Development of Murdoch University Pilot Plant Maintenance & Demonstration Programs

“A report submitted to the School of Engineering and Energy, Murdoch University in partial fulfilment of the requirements for the degree of Bachelor of Engineering”

Shane Mackay

Bachelor of Engineering

Department of Engineering and Energy

Supervisor: Associate Professor Graeme Cole

Associate Supervisor: Dr. Linh Vu
Executive Summary

The structure of the Instrumentation and Control Engineering stream at Murdoch University exposes students to significant practical components as well as extensive development of theoretical knowledge. One of the most industry-like practical components that these students have access to is the Murdoch University Pilot Plant. For this reason, it is important that the Pilot Plant is operating smoothly and kept as up to date as possible. The main purpose of this thesis paper is to ultimately improve the performance of the Pilot Plant and hopefully enable it to become an even more valuable resource.

Throughout the time spent on this thesis paper, various improvements and new aspects have been developed and implemented into the Pilot Plant. One of the first ideas explored through this thesis was a significant recoding of the way instrumentation was activated and controlled, enabling all device control function blocks to be operated when the mode attribute is set to program. It is envisaged that this will allow for significantly less operator confusion as has been experienced since the recent upgrade. Further to this, a significant portion of this thesis examined the development and implementation of both maintenance and demonstration programs. Existing research provided evidence that sequential control modules would be the easiest method for coding the maintenance program. This program was designed to activate both upon user demand and automatically on a preset time to cycle various aspects of the Pilot Plant that are susceptible to seizing during extended periods of dormancy. Minimal running time was an important aspect that needed to be considered and this was achieved, with the program completing in two minutes.

The thesis then goes on to explore the development and implementation of the demonstration program. The demonstration program is quite similar to the maintenance program as it also employs sequential control modules. Its design allows for a systematic approach to turning on pumps and waiting until predefined conditions are met to allow the Pilot Plant to reach a steady state operating condition without any input from an operator. This was successfully achieved with a fully automated program from start to finish which it is envisioned will aid during times when tours of the Pilot Plant are being performed, allowing the tour guide to simply press a button and direct full focus to the explanation about the Pilot Plant. Finally this thesis explores the adaption and creation of new human machine interface pages to be implemented into Station to take full advantage of the newly developed code. Three new pages are created to allow easy integration of the new program settings, while minor modifications are also performed, to make operation of the Pilot Plant significantly simpler.

Through the significant research and development undertaken throughout this thesis, it has become evident that there are many more opportunities and directions that further projects could take to build upon this knowledge base and further improve the Murdoch University Pilot Plant and ultimately the Instrumentation and Control Engineering stream.
Acknowledgements

For their enormous assistance in this project thanks are given to the following individuals:

Project Supervisor: Associate Professor Graeme R Cole,
Lecturer, Murdoch University

Assistant Project Supervisor: Dr. Linh Vu
Lecturer, Murdoch University

Mr Will Stirling
Technical Officer, Murdoch University

Mr John Boulton
Technical Officer, Murdoch University

Jolene Mackay
For her continued support
# Table of Contents

Executive Summary ..................................................................................................................... I  
Acknowledgements .................................................................................................................... II
Table of Contents ...................................................................................................................... III
Table of Figures ......................................................................................................................... IV
List of Tables .............................................................................................................................. IV
List of Appendices .................................................................................................................... V
Acronyms .................................................................................................................................... V

## CHAPTER 1: Introduction ........................................................................................................ 1
  1.1. Project Background and History .............................................................................. 1  
  1.2. Project Scope ............................................................................................................ 2
  1.3. Project Objectives ...................................................................................................... 3  
    1.1.1 General Pilot Plant Maintenance ...................................................................... 4
    1.1.2 Fixing Existing Issues ..................................................................................... 4
    1.1.3 Implementation of Control Loops Tuning through Station HMI ..................... 4
    1.1.4 Development and Implementation of Pilot Plant Maintenance Program .... 5
    1.1.5 Development and Implementation of Pilot Plant Demonstration Program ... 5
  1.4. Project Revisions ......................................................................................................... 5
  1.5. Thesis Structure ......................................................................................................... 6

## CHAPTER 2: Technical Review of Experion PKS ................................................................. 7  
  2.1 Analysis of Available Documentation ....................................................................... 7
  2.2 Control Execution Environment – C300 Controller ................................................. 8
  2.3 Configuration Studio .................................................................................................... 9
  2.4 Control Builder ........................................................................................................... 10  
    2.4.1 HMI Display Builder ....................................................................................... 12
    2.4.2 Station ............................................................................................................ 13

## CHAPTER 3: Existing Plant Modifications ......................................................................... 15  
  3.1 Operation of FBs in Experion PKS ............................................................................ 15

## CHAPTER 4: Implementation of Control Loop Tuning ......................................................... 22

## CHAPTER 5: Development of Maintenance Program ......................................................... 31  
  5.1 Sequential Control Module ....................................................................................... 32
  5.2 Maintenance Program Design ................................................................................... 35
  5.3 Maintenance Program Testing .................................................................................. 39

## CHAPTER 6: Development of Demonstration Program ....................................................... 42  
  6.1 Demonstration Program Design ............................................................................... 42
  6.2 Demonstration Program Testing ............................................................................... 47
List of Appendices

Appendix A.10 Thesis (Electronic)
Appendix A.20 Pilot Plant Upgrade CM Export
Appendix A.21 Control Loop Tuning CM Export
Appendix A.22 Maintenance Program Export
Appendix A.23 Demonstration Program Export
Appendix A.30 Maintenance Program 2nd Phase Export
Appendix A.31 Full Pilot Plant Upgrade 2nd Phase CEE Export
Appendix A.32 HMIWeb Display Screen Modifications

Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>Abnormal Situation Management</td>
</tr>
<tr>
<td>CEE</td>
<td>Control Execution Environment</td>
</tr>
<tr>
<td>CM</td>
<td>Control Module</td>
</tr>
<tr>
<td>CVP</td>
<td>Output of PID in percentage</td>
</tr>
<tr>
<td>FB</td>
<td>Function Block</td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
</tr>
<tr>
<td>IO</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IOM</td>
<td>Input/Output Module</td>
</tr>
<tr>
<td>K</td>
<td>Proportional/Gain Term</td>
</tr>
<tr>
<td>L⁻¹</td>
<td>Inverse Laplace Transform</td>
</tr>
<tr>
<td>MEDE</td>
<td>Microsoft Excel Data Exchange</td>
</tr>
<tr>
<td>MV</td>
<td>Manipulated Variable</td>
</tr>
<tr>
<td>OP</td>
<td>Operating Point</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>PKS</td>
<td>Process Knowledge System</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PV</td>
<td>Process Variable</td>
</tr>
<tr>
<td>PVP</td>
<td>Process Variable in Percentage</td>
</tr>
<tr>
<td>S</td>
<td>Laplace Operator</td>
</tr>
<tr>
<td>SCM</td>
<td>Sequential Control Modules</td>
</tr>
<tr>
<td>SP</td>
<td>Set Point</td>
</tr>
<tr>
<td>SPP</td>
<td>Set Point in Percentage</td>
</tr>
<tr>
<td>T₁</td>
<td>Integral Time (minutes)</td>
</tr>
<tr>
<td>T₂</td>
<td>Derivative Time (minutes)</td>
</tr>
<tr>
<td>α</td>
<td>Filter Constant</td>
</tr>
</tbody>
</table>
CHAPTER 1: Introduction

Chapter 1 of this thesis paper will explore the project scope proposed for this thesis as well as the history and background information necessary for an understanding of the project. Furthermore, the layout of the thesis paper will be discussed along with an outline of the various project objectives.

1.1. Project Background and History

In the Instrumentation and Control Engineering specialisation offered at Murdoch University, students are immersed in the implementation and application of numerous controller strategies. In order to build upon the theoretically explored controllers and develop a practical appreciation for the behaviour of these controllers, students are able to gain first-hand experience with such controllers by use of the Instrument and Control laboratory, which contains module type control rooms allowing the connection of different valves and tanks for control analysis during early development, through to contact with an actual Pilot Plant for third and fourth year students, which allows a more in depth experience of different controllers and their performance.

The Murdoch University Pilot Plant has undergone numerous changes since its construction over ten years ago in 1998 which have been carried out predominantly by Control and Thermal Engineering personnel. At commencement of this thesis the most recent significant change to the Pilot Plant was extracted from Earl Hopkinson’s thesis[1] with Murdoch University and Honeywell Australia. Hopkinson[1] was published in 2010 and directly succeeded previous works executed by Punch[2]. With such recent works being performed, it was important to be fully aware of the exact implementations that had been made, the direction in which developments were progressing and which standards had been adopted recently, with emphasis on these new practises necessarily being conformed to where applicable.
At the time of its unveiling the Pilot Plant was controlled using programmable logic controllers and variables speed drives. The initial design was constructed to simulate to a certain degree the major phases in the Bayer process, specifically the digestion, clarification and precipitation stages. The primary purpose of the Pilot Plant is to act as a teaching aid for engineering students. This facilitates the practical application of designed controllers that students can develop and tune through implementation on industrial grade equipment for a small scale plant. In order to keep abreast of the extensive technological advancements that had developed since its construction, it was important that the Pilot Plant be upgraded, which was successfully achieved through the work of the aforementioned Punch[2] and Hopkinson[1].

Since its upgrade the instrumentation on the Pilot Plant has been updated; however the fundamental aspects of instrumentation remain the same, and are tabulated below.

Table 1.1 - Instruments

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Type</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Transmitter</td>
<td>Magnetic</td>
<td>11</td>
</tr>
<tr>
<td>Level Transmitter</td>
<td>Differential</td>
<td>7</td>
</tr>
<tr>
<td>Density Transmitter</td>
<td>Differential</td>
<td>4</td>
</tr>
<tr>
<td>Temperature Transmitter</td>
<td>Resistive Thermal Detective</td>
<td>4</td>
</tr>
<tr>
<td>Pump</td>
<td>Positive Displacement</td>
<td>8</td>
</tr>
<tr>
<td>Ball Mill</td>
<td>3-phase Motor</td>
<td>1</td>
</tr>
<tr>
<td>Flow Control Valves</td>
<td>Pneumatic</td>
<td>8</td>
</tr>
<tr>
<td>Solenoid Valves</td>
<td>24 Volt Direct Current</td>
<td>10</td>
</tr>
<tr>
<td>Pump/Motor Controller</td>
<td>Variable Frequency Drives</td>
<td>9</td>
</tr>
</tbody>
</table>

As the project currently stands following the first roll out of the upgrade, a small number of improvements remain to be implemented, which must be carried out if the Pilot Plant is to reach its maximum potential possible through the utilisation of the Experion PKS platform in terms of education, knowledge and reliability.

1.2. Project Scope

Throughout this project the continuation of development and improvement of the Murdoch University Pilot Plant maintenance and demonstration programs will be undertaken. This upgrading
and constant improvement has been ongoing. This thesis directly follows the results of previous works performed by students enrolled in the Semester One 2011 offering of ENG454 - Industrial Computer Systems Design, as well as Hopkinson[1] and previous to that Punch[2].

The complexity and size of this project requires an enormous degree of research. This is mainly because the technologies that have been implemented are recent developments for industry with minimal knowledge of prior operating and configuration procedures. Murdoch University staff members and students were not overly familiar with simple navigation or operation of the Honeywell Experion PKS software, let alone the configuration and adaption of the software to meet our requirements. With these considerations taken into account, this software limitation presented a massive challenge in itself to understand and will require a large degree of research.

Initially the project will attain the necessary knowledge foundation and then utilise this foundational knowledge in order to competently tackle the prospect of adapting the already-provided code to allow for the inclusion of the maintenance and demonstration programs. The initial stage of the project will also ensure that all relevant standards, including those recently modified, will be included and adhered to. Furthermore, the project will explore the idea of possible expansion as well as the inclusion of the Modbus communication with the air pressure regulator system.

1.3. Project Objectives

Detailed in this section of the report are the numerous objectives that it is hoped will be achieved before the conclusion of this thesis. These objectives are elaborated further in the sections to follow. As has been mentioned previously, this project revolves heavily around not only the Pilot Plant, but also the Honeywell Experion PKS platform, with the ultimate goal of providing a completely operational and functional Pilot Plant that is fitted with the most enhanced features currently
available and that is ready and able to be used as an excellent teaching tool for undergraduate and postgraduate Instrumentation and Control Engineering students.

At the commencement of this project, the Pilot Plant was functioning in a somewhat sub-par state; the basic parts of the Pilot Plant were operating, which enabled student use for control purposes, however various remaining features of the Pilot Plant were not fully functioning or had limited responses.

The primary objective of this project is to develop a maintenance program that will reduce the down time and required trouble shooting that is common during the commencement of each semester from lack of Pilot Plant operation during semester break.

### 1.1.1 General Pilot Plant Maintenance

This objective aims to ensure that the Pilot Plant remains operational throughout the duration of this project. This objective is necessary because successful integration of the maintenance program will require an operational Pilot Plant to ensure everything is functioning and operating as expected. This will typically include aspects such as if a solenoid valve failed during normal operation then this would be replaced or fixed as part of the thesis; yet it will be considered an ongoing maintenance issue and as such will not be documented.

### 1.1.2 Fixing Existing Issues

The Pilot Plant in its current situation has a list of known and unknown issues and problems with the existing wiring or software which affects the operations of the Pilot Plant. In order to successfully implement the maintenance and demonstration program these will first need to be rectified. Some of these changes will implicitly include suitable adaptations of the existing software to facilitate or prepare for these programs to be able to be integrated with the existing operations.

### 1.1.3 Implementation of Control Loops Tuning through Station HMI

This specification will explore adding further functionality to the existing Station Manager HMI. As the HMI stands currently it is not possible to manipulate any of the tuning parameters for the controllers directly. This is one of the changes that should be implemented into the HMI. Implicit
here is also appropriate modifications to facilitate and simplify actuation operations from the main screens.

1.1.4 Development and Implementation of Pilot Plant Maintenance Program
This is the primary objective and will represent a significant portion of the thesis. This section will explore automating the Pilot Plant for a set period of time on a regular schedule to ensure all Pilot Plant equipment is able to move throughout its required limits. Additionally it may include various sequence operations, new screens and checks for correct operation of all the sensors and actuators in the system to more easily identify faulty devices.

1.1.5 Development and Implementation of Pilot Plant Demonstration Program
Similarly the secondary major objective of the thesis is the demonstration program. This section will explore automating the Pilot Plant to eliminate some of the pressure of tour guides. It is envisioned the program will commence upon a click by the tour guide and perform all necessary states to progress the Pilot Plant to a fully functioning steady state, without user intervention. Additionally it may include various sequence operations, new screens and checks for abnormal situations.

1.4. Project Revisions

Since the submission of the project proposal and progress report there has been one significant alteration to the initial project design. The adjustment was initiated after it was realised that final year Instrumentation and Control Engineering students were only using the second half of the Pilot Plant, and as such the initial project design was revised so as to focus the identification and implementation of working solutions to problems mainly on the first unused section of the Pilot Plant, with the remainder of the code to be written but not implemented for fear of possible down time and increased workload on currently enrolled ENG420 students, who may have to update their Microsoft Excel controllers to accommodate new or changed points or parameters.
1.5. Thesis Structure

The remainder of this thesis paper is broken down into seven chapters to provide insight into the works undertaken on this project. The thesis structure is outlined below:

**Chapter Two:** Provides a literary review as well as research into the Honeywell Experion PKS platform and the C300 Controller.

**Chapter Three:** Describes research and improvements into the existing Pilot Plant program including any major concerns or areas for improvement.

**Chapter Four:** Discusses the Implementation of Control Loops in Station HMI

**Chapter Five:** Reports on the implementation and development of the maintenance program.

**Chapter Six:** Describes the implementation and development of the demonstration program.

**Chapter Seven:** Explores the new HMI Development.

**Chapter Eight:** Reiterates the achievements made throughout this project and discusses possible future direction.
CHAPTER 2: Technical Review of Experion PKS

The aim of this chapter is to conduct a documentation review of both the C300 controller and the Experion Process Knowledge System as little was known about either of these upon the commencement of this project. The successful completion of this is extremely important as there is zero room for error both in the implementation of new code and adjustment of existing code. Incorrect changes in the Pilot Plant could result in down time which would adversely affect any students enrolled in ENG420. As the reader is no doubt aware, the Honeywell Experion software and the C300 controller hardware are Honeywell proprietary equipment which places significant limitations on available documentation.

2.1 Analysis of Available Documentation

Due to the proprietary nature of the Honeywell system, the availability of third party sources of information specific and relevant to the Experion release used in the Pilot Plant was limited at best. The realisation of this saw heavy dependence on Honeywell documentation which provided the best source of knowledge for all aspects, including itemising the capabilities of the various features and procedures for configuration and implementation. The Honeywell documentation that was obtainable for this thesis was the documentation relevant to the 2008c release software that is installed in the Pilot Plant and is provided by Honeywell to customers who utilise their platform.

The most useful source of information was the Honeywell Knowledge Builder documentation[3]. This is an electronic library containing all documentation that is relevant to the Experion platform including information on legacy components. Unfortunately, this collection can be difficult to comprehend, and sometimes provides information conveyed in a way that seems directly targeted at Honeywell engineers. In addition to this source, the remaining resources available were the Honeywell training guides[4],[5],[6] that other students and staff had been presented with previously; however, these were not always trouble-free to understand as the accompanying
simulation software that was to be followed in the guide was not available, which meant it was not possible to follow the training manuals to their entirety and for their originally designated purpose. With this in mind third party sources were able to be employed for other non Honeywell specific issues to help with general information gathering and understanding of various systems and components.

2.2 Control Execution Environment – C300 Controller

The C300 controller implemented for the Murdoch University Pilot Plant at its time reflected the most recent controller available from Honeywell. The C300 controller is based on the same C form factor as all the C300 controller modules, Series C Input and Output modules and Series C FIMs. The C300 controller from the Pilot Plant is shown in Figure 2.1, which demonstrates the plastic case with a round faceplate that helps in identifying the module type and model number as well as illustrating the four LEDs to indicate different status situation and alphanumeric display. The unique form factor is to help maximise heat dissipation which is associated with increased reliability. [7],[3]

![Figure 2.1 - Honeywell C300 Controller located in Pilot Plant Control Cubicle](image-url)
The Pilot Plant has the installed Input Output Modules (IOM) as described in the table below.

Table 2.1 - Installed Input Output Modules for Pilot Plant

<table>
<thead>
<tr>
<th>Input Output Module</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Input</td>
<td>2</td>
</tr>
<tr>
<td>Digital Output</td>
<td>2</td>
</tr>
<tr>
<td>Hart Analogue Input</td>
<td>2</td>
</tr>
<tr>
<td>Hart Analogue Output</td>
<td>1</td>
</tr>
</tbody>
</table>

All configurations with the C300 controller are accomplished with the control builder software through assigning a Control Execution Environment (CEE). Thus one of the major benefits of the C300 and Experion PKS platform is that the same software can be used for all configurations.

2.3 Configuration Studio

Configuration Studio forms the backbone from which the Experion system is configured. The majority of the software that is used to configure various parts of the stem is activated from Configuration Studio, after first loading and logging into Configuration Studio. As can be appreciated, this piece of software was used extensively throughout this project. Configuration Studio can be linked to multiple servers to adjust settings; however, when launching Configuration Studio in the control room computers, Configuration Studio is only linked with the Murdoch University Pilot Plant server named PPServer1. With the inclusion of a central location for loading various other environments it is important to have security measures in place so that only people with appropriate knowledge and security clearance can modify system settings that could affect operation of the Pilot Plant. Configuration Studio accounts for this because, before the software will load, it is necessary for the user to input their user name and password. Further to these measures, access to the system has to first be granted by the network administrator so that use is restricted to the appropriate personnel and not to all individuals with a Murdoch username and password[3][4]. A screen shot of how Configuration Studio appears is illustrated in Figure 2.1. The red boxed area indicates the two most utilised features employed for this project through Configuration Studio, namely Control Strategies and Administration of Control Strategies Database.
As illustrated in the screenshot, Configuration Studio provides access to numerous other features such as security access, historical data logging, trend configuration and display configurations.

2.4  Control Builder

Control Builder was perhaps the most utilised aspect of the Experion PKS system for the duration of this project. Control Builder is the programming environment where the control strategies are created then downloaded to the C300 CEE. Its presentation is somewhat similar to National Instruments’ LabVIEW in that it is a graphical object-oriented programming language. However, the significantly less flamboyant block designs and lack of variety gives it a distinct flavour reminiscent of the ladder logic programming language that is generally employed by Programmable Logic Controller manufacturers. Graphical approaches to programming have been known to dramatically reduce the time and exertion required to implement different programs and this provides no exception. [3][5]
As is evident in the screen shot in Figure 2.3, there are three main windows or operating areas within Control Builder; Project Assignment or Project Monitoring (top left), Library (bottom left) and Project (right). The Project Assignment window provides a list or tree schematic of available parameters that are purpose-built for manipulation and programming. In the case of the Pilot Plant, there is only one physical CPM, the C300, which is named CPM_1. There are three other CPMs currently loaded as well; however, these are Simulation C300 controllers and are for experimentation and trialling code and can be deleted at anytime without any effect on the Pilot Plant’s operation. Branching out from the CPM is the CEE which provides allotment for all of the developed CMs that contain the already programmed FB code of the Pilot Plant. On the same level as the CEE, the IOLink provides access to the installed IOMS as mentioned previously in Table 2.1. The Project Monitoring window provides a duplication of the project window, while additionally offering online monitoring of blocks as loaded on the CPM as well as updating of FBs. Therefore, this feature is used when examining how the currently loaded program is operating on the CPM and to change any numeric values or flags.
The Library window is responsible for the collection of currently installed FBs. FBs from this area are able to be utilised in the program through dragging and dropping the required FB into the Project window. Some of the most popular FBs utilised throughout the Pilot Plant block were the numeric and flags; however, all of the blocks that comprise the program must first be transferred from the Library window. One of the blocks that provided the backbone for the maintenance program was the Sequential Control Module, in spite of the fact that it had not been implemented or even tested previously. The Project window is where the majority of the programming is carried out. This window provides the ability to display all the FBs or the program in the selected CMs that are contained within the CPM. As previously stated, the Project window allows the graphical programming language to be implemented and allows for the naming of points and manipulation of parameters.

2.4.1 HMI Display Builder

Contained within the Experion PKS platform are two HMI Display Builder programs. While both ultimately produce similar end results, that is, a human machine interface that an operator or engineer can interact with, the format of these programs and the features they support differs. This project will primarily consider the HMIWeb Display Builder as the HMI pages already created prior to commencement of this project had been built using this program and if the other package was adopted these would have to be rewritten, which given the time constraints, is not viable.

The HMIWeb Display Builder is Honeywell’s proprietary program used in the development of operator screens which directly communicate with the server. These screens allow for vital information to be visually and audibly presented to the operator. An important feature built into the Display Builder is the ability to interpret Abnormal Situation Management (ASM) standards for operator displays. This feature is particularly important since it has been adopted by previous students and compliance will need to be continued.
As the name HMIWeb Display Builder suggests, it is based on the most popular hypertext mark-up language, html. The construction of HMI pages is not dissimilar to other HMI construction suites where preconfigured blocks are inserted into the page and these are configured to read the associated parameter point on the server. A screen shot of the HMIWeb Display Builder with selected Pilot Plant windows is included in Figure 2.4.

![Figure 2.4 - Screen Shot HMIWeb Display Builder](image)

### 2.4.2 Station

Continuing on from the Display Builder, the program titled Station is the application used to display the created operator screens in the client software. For most students studying the Instrumentation and Control major this will be the only application they use to interact with the Experion PKS system. These pages are constructed to allow for the manipulation of valves and actuators as well as the trending and observation of different readings in the Pilot Plant such as temperature and flows.
Station also has a feature that can control when, and by whom, access to the system is granted. This feature is currently employed to limit system use to the required students during Pilot Plant operating hours [6].
CHAPTER 3: Existing Plant Modifications

At the commencement of this project the Pilot Plant was operating in a state of substandard performance. While its operation was satisfactory for control students to manage, many of the features that were meant to be operational had failed either because they were not functioning originally or because incorrect alterations had been made by previous students which had resulted in malfunctioning. One of the main areas of concern, for which there had been numerous reports of malfunction, was the confusion between operator and program in the mode attribute parameter. This issue has been addressed previously in 2010[1]. However, given the amount of confusion it is best to revisit how the feature is utilised.

3.1 Operation of FBs in Experion PKS

Several different types of FBs in the Experion platform have numerous features. One of the features that has been considerably misunderstood since its introduction to Murdoch University is the mode attribute parameter. An example of such a block is the device control FB that is shown in Figure 3.1. Also shown in the figure is the list of different mode attributes that can be selected, namely program, operator, none and normal. The two mode attributes of interest in this project, both of which are the source of most confusion, are the program and operator modes. These modes are pertinent to how the face block interprets its signals from within the CM.

The operator mode attribute setting is linked with Station as it allows for direct operator input. In the situation where the operator mode attribute is selected, the FB ignores selected wired inputs and performs as required by the operator through that particular FBs point detail. In contrast, when the FB is in the program mode attribute a rather opposite situation occurs. When the FB is in program mode attribute, the FB behaves as is instructed from the selected wired inputs. Consider, for example, the device control FB shown in Figure 3.1. If the OPCMD[0] was true and OPCMD[1] was false then the block in program mode would set the output to true depending on the mode attribute.
setting. From a monitoring and operating point of view, there is nothing that the operator can do through Station in the device control FB point detail in program mode attribute to turn the output to false. The FB is operating as it has been programmed through its wired connections. The mode attribute setting was designed for cases where the operator needs to “override” the wired or programmed code to disable the output. In the operator mode attribute the operator can enter the point detail and select the option to enable or disable the device control, irrespective of the on command being sent through the wired FB. In such a situation, these selected wired connections are ignored. The reason only some of the connections are ignored is clearly demonstrated through the Device Control FB.

Consider a situation where the output is the raw water addition solenoid for either of the storage tanks. The block is set to have an active interlock on the SI input so that the block disengages given an event of abnormal plant operation such as, for example, if the tank is overflowing. In this situation, regardless of whether the mode attribute is set to program or operator, the FB will read the wired connection to the interlock. Therefore, if the interlock is true, and the program inputs are set to true and thus the block output should be true, it will remain false. Alternatively, if the mode attribute is set to operator and the output activated through point detail, the input wired connections will be overwritten. However, the wired interlock connections will not be overwritten and the block will not activate the output.

While this seems quite simple, its application in the Pilot Plant caused confusion given that when an operation using Station was controlling the Pilot Plant at the beginning of this project it was necessary to set the device control face block for some outputs to operator to activate them, while other outputs must be set to program to be activated. For example the 24v raw water solenoids had no way of being activated programmatically and needed to be set to operator to activate while the pumps had to be activated programmatically via a tick box. If set to operator mode these would not
activate. This presented significant confusion especially to the Instrumentation and Control engineering students who did not fully understand the reasoning for this. Furthermore this confusion was amplified when advanced controllers were developed in Excel and communicated through the MSDE protocol. When Excel is being used to communicate to different points it is only possible for Excel to control blocks programmatically. For example, if Excel is to trigger the activation of an output it needs to write a true value to a flag which sends to the PV parameter a value of one that is then programmatically wired to the device control face block. As the reader should be aware by now, because this is a wired command, if the FB is in operator mode, this command will be ignored. Even if the student is familiar with the different mode attributes, recognising when various parts of the Pilot Plant are not updating is difficult and time consuming and ultimately an unnecessary burden.

![Figure 3.1 - Devctla Face Block Experion PKS](image)

In order to alleviate this problem, and also improve on implementation and control operation when using Excel, it was decided to make all instruments in the Pilot Plant controllable through the program attribute. This implication required a significant amount of modification to the coding already in place to allow for extra measures of manipulating actuators and valves.
The coding in itself was quite straightforward in that no overly complex blocks needed to be adopted, just the usual FBs of flags and numerics. The main area of concern was correctly labelling and organising the controller execution order for the FBs. It was important that this was correct, otherwise the blocks may fail to activate or worse yet continue to alternate from an activated to inactivated state. The reason for this was a result of how the original code for the pumps had been written. The pumps operate via a checkbox and are basically a circular program in that the input, if activated, goes through a series of FBs before the device control block and then feeds back into the original checkbox.

In addition to the mode attribute concern, the idea of reintroducing the warning buzzer and warning light was explored and ultimately implemented. A photo of the warning light and buzzer that was previously utilised in the Pilot Plant prior to its upgrade is included in Figure 3.3.
By examining the coding that was implemented in the original C300 controller it appeared that the alarm was to operate only in the case of a tank overflowing. However, this code was not successfully implemented and since the new Experion PKS platform has quite an effective way of dealing with alarm management in terms of Station it was decided to vary the predetermined task of the warning light slightly.

Under this new development, the role of the warning light was to primarily allow for a safe initialisation or start-up sequence for the Pilot Plant and secondly to allow for easy recognition to determine if the Pilot Plant was in operation or lying dormant. The warning light and buzzer require a 24 volt power supply, and as some decisions remain to be made regarding the final resting place of the warning light and buzzer, it has only temporarily been connected to a portable variable voltage DC power supply and placed on top of the control cubicle. Once the final resting place has been decided, the warning light and buzzer needs to be relocated and if appropriate wired to one of the 24 volt power supplies in the control cubicle or an external non-variable 24 volt power supply procured and connected. Ideally this new location should be suitably distanced from any office locations, so as to not cause any adverse affects, whilst also being prominently visible for any person working with or around the Pilot Plant operation skid.

Figure 3.3 – Warning Light & Warning Buzzer
The connection of the warning light and buzzer was by no means an easy task. As no existing documentation could be found for the wiring diagram it was necessary to first study the light, buzzer and accompanying relay closely to determine the correct wiring. Incorrect wiring could destroy both the warning light and buzzer or even the connected power supply. For this reason it was decided to test the connection on a portable DC power supply with overload protection to ensure that at least the power supply would not be damaged if the educated connection assumption was incorrect. This was not the case and the warning light and buzzer fired up as expected. From here the warning light then needed to be connected to the C300 controller. This required investigating how the controller was wired up and to properly locate a vacant digital output on the IOLink for both the warning light and buzzer. Once this was accomplished it was necessary to first test the operation of both devices with a simple program that set the device control FB to true. After this was confirmed the development of the CM code could commence.

The need for a start-up sequence arose primarily for two reasons. Firstly it allowed for extra safety measures and secondly its behaviour would more closely mimic that of a real plant. The implementation of the start-up sequence was quite straightforward. After the operation flag is activated, the warning light and buzzer sound for 60 seconds. Within this time it is not possible to manipulate any of the outputs either indirectly through manipulation of PV’s in Excel or directly through manipulation in Station. The only exception to this is if the device control FB has been set to operator mode attribute, in which case the operation setting can be overwritten. The only way to stop this would be to intercept the connection between the device control FB output and the IOLink. While this would solve the problem of not being able to trigger the blocks, it would cause a new problem as it would make it possible for the device to appear activated through the HMI while not physically being activated. After the minute has expired, the maintenance program, which will be discussed later in Chapter 5, will be activated for a further minute and, following its completion, the
buzzer will then deactivate, leaving only the warning light operating and the operator free to manipulate the Pilot Plant as desired.

In terms of safety, the inclusion of the start-up sequence allows for a certain period of warning for anyone working near the Pilot Plant. It is not uncommon for Murdoch Staff and other engineering students to be working on or near the operation skid in the Pilot Plant room. By giving one minute warning, both visually and audibly, it allows all such persons to consider their situation and check their safety, so that they are aware there could be agitators or pumps starting soon or, more seriously, that the steam addition valves could shortly be in operation, and gives them some time to move away or caution themselves if necessary.
CHAPTER 4: Implementation of Control Loop Tuning

The implementation of control loop tuning into Station was another aspect of this project that was considered necessary to improve the Pilot Plant’s performance. Since the upgrade of the Pilot Plant the ability for operators to easily adjust or manipulate some of the controller parameters had been disabled. The ability to manipulate and select either automatic or manual mode and then as appropriate define either a controller set point or operating point functioned correctly post-upgrade. However, the tuning of controller parameters, as was originally designed, used the PID point detail which required an engineer or higher access level. Since Instrumentation and Control students only receive operator access, this impaired functionality and ability to make necessary changes. A screen shot of the originally intended method of controller manipulation is illustrated in Figure 4.1. It is possible to see the general controller tuning parameters familiar to all control oriented personnel such as control action, integral time, derivate time gain options and overall gain.

![Figure 4.1 - Previous PID Point Detail Tuning Parameter Selection][3]

If an operator tried to change any of the tuning parameters on this page, Station would simply not allow, and instead would demand higher security level before allowing any changes. Naturally a
higher security level was available; however it was not customary to give engineering access out to any operator as it had the ability to cause significant coding changes and general mishap with the Pilot Plant. Examples of such misadventures could include the changing of state in SCMs and changing of control strategies amongst other things like deleting of CEE and IOMs, and the list could go on. So since the access level needed to remain unchanged from operator, it was necessary to implement an alternative. This was easily achieved through the introduction of a new HMI page specifically for control loop tuning which is discussed in Chapter 7. For the HMI to function, however, it was necessary to make some modifications to the coding for each PID FB.

The list of tuning parameters and extra features available in the PID point detail were quite extensive and far more than what is required for fundamental understanding of control theory. As such only the more common features were introduced for manipulation namely overall gain, integral time, derivative time and control equation. It was decided to disregard the controller action parameter as the currently developed code does not allow for selection of MV and PV. It is envisioned that this feature will be implemented by future students. When such code is developed and introduced, it will be quite easy to expand on the code developed here to allow for this extra parameter. A sample of the code implemented for the FP_REF_141 pump PID is shown in Figure 4.2.
As the PID requires engineering access level to make changes to the FB, it was necessary to introduce external mode updates. As is evident in the sample code, four numeric and one typeconvert FBs were introduced into the code for each PID CM and connected to the appropriate block pins of the PID FB. Since the numeric blocks can be modified by an operator, it is possible for an operator to select the value as required and as the PID is now updated externally or programmatically the block pins and does not question the level of security thus automatically updates. The numeric FBs of overall gain, integral time and derivative time were able to be directly connected to the PID FB since the required inputs were PVs whereas the Control Equation or CTLEQN required an enumeration type input which made it necessary to place a Typeconvert FB in between the connection so that an input of 0 would be treated as equation A and similarly a 1 would be treated as equation B and so on and so forth. A table of the enumeration for the different controller equations is presented in Table 4.1.
The exploration of these control algorithms is beyond the scope of this thesis. For convenience a small tabulated summary of the major difference of these different control algorithms is provided below in Table 4.2 since it is now possible to confirm the controller algorithm that is being employed. However, this information will not be explored in any further detail.

### Table 4.1 - Control Equation Type Enumeration

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Control Equation Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Equation A</td>
</tr>
<tr>
<td>1</td>
<td>Equation B</td>
</tr>
<tr>
<td>2</td>
<td>Equation C</td>
</tr>
<tr>
<td>3</td>
<td>Equation D</td>
</tr>
<tr>
<td>4</td>
<td>Equation E</td>
</tr>
</tbody>
</table>

### Table 4.2 - Control Algorithm Types[3][1]

<table>
<thead>
<tr>
<th>Control Type</th>
<th>Main Feature</th>
<th>Controller Algorithm</th>
</tr>
</thead>
</table>
| Equation A   | Proportional, integral and derivate terms all have an impact on the error    | \[ CVP = K \times L^{-1} \left[ \left( 1 + \frac{1}{T_1 s} + \frac{T_2 s}{1 + \alpha \times T_2 s} \right) \right. \]
|              |                                                                             | \[ \times (PVP_s - SPP_s) \] |
| Equation B   | Derivative term has impact on variations in process variable while proportional and integral terms have impact on error. This equation helps to reduce derivative spikes as normally associated with fast set point changes. | \[ CVP = K \times L^{-1} \left[ \left( 1 + \frac{1}{T_1 s} + \frac{T_2 s}{1 + \alpha \times T_2 s} \right) \right. \]
<p>|              |                                                                             | [ \times PVP_s - \left( 1 + \frac{1}{T_1 s} \right) \times SPP_s ] |</p>
<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Proportional and derivative terms impact on variations in process variable while integral term impacts on the error. Implementation of this equation provides a longer settling time and low controller aggression to set point changes.</td>
<td>[ CVP = K \times L^{-1} \left[ \left(1 + \frac{1}{T_1s} + \frac{T_2s}{1 + \alpha \times T_2s} \right) \times PVPs - \left(\frac{1}{T_1s}\right) \times SPPs \right] ]</td>
</tr>
<tr>
<td>D</td>
<td>Integral only control is performed by this equation.</td>
<td>[ CVP = L^{-1} \left[ \left(\frac{1}{T_1s}\right) \times (PVPs - SPPs) \right] ]</td>
</tr>
<tr>
<td>E</td>
<td>Proportional only control is performed by this equation.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

To enable operators to interact with the newly developed code it was necessary to implement some changes to the HMI. A lot of different ideas were explored in terms of how to best enable the manipulation of the implemented controllers. However the best method was through the design of a new HMI page purposely designed for the manipulation of the controller parameters. A screenshot of this page is illustrated in Figure 4.3.
The new control loop tuning parameters HMI emphasises the amount of extra code that had to be written to implement these changes. Given there are eight pumps, four flow control valves and three temperature control valves, each with their own PID FB, this equated to an additional sixty numeric FBs that needed to be inserted and connected to the appropriate PID block pins along with 15 typeconvert FBs that all needed to be appropriately wired. As explained earlier the HMI connects directly to the newly created FBs as is further illustrated below.
In order to be fully confident the performed modifications were successful and actually worked it was necessary to perform an extensive array of tests. It was decided to test the gain and integral terms on both the raw water addition FCV for level control of the non-linear tank as well as the cyclone underflow pump for flow control into the ball mill tank. Both of these control loops are better illustrated in Figure 4.5 and Figure 4.6, with the corresponding tuning parameters highlighted in red and blue respectfully in Figure 4.3.
The cyclone underflow tank pump control loop that is highlighted in red was the first test performed since the implementation of a new HMI screen. The first test was to fully ensure that the automatic mode PID controller was functioning correctly before any settings were modified. The controller gain was originally set to an overall gain of one and integral of three as is boxed in red in Figure 4.3. With these settings it was possible to change the SP to ten and the flow through flow transmitter FT_347 would sit at approximately ten as would be expected. Similarly if an SP of twenty was commanded the PID would manipulate the OP to reach those new conditions. The controller was extremely conservative in achieving them. However, it was possible to conclude the PID controller was definitely working. Following this the SP was returned to ten and the overall gain was changed to three, then the SP changed to twenty as per before. This time the controller was much more aggressive and the PV actually had some oscillations to it. Similarly the Integral time was adjusted to confirm it was working and the effects it produced where what was expected so it was necessary to focus on the next control loop to confirm its operation.
The raw water addition control loop was tested in a similar fashion to the cyclone underflow control loop with the main difference being the PV for the first test was the flow rate where as the PV for this test is the level in the non-linear tank or more specifically LT_542. Again the controller was setup to automatic mode with the original parameters of overall gain of eighteen and integral time of four and derivative time of zero. With these settings an SP of fifty was set and the OP responded extremely quickly to drop the tank level to fifty which was then maintained. The SP was then changed to sixty and the results observed were the same. Next the gain was reduced to one and the SP changed; as expected the controller was much slower to respond and the time taken to reach SP was increased enormously. Similarly the integral time was adjusted and the effect was as expected. Therefore it was possible to conclude overall that the introduction of control loop tuning has been successfully implemented for operators that do not have engineer security level access.
CHAPTER 5: Development of Maintenance Program

As discussed previously, the primary reason for implementing a maintenance program is due to long periods of inactivity in the Pilot Plant. While it is not specifically the period of inactivity that causes the problem, it is what this inactivity causes when the system needs to be operated again. It is well known that any simple machine or device, such as a valve, will have a tendency to become fixed in its current position/orientation if left un-operated/not cycled for an extended period of time. The Pilot Plant is no exception. During semester transitions and non-teaching weeks, the valves and actuators in the Pilot Plant can lay dormant for over twelve weeks at a time. This is more than enough time for the valves to seize up. As a result of this extended dormancy, when semester commences and Instrumentation and Control students require a fully operating Pilot Plant, a great deal of time and effort must be consumed in attempting to cycle all the valves and fix any problems with the Pilot Plant. As is usually the case the first operator will request an operation to be performed and the device fails to operate. This will usually provide an abnormal situation or alarm on the operators Station HMI. However, the presence of an alarm is by no means a solution to the problem as the operator may not always able to fix the problem and the solution may not necessarily be simple or cheap.

The purpose of the maintenance program is to prevent such problems from occurring. The maintenance program has been targeted specifically at the cycling of three way valves, actuators and pneumatic valves, as these seem increasingly susceptible to seizing. Ideally the operation of all instrumentation, namely the pumps and ball mill, would have been included. However, their inclusion would not only have required the raw water to have been connected but would have dramatically increased the minimum running time of the maintenance program from about one minute to just shy of half an hour. At this stage the running of the maintenance program has been designed around draining all of the water from the pilot plant tanks at the end of each semester and turning off the raw feeds leaving only the compressed air. If it is so desired to not drain the Pilot
Plant completely of water later on then it is envisioned that this program will have paved the way to development of a maintenance program that includes the ability to circulate the dormant water if this feature is required. This situation is explained in further detail in Chapter 6. The implementation of such a system was not easy because it required an enormous understanding of how the original Experion code was programmed. This section of the project also bought about the first truly new method of coding to be implemented into the Murdoch University Pilot Plant.

The idea of SCMs to the Pilot Plant had not been considered or implemented in any form at Murdoch University at the commencement of this project. The idea of using such a feature arose from logically addressing how the maintenance program could be written using traditional CMs. It was almost immediately apparent that there must be a better method, leaving further research to be implemented through VBA programming or SCMs. Since SCMs provide greater feedback in terms of monitoring and is a feature that Honeywell promotes, it was decided to proceed along this path. Before programming could commence on the SCMs for the maintenance program it was important to fully understand all aspects of the SCMs before their implementation.

### 5.1 Sequential Control Module

Sequential control modules are a standard feature provided by the Experion software. It is a section of the Honeywell system that is tailored towards the simplification and implementation of batch logic. In their simplest form SCMs allow for the easy implementation of state machine design with a logical progression of different steps and transitions as required. Further to this, the SCMs also provide abnormal situation handling where it is possible to call alternative sequence execution as specified. These abnormal handlers are typically named checking, interrupt, restart, hold, stop and abort with the actual code being written in the main handler. Each handler is able to possess one invoke step which if true will initiate and commence the running of that handler. Shown in Figure 5.1 is the typical minimum conditions found in a handler [7].
In Figure 5.1 the invoke can be seen as the first step that will activate the SCM or handler if that condition is true. This is then usually followed by a step where the required action is performed, which is in turn followed by a transition where the handler waits to make sure the previous step has been performed before moving on and reporting a fault if a preset time has been exceeded. Finally the SCM is reinitialised back to the required state. If the SCM is running the main handler, the initialisation will usually call the stopping handler, but if the SCM is running the stopping handler, then the SCM will reinitialise back to the idle state. Generally the handler will have a significant number of steps and transitions, but will still contain these four basic properties. Figure 5.2 provides a sample of how the SCM is presented to the engineer through programming along with the available FBs.
Since the new modifications to the Pilot Plant included a small warning period before the plant could be operated it was decided to also include the operation of the maintenance program after this warning period. Since this was the case it was important to keep the running of the maintenance program as short as possible so as to not inconvenience or penalise the initial operation of the Pilot Plant heavily. The maintenance program, as it is currently coded, runs for sixty seconds and cycles actuators from fully-open to fully-closed approximately twenty times. This should be adequate to ensure there is no seizing of instrumentation if this is run once a day; however, the duration of the maintenance program can be easily increased if the original sixty seconds is not sufficient.

Furthermore, as safety is a very importance aspect at Murdoch University, it is important that when the maintenance program is automatically scheduled to start that the warning light and buzzer are also alarmed to warn any personnel in the room that the Pilot Plant could commence operation.
shortly. Correspondingly the design of the maintenance program has taken into consideration the safety of leaving raw material connections such as pressurised air, water and steam turned on to the Pilot Plant around the clock and it was decided that this is not desirable. Consequently, the maintenance program has been designed to operate a dry plant, in that solenoids are operated without a medium travelling through them with the exception of air in certain cases which does not present any problems to the instrumentation and means that the SCM can be run with only the compressed air supply connected and the other two raw connections to the Pilot Plant left disconnected.

5.2 Maintenance Program Design

The maintenance program is designed to invoke the main handler when any of three conditions are met. The first possible call for invoke is when the user has initiated operation of the Pilot Plant by setting the operation.user_plant.pvfl flag to one. This is the main tag that needs to be set if the operator is to be able to run the Pilot Plant and will trigger the main handler. The second method for activating the maintenance program is if the hour and minute time of the CPM corresponds with the set time in the SCM which is currently 0800 hours. Finally the last option to invoke the handler is through activating the maintenance flag, which can be activated while the Pilot Plant is operating, if the user feels a valve or instrument may be seized. The sample code for the invoke for the main handler is demonstrated below in Figure 5.3. As this SCM is being viewed in monitor mode, the areas highlighted in green are the rungs of the transition condition that are true and so it is easily possible to identify which extra conditions need to be meet for the maintenance program to commence.
The operation of the maintenance program is not overly complicated. It fundamentally adopts a batch type arrangement and systematically cycles through the opening and closing of different valves. A small sample of the maintenance program SCM for the first half of the Pilot Plant is provided in Figure 5.4. Once the handler has commenced operation the SCM works systematically through the Pilot Plant, cycling all the valves and solenoids in groups.
When the maintenance program has successfully completed and all of the function blocks have been returned to program mode the buzzer will cease to operate and the operator will be presented with the message ‘maintenance program completed successfully.’ It is important to note that the operator will be presented with the same message even if the maintenance program does not in fact run successfully. Take for example the situation when the raw air supply is not turned on to the Pilot Plant. The maintenance program will run and the dc solenoids will switch. However the three way valves and pneumatic valve will not move since they require compressed air. In this situation warning alarms should go off for the three way valves; however, a lack of feedback for the pneumatic valves...
will mean no alarms will be set. Consideration was given to aborting the maintenance program should the air supply not be turned on. However the air being disconnected poses no risk of damage to the Pilot Plant and does allow the maintenance program to run the dc solenoids so they still receive a cycling, making this advantageous. The only situation where the message would not be communicated to the operator is if the abort handler is invoked. Currently, given the transition conditions this would only be possible if the abort timer exceeded the maintenance timer which could only result if the PV for either timer was manipulated through control builder or Microsoft Excel.

In order to allow for the setting of the time and automatic maintenance mode flag, a new HMI was developed that allowed for manipulation of these parameters. A screen shot of the new HMI is included below with the new additions highlighted in red. Here it is possible to directly start the maintenance program and set the duration of the program to run if a longer duration is required. This can be increased until a maximum of 8000 seconds or approximately 2 hours 13 minutes. The hour and minute numeric boxes can also be used to adjust when the SCM will automatically operate. Currently the boxes are set with operator privilege because it is not fully implemented; however once they are fully implemented the required property to edit them can be set to engineer level so that only appropriate operators can manipulate the settings. For example if the hour is set to 8 and the minute is set to 4, as is shown below, the SCM will trigger, provided automatic maintenance is also checked at 8:04am every business day.
5.3 Maintenance Program Testing

The testing of the maintenance program was accomplished by loading up the appropriate HMI in station and then activating the appropriate parameters such as maintenance start and user program on to ensure the maintenance SCM commenced. When the SCM was running it was then necessary to ensure the desired tasks were achieved. This was easiest to confirm by physically observing the HMI for changes. Some of the changes that were observed during this period are included below to confirm what was coded was actually working.
Figure 5.6 - Maintenance Program Testing through Storage Tank and Ball Mill Tank HMI
The results below were observed as the maintenance program ran for the set duration and it was possible to systematically watch all the solenoid valves, 3WV and pneumatic valves being cycled as per Table 5.1. The last six steps were repeated until the expiration of the maintenance timer which then set the plant operation inactive and closed or zeroed all valves and solenoids.

Table 5.1 - Maintenance Program Testing Observation

<table>
<thead>
<tr>
<th>Instrument/Parameter</th>
<th>Operation</th>
<th>Reference Figure 5.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation Checkbox</td>
<td>Set Active</td>
<td></td>
</tr>
<tr>
<td>24V DC solenoids</td>
<td>Set Open</td>
<td></td>
</tr>
<tr>
<td>3 way Position Valve</td>
<td>Set Position 1</td>
<td></td>
</tr>
<tr>
<td>Pneumatic Valve</td>
<td>Set 100%</td>
<td></td>
</tr>
<tr>
<td>24V DC solenoids</td>
<td>Set Closed</td>
<td></td>
</tr>
<tr>
<td>3 way Position Valve</td>
<td>Set Position 2</td>
<td></td>
</tr>
<tr>
<td>Pneumatic Valve</td>
<td>Set 0%</td>
<td></td>
</tr>
</tbody>
</table>

This test was performed for all of the methods for activating the maintenance program and the tests were passed. As already mentioned, this thesis only explores the operation of the maintenance program developed for the first half of the Pilot Plant since its operation has been confirmed. The code for the first half of the Pilot Plant is included in appendix A.21, while the full code which has not been implemented or tested at the writing of this paper is included in appendix A.30.
CHAPTER 6: Development of Demonstration Program

The fundamentals of the demonstration program were actually developed originally for the maintenance program. Originally the idea was to have the Pilot Plant operate in steady state conditions with the valves being exercised during this operation. However this idea was abandoned early on after first basic implementation when completion times of 20 to 35 minutes were recorded depending on tank levels. Time factor aside, adopting this steady state maintenance program would have had the advantages of allowing the pumps and motors to run thus ensuring that there was no seizing of bearings. The advantages for the adoption of such a system however seem to be significantly outweighed by disadvantages. Primarily the main disadvantage would be that the supply water would need to also be continually connected to the Pilot Plant for automatic maintenance. Furthermore there is an increased risk of water spillage and leakage especially given the numerous manual switches which could be set to block the output from a pump or even set open to drain a tank. Depending on the location of the manual valve such an event may not even be detectable by the Experion system given the limited amount of instrumentation and could in fact result in significant water spillage especially given the lengthy duration of the program.

6.1 Demonstration Program Design

As already alluded to, the demonstration program utilises SCM in the same manner as the maintenance program. It also contains three handlers, namely the main, stop and abort handlers. Activation of the main handler is only possible through positive flagging of the demonstration flag FB to a value of one while the Pilot Plant is not in operation. This is confirmed through Figure 6.1 below which shows the main handler invoke.
Upon commencement of the main handler, the program systematically runs through the Pilot Plant, setting up controllers to automatic and waiting until appropriate levels are reached before activating pumps.

A simple way to illustrate this is to run through a section of the SCM code as is included in Figure 6.2. Here it is possible to notice the start of the demonstration SCM just after the initialisation and Pilot Plant setup steps and transitions. Running parallel to the SCM code is the appropriate HMIs which help to demonstrate what the program is doing through to how an operator could achieve the same task. While the program works very similar to interacting with the HMI, the SCM does it through manipulation of different FBs not always accessible via Station, so it is not truly possible to replicate with the HMI only which is why some commands such as turning on the feed pump then turning it off again seem pointless. However this is necessary given the extra programming.
Figure 6.2 - Selection of Demonstration SCM with corresponding HMI equivalent
When the maintenance program first commences any steps that contain a device control FB, it is first necessary to place that block in program mode attribute, as is the case with the raw water solenoids. This step is necessary to allow for the case where an operator may have previously set the FB to operator mode attribute and is performed for each device control FB and as such will not be mentioned any further. Each of the raw water valves are placed into automatic mode and the outputs activated. From here the next transition is possible once either of the storage tanks have a level greater than five percent. When this condition is met the OP to the feed pump is set to seventy five percent and placed in manual mode. The pump is then switched on. From here the transition waits for the ball mill tank to reach a level greater than twenty percent and ensures that the feed pump remains active. When this condition is met, the ball mill pump SP is set to eighty percent, the PID is set to auto and the ball mill pump is switched on. Following this, both the ball mill and ball mill tank agitator are set to automatic mode. Similar procedures are then carried out on the remainder of the Pilot Plant, up until the lamella underflow pump. The exact instructions can be easily obtained through examining the complete stage one demonstration SCM code available in Appendix A.21.

The duration of the demonstration is easily set through the HMI, which allows entering the time in seconds. This setting however does not take into account the time required to enter the demonstration program into a steady state mode and simply activates the stop handler at the expiration of this timer. For example if the storage tanks are both at zero percent capacity and the demonstration timer is set to ten minutes, the demonstration will finish well before the storage tanks have reached five percent and a pump is even activated. The only other way the stop handler is activated is if the demonstration flag is turned off through the HMI while the program is operating. The HMI additions used for manipulation of the demonstration start and demonstration duration timer is shown below in Figure 6.3.
Finally, the abort handler can be invoked by the expiration of a one minute timer after the demonstration timer expiration or if a tank level exceeds ninety percent once the steady state flag has been tripped. This provides a certain degree of safety in the Pilot Plant; however the tour guide does need to have some understanding of the Pilot Plant as there are some conditions in which the demonstration mode could result in water on the operation skid. Both of the invokes for the abort and stop handler are illustrated in Figure 6.4 and Figure 6.5 respectively.
Figure 6.5 - Demonstration Program Stop Handler Invoke

6.2 Demonstration Program Testing

The demonstration program was tested in much the same methodology as the maintenance program. The testing required the observation that everything was being performed as expected. This was mostly performed by activating the main handler then observing the operation of the HMI. Firstly the Pilot Plant was setup so that all tanks for the first and second half of the Pilot Plant were around 50% full with the exception of the storage tanks which were around 90% full. The demo SCM was then set to running and the demonstration program set to work. It activated the plant operation flag, where the warning light and buzzer immediately sprung to life. After the warning period the raw water solenoids were set to active and feed pump set to manual with OP of 75 and then turned on. The ball mill pump was then set active almost immediately as the level in the ball mill tank was just greater than 50% while the agitator and ball mill also commenced operation. The remainder of the Pilot Plant also worked exactly as was expected given the conditions and continued to run until the expiration of the demonstration timer, where everything was shut off correctly.

For the second test of the demonstration program it was decided to see if it would work with each of the vessels completely dry. For this test each of the manual drains for all of the tanks were opened and finally closed when the tanks were completely dry and the level transmitters were reading NaN as the height was well below the zero range for the transmitter. The SCM was then started with the duration increased to 8000 this time. After the start-up sequence completion the raw water
solenoids activated immediately; however it took approximately thirty minutes before the storage tanks were above 5% full. After they were above the trigger point the program ran exactly as was expected with the understanding each new transition that contained a tank, such as the ball mill or cyclone underflow to be above 50% to proceed, required a significant amount of time since the vessels were empty.

The third test was not only to test the operation of the demonstration handler but also to ensure the abort handler operated correctly should it ever be required. For the testing of the demonstration handler this time all of the device control blocks were set to operator mode. While this should never be the case with the new coding of the Pilot Plant it accounts for the possibility. The demonstration program was commenced and after the start-up period the program operated successfully, correctly switching all FBs back to program mode attribute before they were activated. It was approximately two minutes after this that the demonstration program had the Pilot Plant operating in a steady state condition with all controllers active and the steady state flag activated. With this state achieved it was possible to test the abort handler. To do this the pump from the ball mill tank (BMP_241) was turned off. This quickly saw the level of the tank rise up from 80% capacity. Furthermore the OP of the CUFP also started to decrease almost immediately as there was no longer any flow into the cyclone underflow tank. The level in the ball mill tank continued to increase until the level transmitter recorded a value above 90% which was approximately one minute later. At this time the whole Pilot Plant shut down and every pump and valve was closed and no overflowing of any vessel occurred.

Through the running of these three significantly different tests it was evident that the demonstration program had been successfully implemented. In all three demonstration tests there had been no vessel overflowing or times where the program was implemented and it did not perform its designed task. The demonstration program allowed for the blocks to have been set in operator mode and the
program was able to change them back before adjusting SPs and OPs for different devices. The program also allows for significant freedom of the operator by allowing them to disable pumps without the abort handler being called; however if a tank level rises dangerously high then the abort sequence will take over and correctly shutdown the Pilot Plant.
CHAPTER 7: Development of HMI

The development of the new HMI screens carries on directly from work previously performed by both Punch and Hopkinson[2],[1]. As the HMI Screens had been developed to conform to certain display guidelines, it was necessary to first research the guidelines to gain an appreciation of them and to ensure compliance, both in terms of the new additional information being added to the HMI as well as ensuring compliance with some of the already designed HMI screens.

Punch[1] introduced and implemented the idea of a hierarchy into the Pilot Plant HMI. As such it was necessary to ensure the new screens that were modified provided the same features they were originally designed for as well as conforming to the hierarchy. Punch[1] created eight Pilot Plant HMI displays that were designed to conform to the ACM guidelines. His work consisted of one level 1 screen, six level 2 screens and finally one level 4 screen. The extra work performed to the coding and development of both the maintenance and demonstration programs required upgrades to each of the HMI pages along with the following additional HMI pages:

- 1x Level 4 HMI titled “Control Loop Tuning Parameters”
- 1x Level 4 HMI titled “Program Settings”
- 1x Level 4 HMI titled “Auto/Manual Settings”

A screen shot of the newly developed control loop tuning parameters HMI is shown in Figure 7.1. The control loop tuning parameters are now much more accessible and adjusting them only requires the operator to enter the required value into the appropriate text box. The control equation is also shown in the form of a drop down combo box as is illustrated in Figure 7.1.
The existing HMIs needed to have a number of checkboxes inserted in addition to the new warning light and warning buzzer device control box which was added to the top right of each HMI screen. A screen shot of the new features are illustrated in Figure 7.2. These new additions were included for every level two HMI.

The new addition allowed for straightforward operation of the Pilot Plant since the new checkbox was required to be checked for any of the valves or pumps to be controlled. It also allowed for the
operator to check what stage the Pilot Plant was up to in terms of the operation of the warning light and warning buzzer. As is the case with the original HMI designs, when a device or output is active, that unit turns a dark grey; this is the situation that occurs when either the light or buzzer are active.

Further to the addition of the operation box, an array of checkboxes and combo boxes were installed on the relevant level 2 HMIs to allow for the extra code introduced earlier and to take advantage of the fact that the Pilot Plant could be operating in program mode. A screenshot of both the old and new HMIs has been included below to provide a comparison of the new additions made to each HMI page.

Figure 7.3 - Level 2 "Storage Tanks" Old HMI Design
Clearly evident in Figure 7.4 are the various extra controls placed next to each actuator and three way valve. These new additions not only allow for the appropriate selection of manual control or automatic control from of a drop down combination box, but also for the ability to control the entire HMI page through the program mode attribute. This brings the added benefit of reduced operator confusion as well as a reduction in the number of clicks required to activate solenoids and valves. Consider for example the previous HMI shown in Figure 7.3. If the user wished to activate the compressed air solenoid to storage tank two, the user would first need to click on the compressed air solenoid, which would then bring up that device control point detail parameter. Following this, the operator would need to ensure operator was selected in the mode attribute and then click on open or close in the OP selector. As is evident, the new HMI, by allowing the user to simply check or
uncheck the check box next to the associated device, makes for a much easier activation/deactivation process. It also allows for the user to logically apply this process to the remainder of the Pilot Plant, as the process for every unit is the same as it was for the pumps.
CHAPTER 8: Concluding Remarks

This chapter examines the work performed during the entire thesis project. The major objectives of the project are revisited and concluded. Finally, any further works to be completed and the possible direction of any future undertakings are recommended to help further develop the Murdoch University Pilot Plant.

8.1 Project Conclusions

During this thesis a research of the Honeywell Process Knowledge System Platform and C300 controller has been undertaken. This research provided the backbone for this thesis paper in terms of the required knowledge to enable an understanding of how the Pilot Plant operated and how to edit the existing code to implement the necessary changes. The first major task performed was the reintroduction of the warning light and warning buzzer. This allowed for a visual and audible warning so that anyone working near the operation skid would be made aware that the Pilot Plant could commence operation shortly and to move away. This also made it simple to identify if the Pilot Plant was in operation.

Following this was the implementation of code that allowed for straightforward alteration of control loop tuning parameters. The previous method required engineering security and required the operator to know how to enter the PID point detail parameter. The new method allows for changing of the PID parameters overall gain, integral time and derivative time adjustment programmatically with only operator access level. Further to this was the introduction of SCMs into the Pilot Plant with the maintenance program. A maintenance SCM was written with three handlers to systematically cycle through different actuators and valves of the Pilot Plant. Its main purpose is to reduce the down time of the Pilot Plant during periods of dormancy and it is set to operate at 0800 hours every weekday morning for one minute allowing for approximately twenty cycles of valves and actuators.
Similarly to the maintenance program, the demonstration program was built through a SCM with three handlers. The primary purpose of the demonstration program will be to make it possible for tour guides to easily demonstrate an operating Pilot Plant and run through some of its features. For this reason it needs to be fully automated to run at the press of a button and to cease any operations correctly and safely if any abnormal situations are encountered. Finally the new HMIIs were developed to fully support both the coding changes and new SCMs introduced into the Pilot Plant. Three new level HMI pages were introduced with one dedicated to controller loop tuning, one for auto/manual activation limits and one to house the program settings for both of the SCMs created.

Throughout this thesis, various documents have been updated and already-produced guidelines have been conformed to, including the ASM guidelines and Pilot Plant coding guidelines. Ultimately this thesis has achieved the revised project goals. Originally it was desired to have all of the objectives implemented and tested on the whole Pilot Plant. However this has only been achieved on the first half of the Pilot Plant which was not in use by currently enrolled ENG420 students. The coding for the second stage has been completed; however it has not been implemented or tested.

### 8.2 Proposed Future Works

This thesis has only begun to explore the large array extra features that can be implemented into the Murdoch University Pilot Plant. A number of proposed further works are summarised below which are recommended tasks that could help further the project in the future. While some are obvious and quite easy to implement others may require significant research.

- Deployment and testing of phase one HMI screens - These have only been tested on the server and need to be rolled out to all licensed Station clients.
- Phase two roll out - The code has been developed for the second half of the Pilot Plant; however it has not been implemented or tested.
- Investigation into Modbus communication with the air compressor – This will enable the commencement of the maintenance and demonstration programs to be more
sensitive to conditions; neither of these should operate if there is insufficient or low compressor pressure.

- Trending for the maintenance or demonstration program – This would allow for the possibility of detecting when any instrumentation needs re-calibrating or if any valves may be showing signs of failing.
Bibliography


## Appendices

Specific to the nature and quantity of the appendices a DVD has been supplied which encapsulates the following documentation, however some are also printed where appropriate.

<table>
<thead>
<tr>
<th>Appendix Number</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>DOCUMENTATION</strong></td>
</tr>
<tr>
<td>A.10</td>
<td>Thesis Electronic</td>
</tr>
<tr>
<td></td>
<td><strong>SOFTWARE DEVELOPMENTS 1st Phase</strong></td>
</tr>
<tr>
<td>A.20</td>
<td>Pilot Plant Improvement Upgrade CM Export</td>
</tr>
<tr>
<td>A.21</td>
<td>Control Loop Tuning CM Export</td>
</tr>
<tr>
<td>A.22</td>
<td>Maintenance Program Export</td>
</tr>
<tr>
<td>A.23</td>
<td>Demonstration Program Export (Printed)</td>
</tr>
<tr>
<td></td>
<td><strong>SOFTWARE DEVELOPMENTS 2nd Phase</strong></td>
</tr>
<tr>
<td>A.30</td>
<td>Maintenance Program Export (Printed)</td>
</tr>
<tr>
<td>A.31</td>
<td>Full Pilot Plant Upgrade CEE Export</td>
</tr>
<tr>
<td>A.32</td>
<td>HMIWeb Display Screen Modifications (Printed)</td>
</tr>
</tbody>
</table>
Appendix A.23 Demonstration Program SCM
### Appendix A.23

#### Transition Conditions and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demonstration Start</td>
<td>OPERATION.DEMONSTRATION.PVFL = 1;</td>
</tr>
<tr>
<td>2</td>
<td>Plant not in Operation</td>
<td>OPERATION.PLANT_OPERATION.pvfl=0</td>
</tr>
</tbody>
</table>

```
P1:And
P2:None
P3:None
```

#### Step Outputs and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set plant PLANT_USER_FLAG ON</td>
<td>OPERATION.PLANT_USER_ON.PVFL := 1</td>
</tr>
<tr>
<td>2</td>
<td>Set plant PLANT_PROG_FLAG ON</td>
<td>OPERATION.PLANT_PROG_ON.PVFL := 1</td>
</tr>
<tr>
<td>3</td>
<td>Reset Maintenance Flag</td>
<td>MAINTENANCE.Maint_SS.pvfl := 0</td>
</tr>
<tr>
<td>4</td>
<td>Set Maintenance Timer ON FLAG</td>
<td>MAINTENANCE.DEMO_TIMER_ON.pvfl := 1</td>
</tr>
<tr>
<td>5</td>
<td>Set Maintenance Buzzer ON</td>
<td>MAINTENANCE.WARN_BUZZ_ON.pvfl := 1</td>
</tr>
</tbody>
</table>

```
P1:And
P2:None
```

#### Transition Conditions and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Plant Ready Flag ON</td>
<td>OPERATION.PLANT_RDY.PVFL := 1</td>
</tr>
</tbody>
</table>

```
P1:And
P2:None
P3:None
```

#### Step Outputs and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raw Water SV_104 to Program Mode</td>
<td>SV_104.DEVCTL_MODEATTR := 2</td>
</tr>
<tr>
<td>2</td>
<td>Raw Water SV_124 to Program Mode</td>
<td>SV_124.DEVCTL_MODEATTR := 2</td>
</tr>
<tr>
<td>3</td>
<td>Set SV104 to Auto</td>
<td>SV_104.AUTO.pvfl := 1</td>
</tr>
<tr>
<td>4</td>
<td>Set SV124 to Auto</td>
<td>SV_124.AUTO.pvfl := 1</td>
</tr>
<tr>
<td>5</td>
<td>Set Maintenance Auto ON SV104</td>
<td>MAINTENANCE.SV_104.auto.pvfl := 1</td>
</tr>
<tr>
<td>6</td>
<td>Set Maintenance Auto ON SV124</td>
<td>MAINTENANCE.SV_124.auto.pvfl := 1</td>
</tr>
<tr>
<td>7</td>
<td>Set Maint SV104 OFF since Auto</td>
<td>MAINTENANCE.SV_104.ON.pvfl := 0</td>
</tr>
<tr>
<td>8</td>
<td>Set Maint SV124 OFF since Auto</td>
<td>MAINTENANCE.SV_124.ON.pvfl := 0</td>
</tr>
</tbody>
</table>

#### Transition Conditions and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tank Height &lt; 5%</td>
<td>OPERATIONS: Tank Height &lt; 5%</td>
</tr>
</tbody>
</table>

```
P1:And
P2:None
```

```
P1:And
```

```
P1:And
```

```
P1:And
```

```
P1:And
```

#### Step Outputs and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raw Water SV_104 to Program Mode</td>
<td>SV_104.DEVCTL_MODEATTR := 2</td>
</tr>
<tr>
<td>2</td>
<td>Raw Water SV_124 to Program Mode</td>
<td>SV_124.DEVCTL_MODEATTR := 2</td>
</tr>
<tr>
<td>3</td>
<td>Set SV104 to Auto</td>
<td>SV_104.AUTO.pvfl := 1</td>
</tr>
<tr>
<td>4</td>
<td>Set SV124 to Auto</td>
<td>SV_124.AUTO.pvfl := 1</td>
</tr>
<tr>
<td>5</td>
<td>Set Maintenance Auto ON SV104</td>
<td>MAINTENANCE.SV_104.auto.pvfl := 1</td>
</tr>
<tr>
<td>6</td>
<td>Set Maintenance Auto ON SV124</td>
<td>MAINTENANCE.SV_124.auto.pvfl := 1</td>
</tr>
<tr>
<td>7</td>
<td>Set Maint SV104 OFF since Auto</td>
<td>MAINTENANCE.SV_104.ON.pvfl := 0</td>
</tr>
<tr>
<td>8</td>
<td>Set Maint SV124 OFF since Auto</td>
<td>MAINTENANCE.SV_124.ON.pvfl := 0</td>
</tr>
</tbody>
</table>

```
P1:And
P2:None
```

```
P1:And
```

```
P1:And
```

```
P1:And
```

```
P1:And
```

```
P1:And
```

```
P1:And
```
Appendix A.30

Appendix A.30 Maintenance Program SCM
Appendix A.30

Transition Conditions and Descriptions

1. Time HOUR
   CEE_01.HOUR = MAINTENANCE.HOUR
2. Time Minute
   CEE_01.MINUTE = MAINTENANCE.MINUTE
3. BUSINESS DAY
   CEE_01.DAY <> AND CEE_01.DAY < 1
4. Plant not in Operation
   OPERATION.PLANT_OPERATION.pvfl = 0
5. Automatic Maintenance Active
   MAINTENANCE.AUTO.pvfl = 1
6. Maintenance Start
   OPERATION.MAINTENANCE.pvfl = 10
7. Plant in Operation
   OPERATION.PLANT_OPERATION.pvfl = 1
8. PLANT_USER_ON FLAG
   OPERATION.PLANT_USER_IN_OUT = 1
9. Demonstration Not Active
   OPERATION.DEMONSTRATION.pvfl = 0

Start Conditions

Step Outputs and Descriptions

1. Set plant PLANT PROG FLAG ON
   OPERATION.PLANT_PROG.pvfl = 1
2. Reset Maint SS Flag
   MAINTENANCE.Maint_SS.pvfl = 20
3. Set Maint Timer on FLAG ON
   MAINTENANCE.Maint_TIMER.pvfl = 1
4. Set in Maintenance Mode Flag
   MAINTENANCE.MAINTENANCE.pvfl = 1
5. Set Maintenance Buzzer ON
   MAINTENANCE.WARN_BUZZ.pvfl = 1
6. Set Maint Timer for 2 Mins
   MAINTENANCE.MAINT_TIMERDELAYTIME = 120
7. Set Maint Fault Timer for 2.5 Mins
   MAINTENANCE.MAINT_FAULT_DELAYTIME = 150

M_STEP00
Set Plant in Operation

Transition Conditions and Descriptions

1. Plant Ready Flag Triggered
   OPERATION.PLANT_RDY.pvfl = 1

Step Outputs and Descriptions

1. Raw Water SV_104 TO PROGRAM Mode
   SV_104.DEVCVLA.MODEATTTR = 2

M_TRANS10
Plant Ready
Appendix A.30

### Transition Conditions and Descriptions

1. Time HOUR
   CEE_01.HOUR = MAINTENANCE.HOUR
2. Time Minute
   CEE_01.MINUTE = MAINTENANCE.MINUTE
3. BUSINESS DAY
   CEE_01.DAY \(!=\) AND CEE_01.DAY < 1
4. Plant not in Operation
   OPERATION.PLANT_OPERATION = 0
5. Automatic Maintenance Active
   MAINTENANCE.AUTO = 1
6. Maintenance Start
   OPERATION.MAINTENANCE = 10
7. Plant in Operation
   OPERATION.PLANT_OPERATION = 1
8. PLANT USER ON FLAG
   OPERATION.PLANT_USER_ON = 1
9. Demonstration not Active
   OPERATION.DEMONSTRATION = 0

### Step Outputs and Descriptions

- Set plant PLANT PROG FLAG ON
  OPERATION.PLANT_PROG = 1
- Reset Maint SS Flag
  MAINTENANCE.Maint_SS = 20
- Set Maint Timer On FLAG
  MAINTENANCE.Maint_TIMER = 1
- Set in Maintenance Mode Flag
  OPERATION.MAINTENANCE = 1
- Set Maintenance Buzzer ON
  MAINTENANCE.WARN_BUZZ = 1
- Set Maint Timer for 2 Mins
  MAINTENANCE.Maint_TIMER_DELAY = 120
- Set Maint Fault Timer for 2.5 Mins
  MAINTENANCE.Maint_FAULT_DELAY = 150

### Transition Conditions and Descriptions

1. Plant Ready Flag Triggered
   OPERATION.PLANT_RDY = 1

### Step Outputs and Descriptions

- Raw Water SV_104 to PROGRAM Mode
  SV_104.DEVCTL.MODEATTR = 2
- Set SV_104 high
Appendix A.30

Transition Conditions and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time HOUR</td>
</tr>
<tr>
<td></td>
<td>CEE_01.HOUR = MAINTENANCE,HOUR.py</td>
</tr>
<tr>
<td>2</td>
<td>Time Minute</td>
</tr>
<tr>
<td></td>
<td>CEE_01.MINUTE = MAINTENANCE,MINUTE.py</td>
</tr>
<tr>
<td>3</td>
<td>BUSINESS DAY</td>
</tr>
<tr>
<td></td>
<td>CEE_01.DAY &lt;&gt; AND CEE_01.DAY&lt;&gt;1</td>
</tr>
<tr>
<td>4</td>
<td>Plant not in Operation</td>
</tr>
<tr>
<td></td>
<td>OPERATION,PLANT_OPERATION.py= 0;</td>
</tr>
<tr>
<td>5</td>
<td>Automatic Maintenance Active</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,MAINT_AUTO,PF1= 1;</td>
</tr>
<tr>
<td>6</td>
<td>Maintenance Start</td>
</tr>
<tr>
<td></td>
<td>OPERATION,MAINTENANCE,PF1= 10;</td>
</tr>
<tr>
<td>7</td>
<td>Plant In Operation</td>
</tr>
<tr>
<td></td>
<td>OPERATION,PLANT_OPERATION.py= 1</td>
</tr>
<tr>
<td>8</td>
<td>PLANT_USER_ON FLAG</td>
</tr>
<tr>
<td></td>
<td>OPERATION,Plant_User_ON_R,OUT=1</td>
</tr>
<tr>
<td>9</td>
<td>Demonstration / Active</td>
</tr>
<tr>
<td></td>
<td>OPERATION,DEMONSTRATION,PF1= 0</td>
</tr>
</tbody>
</table>

Step Outputs and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set plant PLANT_PROG_FLAG ON</td>
</tr>
<tr>
<td></td>
<td>OPERATION,PLANT_PROG_ON,PF1= 1</td>
</tr>
<tr>
<td>2</td>
<td>Reset Maint SS Flag</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,Maint_SS,PF1= 0</td>
</tr>
<tr>
<td>3</td>
<td>Set Maint_Timer_on_FLAG</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,Maint_TIMER_ON,pF1= 1</td>
</tr>
<tr>
<td>4</td>
<td>Set in Maintenance Mode Flag</td>
</tr>
<tr>
<td></td>
<td>OPERATION,MAINTENANCE,PF1= 1</td>
</tr>
<tr>
<td>5</td>
<td>Set Maintenance Buzzer ON</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,Warn_Buzz_ON,PF1= 1</td>
</tr>
<tr>
<td>6</td>
<td>Set Maint Timer for 2 Mins</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,Maint_TIMER,DELAYTIME = 120</td>
</tr>
<tr>
<td>7</td>
<td>Set Maint Fault Timer for 2.5 Mins</td>
</tr>
<tr>
<td></td>
<td>MAINTENANCE,Maint_FAULT,DELAYTIME = 150</td>
</tr>
</tbody>
</table>

Transition Conditions and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Plant Ready Flag Triggered</td>
</tr>
<tr>
<td></td>
<td>OPERATION,PLANT_RDY,PF1= 1</td>
</tr>
</tbody>
</table>

Step Outputs and Descriptions

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raw Water SV_104 to PROGRAM Mode</td>
</tr>
<tr>
<td></td>
<td>SV_104.DEVFCL,MODEATTR = 2</td>
</tr>
<tr>
<td>2</td>
<td>Set SV_104 flag</td>
</tr>
</tbody>
</table>
Appendix A.30

Transition Conditions and Descriptions
1. Time HOUR
   CEE_01.HOUR = MAINTENANCE.HOUR(py)
2. Time Minute
   CEE_01.MINUTE = MAINTENANCE.MINUTE.py
3. BUSINESS DAY
   CEE_01.DAY <= 7 AND CEE_01.DAY < 1
4. Plant not in Operation
   OPERATION.PLANT_OPERATION.py = 0
5. Automatic Maintenance Active
   MAINTENANCE.AUTO.PVFL = 1
6. Maintenance Start
   MAINTENANCE.Maintenance.PVFL = 10;
7. Plant in Operation
   OPERATION.PLANT_OPERATION.py = 1
8. PLANT_USER_ON FLAG
   OPERATION.PLANT_USER_ON.R.OUT = 1
9. Demonstration not Active
   OPERATION.DEMONSTRATION.PVFL = 0

Step Outputs and Descriptions
1. Set plant PROG FLAG ON
   OPERATION,PLANT_PROG_ON.PVFL = 1
2. Reset Maint_SS Flag
   MAINTENANCE,Maint_SS.PVFL = 0
3. Set Maint_TIMER On FLAG ON
   MAINTENANCE,Maint TIMER.ON.PVFL = 1
4. Set in Maintenance Mode Flag
   OPERATION,MAINTENANCE.PVFL = 1
5. Set Maintenance Buzzer ON
   MAINTENANCE,WARN BUZZ ON.PVFL = 1
6. Set Maint Timer for 2 Mins
   MAINTENANCE,Maint TIMER,DELAYTIME = 120
7. Set Maint Fault Timer for 2.5 Mins
   MAINTENANCE,Maint_FAULT,DELAYTIME = 150

Transition Conditions and Descriptions
1. Plant Ready Flag Triggered
   OPERATION,PLANT_RDY.PVFL = 1

Step Outputs and Descriptions
1. Raw Water SV_104 lo PROGRAM Mode
   SV_104.DEVCL,MODEATT.R = 2
   Set SV_104 Flag
Appendix A.30

Transition Conditions and Descriptions

1. Time HOUR
   CEE_01.HOUR = MAINTENANCE.HOUR

2. Time Minute
   CEE_01.MINUTE = MAINTENANCE.MINUTE

3. Business Day
   CEE_01.DAY <> AND CEE_01.DAY < 1

4. Plant Not in Operation
   OPERATION,PLANT_OPERATION.PVFL = 0

5. Automatic Maintenance Active
   MAINTENANCE,MAINT.AUTO.PVFL = 1

6. Maintenance Start
   OPERATION,MAINTENANCE.PVFL = 1

7. Plant in Operation
   OPERATION,PLANT_OPERATION.PVFL = 1

8. PLANT_USER_ON_FLAG
   OPERATION,PLANT_USER_ON.R.OUT = 1

9. Demonstration Not Active
   OPERATION,DEMONSTRATION.PVFL = 0

Invoke MAINTENANCE Start Conditions

Step Outputs and Descriptions

1. Set plant PLANT_PROG_FLAG ON
   OPERATION,PLANT_PROG.ON.PVFL = 1

2. Reset Maint_SS Flag
   MAINTENANCE,Maint_SS,pvfl = 20

3. Set Maint_Timer ON FLAG ON
   MAINTENANCE,Maint_TIMER_ON.pvfl = 1

4. Set in Maintenance Mode Flag
   OPERATION,MAINTENANCE.PVFL = 1

5. Set Maintenance Buzzer ON
   MAINTENANCE,WARN_BUZZ_ON.pvfl = 1

6. Set Maint Timer for 2 Mins
   MAINTENANCE,Maint_TIMER,DELAYTIME = 120

7. Set Maint Fault Timer for 2.5 Mins
   MAINTENANCE,Maint_FAULT,DELAYTIME = 150

M_STEP00 Set Plant in Operation

Transition Conditions and Descriptions

1. Plant Ready Flag Triggered
   OPERATION,PLANT_RDY.PVFL = 1

Invoke MAINTENANCE

Step Outputs and Descriptions

1. Raw Water SV_104 to PROGRAM Mode
   SV_104,DEVCVLA,MODEATTR = 2

2. Set SV_104 Flag
Appendix A.32 HMIWeb Display Screen Modifications
Appendix A.32
Appendix A.32