Analysis of Process Control Methods in a First Order Motor System

Olivia Kuebler, Adrian Marusic, Luke Oluoch, and Jonathan White

March 20, 2019

Abstract

In any large-scale processing plant, the most important aspect of production efficiency is consistency in system control. Process control allows the user to obtain consistency in their process through the use of different control algorithms. For this project, IMC and IMC-PID controls were examined to determine which control was more consistent in modeling an oscillating harmonic function for a first order motor-tachometer system. The algorithms were examined for not only accuracy, but for a better understanding of the drawbacks inherent in each. Tuning parameters were found by using a regression of data that showed how well the system responded to changes in frequency. That regression solved for parameters used in the IMC system which, in turn, were used to find the IMC-PID parameters. The two control algorithms were then compared on their ability to track a response under a variety of conditions resulting in the IMC-PID generally being the most advantageous system.

Contents

1 Introduction **1**

2 Obtaining Parameters for Models **1**

3 Graphical Results and Comparison **2**
- i **Normal Conditions** 2
- ii **Extreme Frequencies and Amplitudes** 3
- iii **Disturbance** 3
- iv **Response Limitations** 4
- v **Reset Windup** 4
- vi **IMC Advantages and Disadvantages** 5

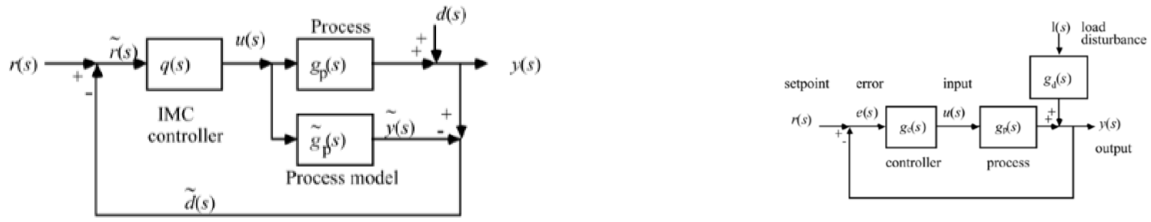
4 Conclusion **5**

5 Bibliography **6**

6 Appendix **7**
- i **Regression Data Collection LabView Code** 7
- ii **IMC-PID LabView Program Code** 7
- iii **IMC LabView Program Code** 8

1. Introduction

With the advent of modern technology, process control systems have become a vital asset in industry. The most common process control system used is PID (Proportional, Integral, Derivative) control, which is used in most industrial settings that require continuous modulated control. IMC (Internal Model Control) is another method of control that requires an accurate model to adjust to a given setpoint. The model serves the purpose of allowing the process to respond to the setpoint more efficiently. Equations for the IMC control and PID control can be manipulated to become equivalent, leading to an IMC-PID hybrid feedback control system. This type of control uses IMC parameters to find PID parameters.



(a) *IMC Block Diagram*

(b) *IMC-PID Block Diagram*

Figure 1: *System Block Diagrams*

These block diagrams correspond to the following equations: IMC Block Diagram:

$$y(s) = \frac{g_p(s)q(s)}{1 + q(s)(g_p(s) - \tilde{g}_p(s))} r(s) \tag{1}$$

IMC-PID Block Diagram With Load:

$$y(s) = \frac{g_d(s)}{1 + g_p(s)g_c(s)} l(s) + \frac{g_d(s)}{1 + g_p(s)g_c(s)} r(s) \tag{2}$$

IMC-PID Block Diagram Without Load

$$y(s) = \frac{g_d(s)}{1 + g_p(s)g_c(s)} r(s) \tag{3}$$

The goal of this project was to implement an IMC and IMC-PID control for a first order motor-tachometer system. The system is comprised of two gears driven by a motor controlled with a LabJack signal with a maximum of 5V and a minimum of 0V.

2. Obtaining Parameters for Models

To obtain reasonable responses from this system for both IMC and IMC-PID controls, specific parameters must be found in order to properly tune the system. For IMC control, values were needed for the process gain k_p, time constant τ_p and minimum driving voltage M_{min} that are based on Equation 4 which is derived below.:

IMC Equation:

$$u(\theta) = M(\theta) - M_{min} = \frac{\tau_p}{\lambda k_p}\{r(\theta) - y(\theta) + y^m(\theta)\} + \frac{1}{\lambda k_p}\int_0^{\theta}\{r(t) - y(t) + y^m(t) - k_p u(t)\}dt \tag{4}$$

where

$$y^m(\theta) = \frac{1}{\tau_p}\int_0^{\theta}\{k_p u(t) - y^m(t)\}dt \tag{5}$$

r(t) and r(θ) are the model inputs, y(t) and y(θ) are the system responses, $y^m(t)$ and $y^m(\theta)$ are the model responses, and u(t) and u(θ) is the model output. The parameters that required tuning were k_p, λ, and τ_p.

Tuning IMC Parameters Equation:

$$c(\theta) = k_p(M_0 - M_{min})[1 - e^{-\frac{\theta}{\tau_p}}] + \frac{k_p M_A}{\tau_p(\omega^2 + \frac{1}{\tau_p^2})}[\omega e^{-\frac{\theta}{\tau_p}} + \frac{sin(\omega\theta + \phi)}{\tau_p cos(\phi)}] \tag{6}$$

where $c(\theta)$ is the system response, M_A is the amplitude, and $\phi = tan^{-1}(-\omega\tau_p)$.

To control the system using these equations, the LabView program shown in Appendix iii was developed. The program takes data for the response of the motor based on inputs of amplitude, frequency, and offset, based on a sine function. The data was collected with the intent of keeping the amplitude (0.3) and offset (2.5) constant while the frequency was varied from 0.50 to 1.20 in increments of 0.10. After the data was collected, the parameters could be regressed based on the above equation. The results can be seen in Table 1.

Table 1: *Results of Parameter Regression*

IMC	IMC-PID
$K_p = 14.1784$	$k_p = 0.555070$
$\tau_p = 1.84945$	$k_i = 0.540702$
$M_{min} = 2.32524$	$k_d = 0$

Once those parameters were established, λ could be found using the IMC labview program as seen in Appendix i. By observing the system's response to varying λ (the filter), a value was found that provided the least amount of error between the setpoint and the response. The value of λ that produced the least amount of error was 0.235.

For the IMC-PID, we are looking for tuning parameters k_p, k_i and k_d which were found through the following equations:

IMC-PID equation:

$$u(\theta) = k_p e(\theta) + k_i \int e(\theta) d\theta + k_d \frac{de(\theta)}{d\theta} \tag{7}$$

where $e(\theta)$ is the error between the system and the response. The parameters that required tuning were k_p and k_i. k_d was not applicable to this particular system, therefore it was left at zero.

IMP-PID Tuning:

$$k_p = \frac{\tau_p}{\lambda k_p} \tag{8}$$

$$k_i = \frac{1}{\tau_p} \tag{9}$$

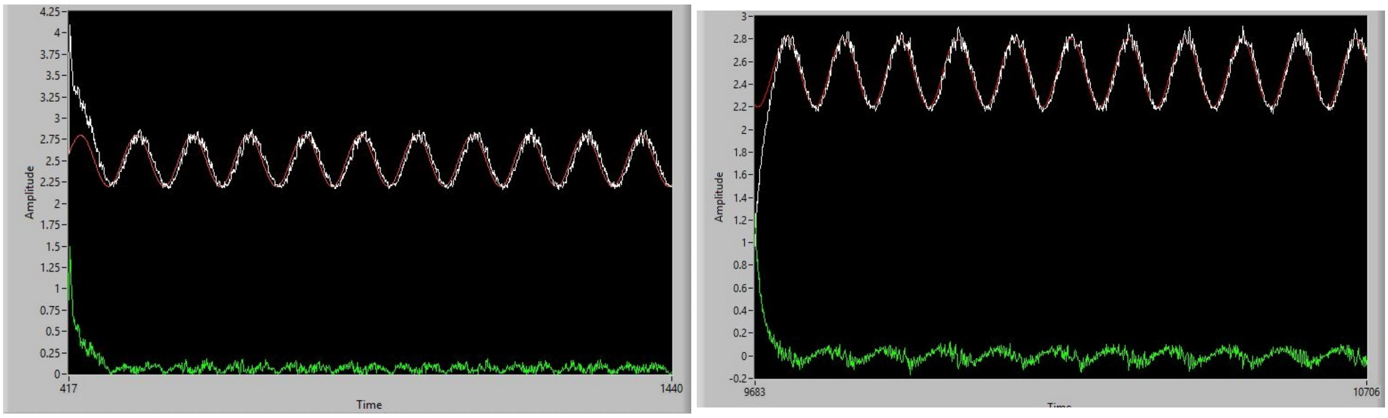
The values for these can be seen in Table 1.

3. Graphical Results and Comparison

i. Normal Conditions

The first comparison considered between the IMC and IMC-PID was how well they followed the setpoint under normal operating conditions.

For these plots and all subsequent plots, the white line is the system's response, the red line is the set point, and the green line is the error between the two.



(a) *Standard IMC Run*

(b) *Standard IMC-PID Run*

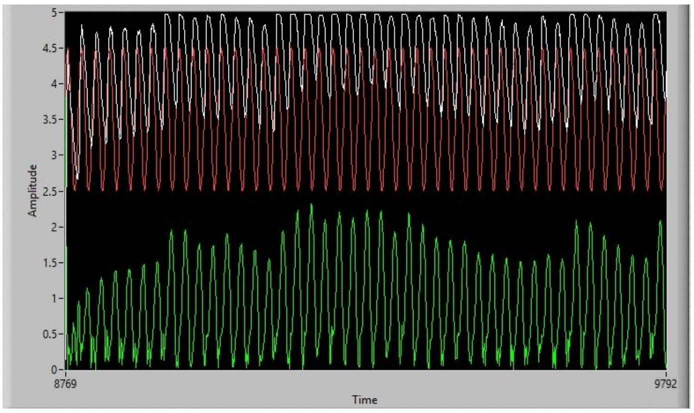
Figure 2: ***Standard IMC and IMC-PID Comparison***

Based on these standard runs, it is clear that both responses are a little "noisy" when trying to follow the set point. This can most likely be attributed to the fact that the motor is not a perfect system, and can only follow a set point to a certain degree, especially while that set point is shifting.

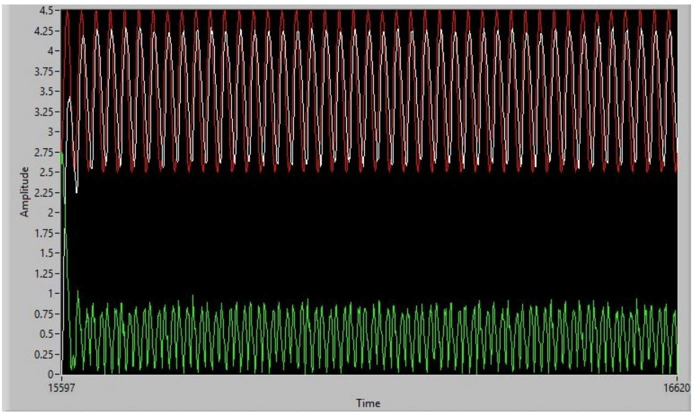
Despite this noise, it is clear that both control algorithms follow the set point fairly well. The IMC-PID seems to follow the set point just a bit more closely than the IMC, which has a response that is shifted slightly to the right of the set point.

ii. Extreme Frequencies and Amplitudes

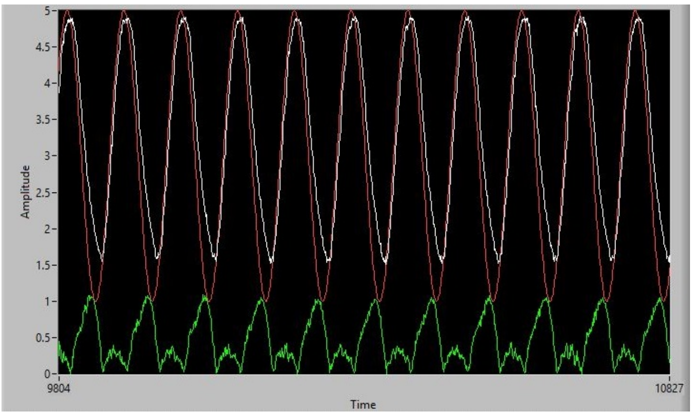
After taking the standard runs for both systems, extremes were tested for both amplitude and frequency to see how well the systems could keep up with the expected response. Due to the limitations in the speed of calculation and response, it makes sense that the system cannot follow the set point if it changes especially quickly.



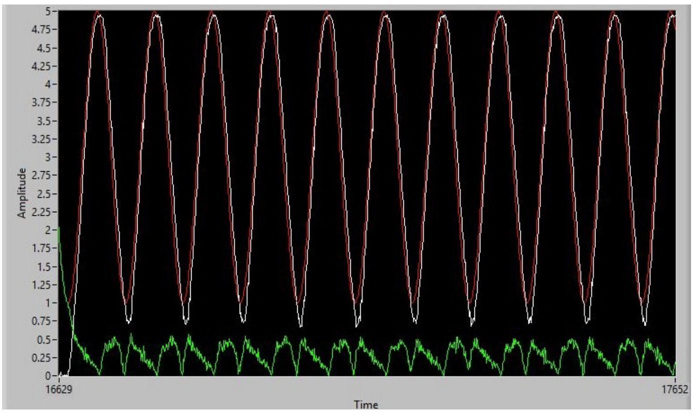
(a) *IMC High Frequency*



(b) *IMC-PID High Frequency*



(c) *IMC High Amplitude*



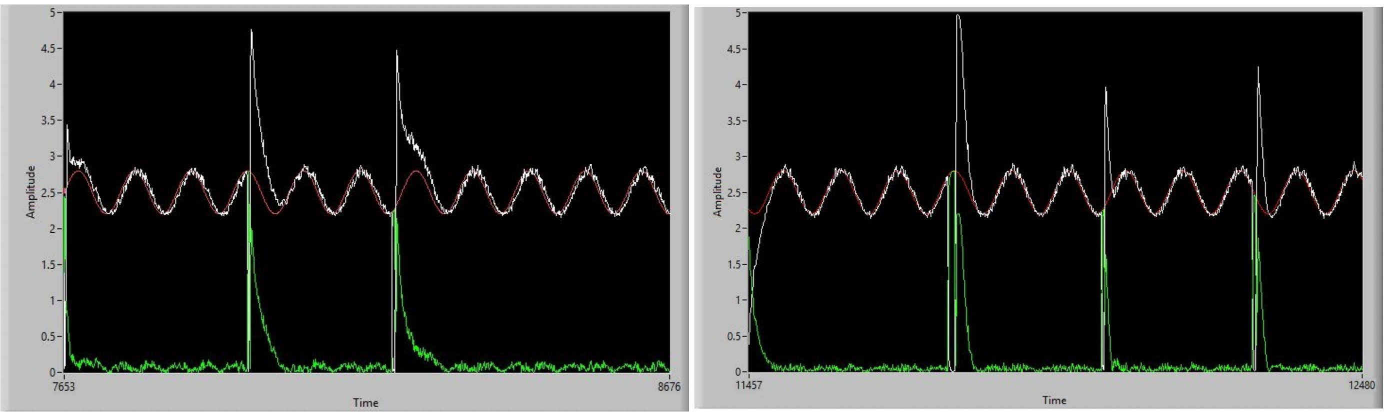
(d) *IMC-PID High Amplitude*

Figure 3: ***High Frequency and Amplitude Comparisons***

It is clear that in both instances of testing extreme amplitude and frequency values, the IMC-PID kept to the setpoint better than the IMC system. When testing for high frequency, the IMC remained mostly above and within the upper portion of the set amplitude, while the IMC-PID remained within it. When testing for high amplitude, the IMC had a tendency to undershoot the top and bottom of the sine curve, while the IMC-PID would also slightly undershoot, but kept to the curve a bit better.

iii. Disturbance

The next comparison considered was how well each system responded to a disturbance. These disturbances were created by physically perturbing the motor as it was running. Although disturbances in this system were uncommon and needed to be induced intentionally, disturbances in industry can occur in a number of places and as such it is important to recognize which control algorithm can recover faster from disturbances.



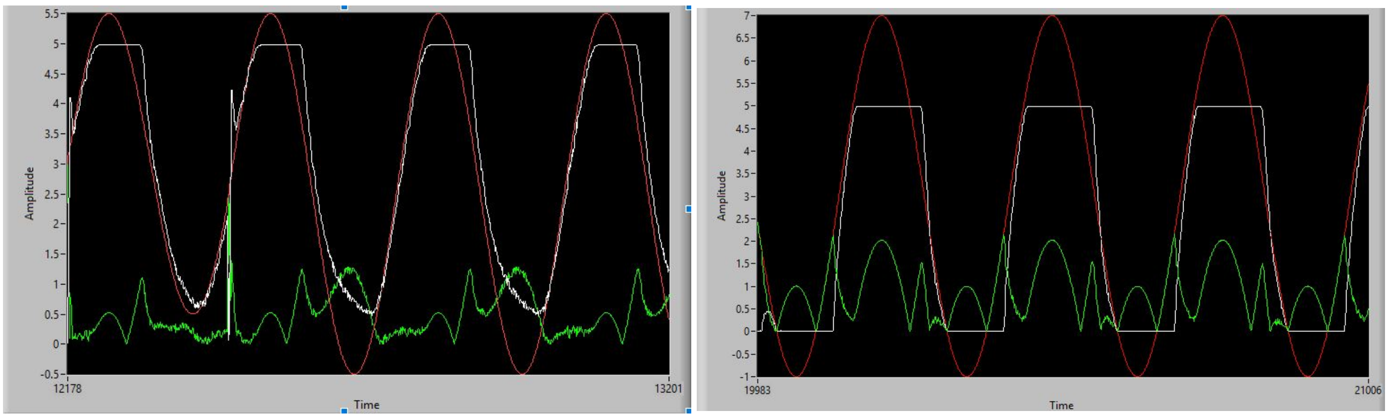
(a) *IMC Applying a Disturbance* **(b)** *IMC-PID Applying a Disturbance*

Figure 4: *Disturbance Comparison*

While the magnitude of the disturbances were difficult to keep similar between systems, it is clear that the IMC-PID system responds better, but both respond fairly well.

iv. Response Limitations

The motor system has limits on how it can respond to the inputs applied to it. There is a maximum possible voltage of 5V and a minimum of 0V. There are also separate minimum voltages for the motor as it starts up and another minimum while running. This can be seen on the graphs, which show that both control algorithms will peak at 5V since that's the maximum the motor can provide, and neither control can go lower than 0V, so both react accordingly, with the IMC-PID going to 0V and staying there until the demanded response is above 0V, and the IMC keeps running at a positive voltage and catches back up with the model on its way up to higher voltages.



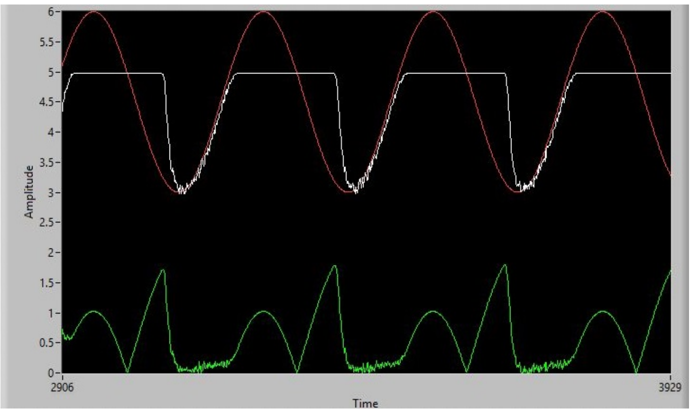
(a) *IMC for a Sine Wave Above and Below Reasonable Range* **(b)** *IMC-PID for a Sine Wave Above and Below Reasonable Range*

Figure 5: *Limitation Comparison*

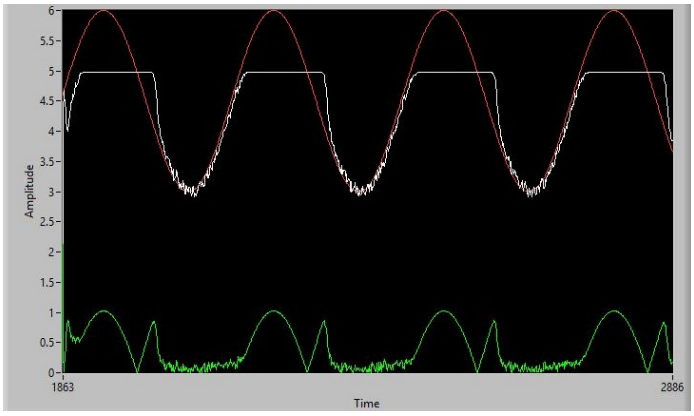
v. Reset Windup

Reset windup occurs when the system is asked to produce a response that is not mechanically feasible, and the control algorithm attempts to compensate by asking for a higher and higher response from the system. This results in the system overcompensating, and when the target value returns into the feasible range of the system, there is a delay in returning to the set point. This can be greatly reduced by putting limits on the integration terms in the IMC-PID, and by putting limits on u* in the IMC.

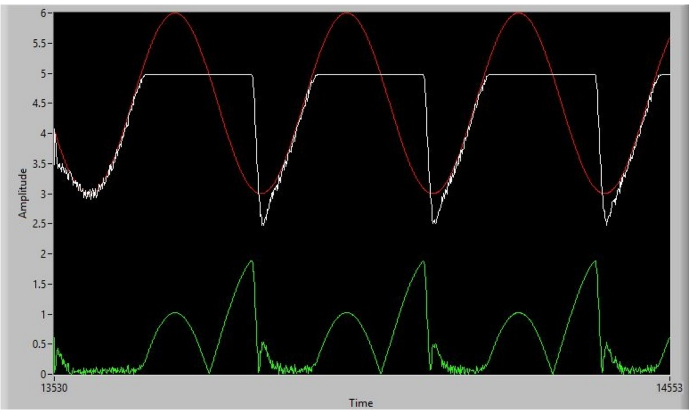
The graphs show that the windup is greatly reduced with the right limits, and the IMC and IMC-PID are better able to get the system back on track with these limits. It is worth noting that the limits need to be tuned to the system, as shown with the difference between a limit of 3.1 and 5 on the IMC-PID controller. It can be seen that a limit of 3.1 fits the system much better than a limit of 5.



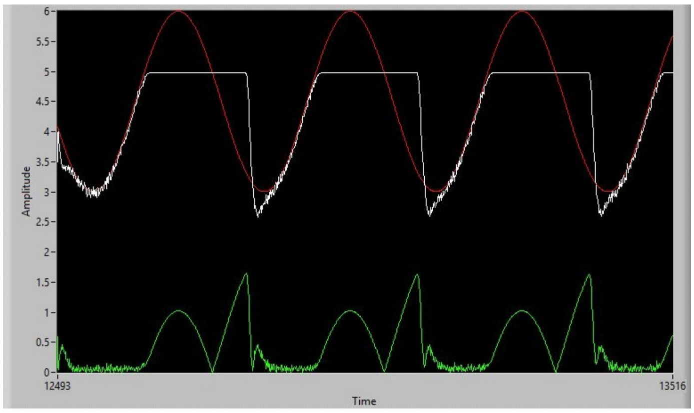
(a) *IMC With Reset Windup*



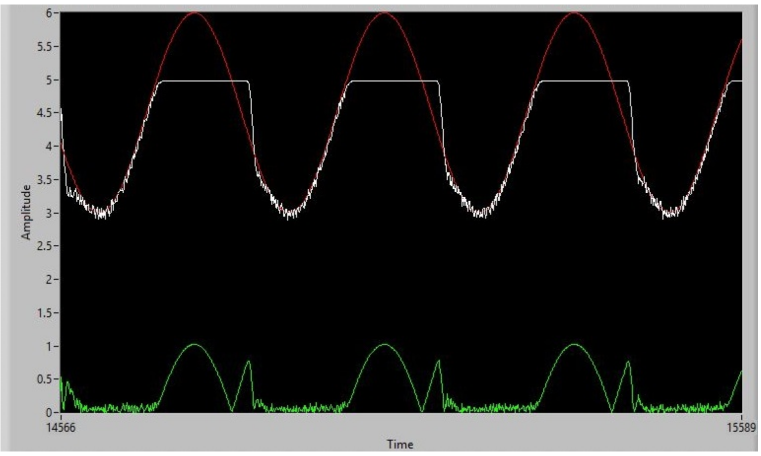
(b) *IMC No Reset Windup*



(c) *IMC-PID With Reset Windup*



(d) *IMC-PID No Reset Windup (Upper Limit of 5 on Integration)*



(e) *IMC-PID No Reset Windup (Upper Limit of 3.1 on Integration)*

Figure 6: *Reset Windup Comparisons*

vi. IMC Advantages and Disadvantages

Despite not keeping up with the IMC-PID in all circumstances, IMC control is advantageous over standard PID feedback control for certain situations. If there is a model involved, the IMC is usually more consistent than standard feedback as it takes a model into account. Due to this, the response to any disturbance or error will be superior to what the standard feedback can provide. Secondly, the IMC control algorithm bypasses the need to tune the system if a good model is available, unlike PID control, and can be much more accurate due to being based on model. However, IMC control will work poorly when the system is unstable or cannot be modeled well.

4. Conclusion

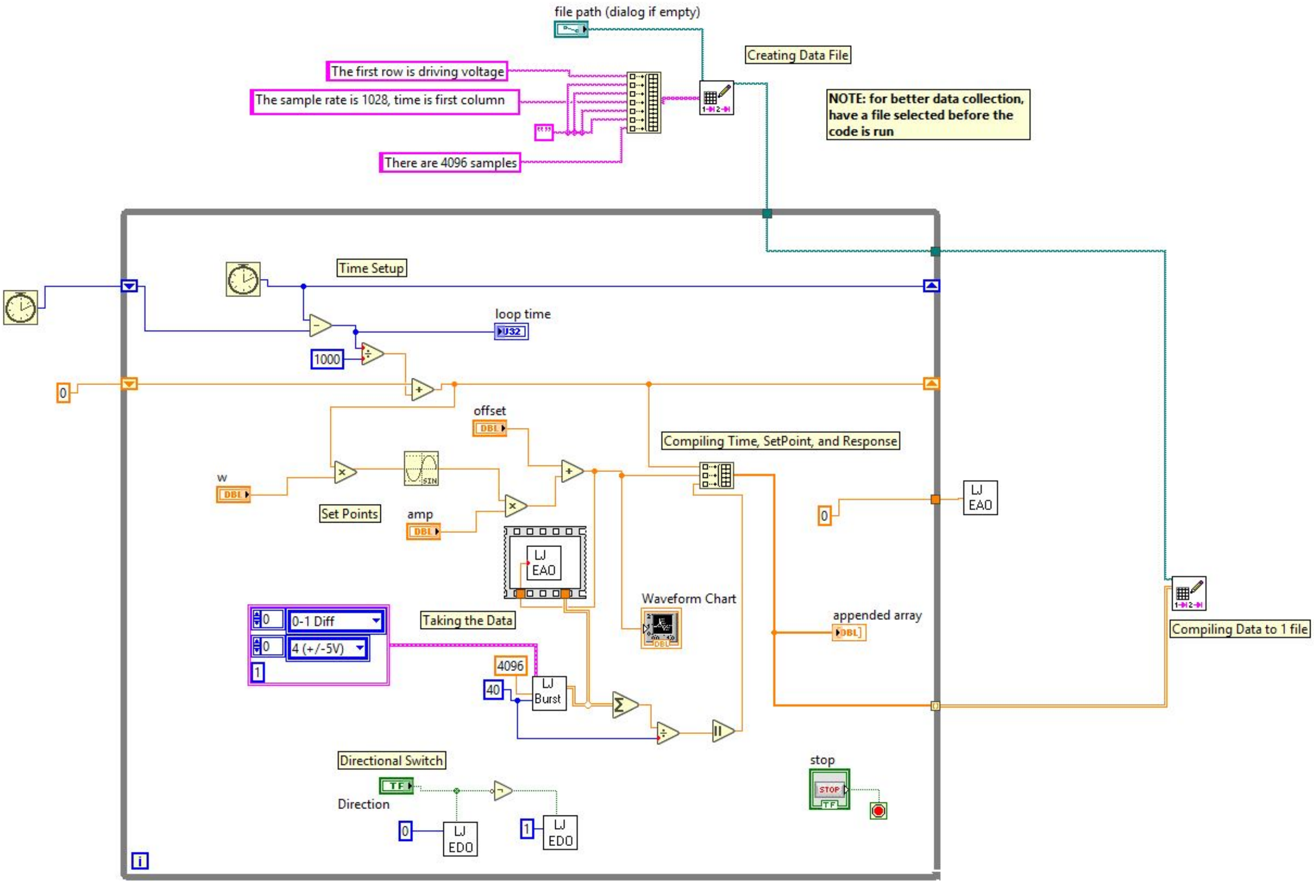
Based on all comparisons made in this paper, the IMC-PID was generally able to respond better to setpoint changes than the IMC. This is vital information to know and understand as a variety of errors can occur while a process is being carried out, and having a system that can recover from errors more efficiently saves a significant amount of energy. Although an IMC is advantageous in some settings when compared to the traditional feedback control, it is best to use an IMC-PID in most cases.

5. Bibliography

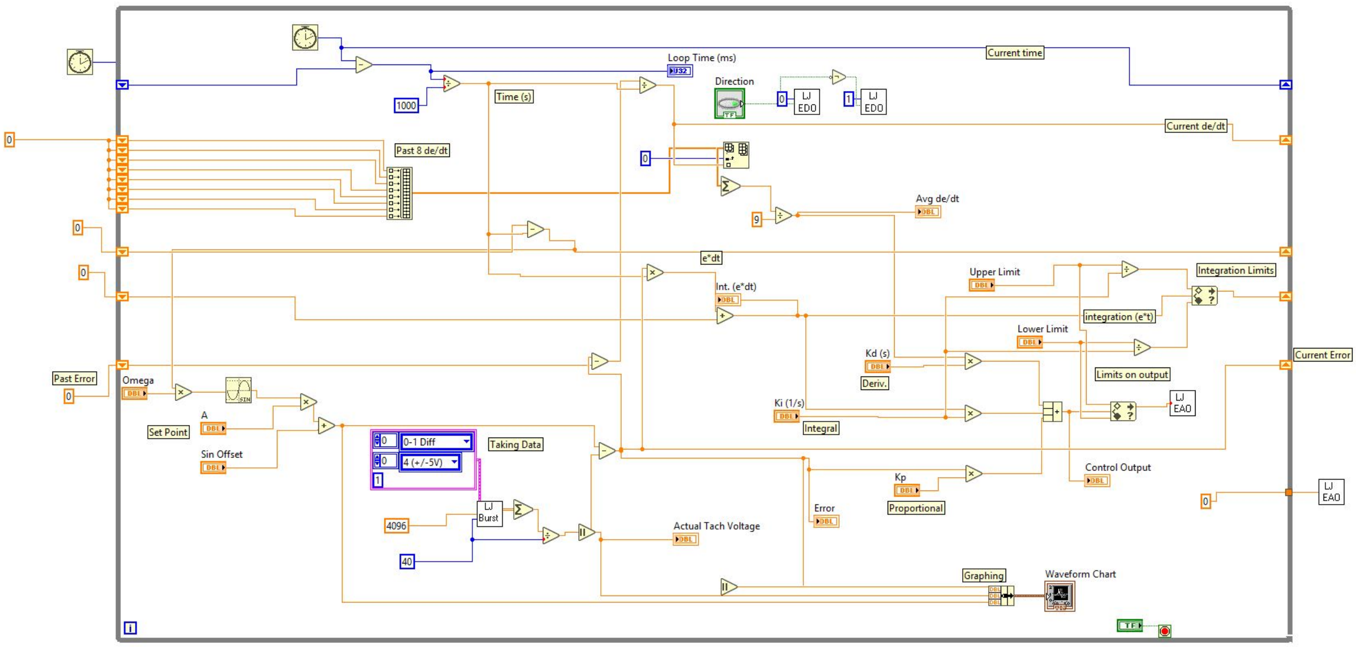
Bequette, B. Wayne. *Process Control: Modeling, Design, and Simulation.* 1st ed., Prentice Hall, 2010.

6. Appendix

i. Regression Data Collection LabView Code



ii. IMC-PID LabView Program Code



iii. IMC LabView Program Code