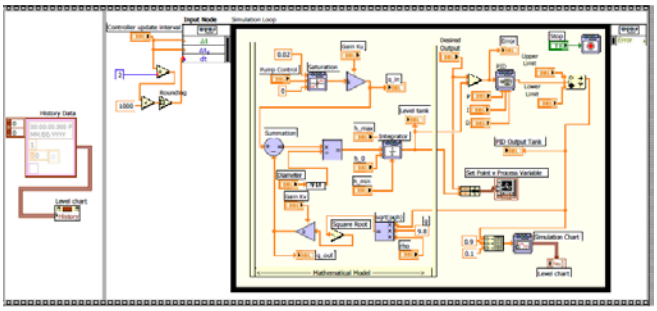


Fig. 14. Block Diagram of the simulator

E. *Problems Encountered*

It was unclear on how the RT-578 was modeled as it had not been specified in any of the manuals provided, and all attempts in communicating with the developers were in vain. Hence, a general model was created and implemented.

Inexperience in LabVIEW programming and lack of time also proved a hindrance in simulating the system. However, this program can act as a basis for future projects and can be built upon to produce a fully working simulator.

F. *Labour Costs*

TABLE IX
LABOUR COSTS

Title	Cost ($)	Hours	Total Cost
Estimated Cost of Time Spent on Project	150.00	48	7,200.00
LabVIEW Maintenance costs			
Technician Time	80.00	1	80.00
			7280.00

VII. CONCLUSION

After much consideration it has been decided that the most appropriate and efficient method if implementing online laboratories utilizing the RT578 control rig is to make the appropriate physical modifications and control the rig directly from a LabVIEW based hardware client. These physical modifications have been made and tested.

From tests it has been discovered that these modifications are sufficient for controlling all aspects of the rig, with the exception of the heating element which requires some more advanced modifications to TTL circuitry.

Several LabVIEW VI's have been written to test the control capabilities of the LabVIEW server via a cRIO and these modifications made to the machine. These VI's succeeded on all accounts in proving that the LabVIEW server was capable of reading all sensor data and controlling all actuators.

Furthermore, a basic level control PID was successfully implemented, demonstrating directly that a LabVIEW based hardware client for control is a valid concept.

Proposed future work on the machine will allow for fully automatic control of the physical states of the machine, allowing it to be run entirely remotely for multiple labs at once.

APPENDIX

Appendix A – Modifications made to the RT578:

Switches S4 and S5 are mounted on the control panel door and are used to switch the outputs from the existing ABB controller to the pump motor inverter and the control valve. The output from each of these switches has been disconnected. Reconnection of these wires will return the machine to the factory configuration.

Caution: If these wires are re-connected, the machine should not be operated remotely or damage may occur.

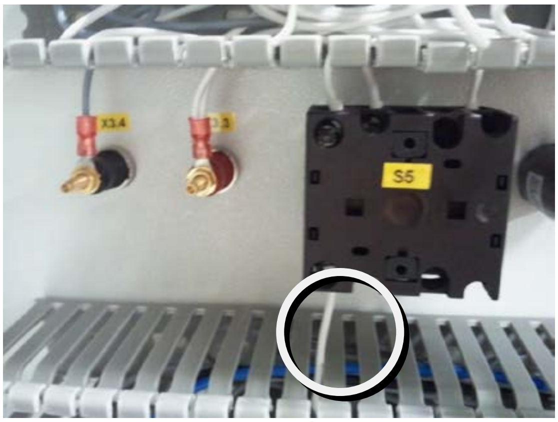


Fig. 15. Wire Indication 1

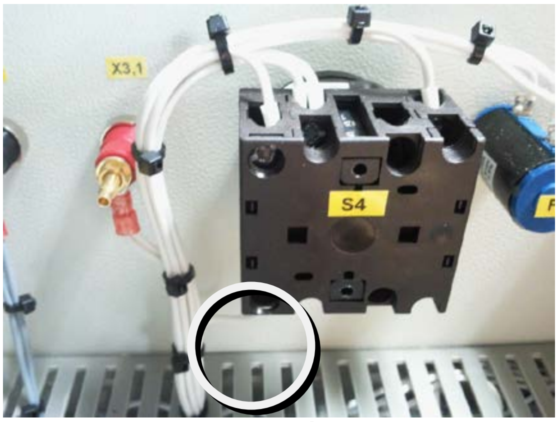


Fig. 16. Wire Indication 2

Appendix B – Derivation of Height of water in the tank

The model of the system has the water flowing into the tank from a pump at a specified flow rate

- Using equation for mass of water in the tank,

$$m = \rho V \qquad (4)$$

where,

ρ is the density of water with a default value of 997 kg/m^3 at room temperature.

V is the volume of water in the tank.

Therefore, at any instant t,

$$m(t) = \rho A h(t) \qquad (5)$$

- The flow rate of water into the tank from the pump is given by,

$$Q_{in} = K_{in} v(t) \qquad (6)$$

where,

K_{in} is the proportionality constant.

$v(t)$ is the current used to drive the pump

- Pressure at the bottom of the tank is given by (*Johnson* 1997):

$$P = \rho g h \qquad (7)$$

- The outlet volumetric flow through the valve is proportional to the square root of the pressure drop over the valve. This pressure drop is assumed to be equal to the hydrostatic pressure at the bottom of the tank (sqrt means square root):

$$Q_{out}(t) = K_{out}\sqrt{\rho g h(t)} \qquad (8)$$

- Conducting a mass balance on the total tank-water mass

$$\frac{dm}{dt} = \rho Qin(t) - \rho Qout(t) \qquad (9)$$

Using (5), (6) and (8) in (9),

$$\frac{d(\rho A h(t))}{dt} = \rho K_{in} v(t) - \rho K_{out}\sqrt{\rho g h(t)} \qquad (10)$$

which can be simplified to,

$$\frac{dh}{dt} = \frac{1}{A}[\, K_{in} v(t) - K_{out}\sqrt{\rho g h(t)} \qquad (11)$$

References

[1] *PID and Fuzzy Logic Toolkit, 2011* http://sine.ni.com/nips/cds/view/p/lang/en/nid/209054

[2] Fluke 87 datasheet http://assets.fluke.com/manuals/87______umeng0800.pdf

[3] Grundfos, *CR, CRI, CRN Installation and Operating Instructions*. Available from: <http://www.byr.com.co/site/pdfs/industrial/Grundfos/Industrial_Grundfos/Multietapas_CR/Instrucciones_CR.pdf>. [8-10-2011].

[4] GUNT, *RT 578 Control of 4 Variables from Process Engineering*. Available from: <http://www.gunt.de/static/s3693_1.php?p1=&p2=&pN=;;#>. [9-10-2011].

[5] University of Michigan, *Control Tutorials for Matlab - PID Tutorial*, available at (http://www.engin.umich.edu/group/ctm/PID/PID.html, Nov 2011)

[6] National Instruments, *PID Theory Explained* available at (http://zone.ni.com/devzone/cda/tut/p/id/3782, Nov 2011)

[7] Shaw, John A. *PID Chap 6. Loop Tuning* available at (http://www.jashaw.com/tutorial/pid6.html, Nov 2011)

[8] GUNT Hamburg, Experiment Instructions for the RT-578, Training Manual, Germany 06/2009

[9] The Engineering Toolbox 2003. *Orifice, Nozzle and Venture Flow Rate Meters*, accessed 20 October 2011, <http://www.engineeringtoolbox.com/orifice-nozzle-venturi-d_590.html>.

[10] Johnson, C 1997. *Pressure Principles*, NI Developer Zone, accessed 12 October 2011, <http://zone.ni.com/devzone/cda/ph/p/id/190>.

[11] Honrath, R.E, 1995. *Mass and Energy Balances*, Michigan Technological University, accessed 18 October 2011, <http://www.cee.mtu.edu/~reh/courses/ce251/251_notes_dir/node3.html>.

[12] Ma, J and Nickerson, J, 2006. Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review, Stevens Institute of Technology.

[13] Paurobally, R 2010. *Control and Mechatronics*, lecture notes distributed in Control and Mechatronics MCTX3421 at The University of Western Australia, Crawley on 1 August 2010.