University of Iowa
Iowa Research Online

Theses and Dissertations

Summer 2010

Improved modeling and optimal control of an electric arc furnace

Jared James Snell
University of Iowa

Copyright 2010 Jared James Snell

This thesis is available at Iowa Research Online: http://ir.uiowa.edu/etd/741

Recommended Citation
Snell, Jared James. "Improved modeling and optimal control of an electric arc furnace." MS (Master of Science) thesis, University of Iowa, 2010.
http://ir.uiowa.edu/etd/741.

Follow this and additional works at: http://ir.uiowa.edu/etd

Part of the Electrical and Computer Engineering Commons

IMPROVED MODELING AND OPTIMAL CONTROL OF AN ELECTRIC ARC FURNACE

by

Jared James Snell

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Electrical and Computer Engineering in the Graduate College of The University of Iowa

July 2010

Thesis Supervisor: Professor Er-Wei Bai

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

____

MASTER’S THESIS

____

This is to certify that the Master’s thesis of

Jared James Snell

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Electrical and Computer Engineering at the July 2010 graduation.

Thesis Committee: ____
Er-Wei Bai, Thesis Supervisor

____
Punam Saha

____
Anton Kruger

ACKNOWLEDGMENTS

I would like to thank my committee members, Professor Punam Saha and Professor Anton Kruger for their time and volunteering some of their summer to help.

I would like to thank Professor Er-Wei Bai for all the direction, and always making me examine things more closely. I don't think I've ever met someone more excited about teaching.

I'd like to thank Zhijun Cai for all the help on everything graduate school related and support as a research assistant.

I'd like to thank Billy Bryant for all the help, interviews, and for sharing his knowledge of electric arc furnaces.

Most of all I'd like to thank my sponsor, The Iowa Energy Center for their research grant and Gerdau Ameristeel, Wilton Iowa Facility for their help.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter introduces the motivation, background, and organization of the thesis.

1.1 Motivation

Steel is one of the fundamental materials of the modern age. It is used in almost every facet of life today. In most markets, there are a limited amount of resources available to the producer and consumer. The steel market is no exception. At the time this thesis was written, there is roughly 122 million metric tons of steel produced in the world every month [6].

Much of this steel is produced with directly mined iron ore. With this amount of steel already produced and iron ore always decreasing, recycling can be a beneficial option for both cost and resources. In fact, generally the least costly method of producing steel involves using scrap steel. Facilities that use this recycling method are called mini-mills and use a furnace called an EAF (electric arc furnace). While EAF's are usually small in comparison to a typical blast furnace, they have many advantages. EAF's use scrap steel, which can be found just about anywhere, so it does not need to be located near an iron source, but can be placed closer to the customer to save costs. EAF's also generally have lower energy costs than blast furnaces [3].

These facts contribute to the rise of steel production by EAF and why now over half of the United State's steel is produced in mini-mills. Recently, the steel industry in the U.S. has been hit hard by a recession, scaling back production and making it necessary to cut every cost possible. The EAF continues to have a severe lack of automation in comparison to most industrial processes. Most decisions are made by operators inspecting the furnace personally. This can lead to a poor quality product, inefficient use of energy, and a need for well experienced operators. These reasons and more make it highly desirable to have an optimal control scheme available to use with the EAF.

1.2 Background

The Gerdau Ameristeel mini-mill facility in Wilton, Iowa includes an 80 ton 25MW AC EAF. When operating at full capacity it employs about 350 workers and was commissioned in 1975. The factory makes about 100 different kinds of finished product and can make those products out of roughly 100 different grades of steel. The facility is different in that it does not have a ladle refining station. In order to make up for this, all the alloying must take place in the EAF.

1.2.1 Melting Process Description

The melting process describes what happens from the time that scrap steel is first dumped into the furnace to the time that the steel becomes completely liquid.

First, a crane loads some amount of scrap from nearby storage into a charging bucket. The same crane then picks up the charging bucket and takes the scrap to the furnace. It then drops the scrap into the furnace, which is called charging the furnace. At the Wilton facility, there are typically three charges to a heat. A heat is one batch of liquefied and refined steel. The exact amount of charges as well as the size of the charges varies. This can depend on the operator, the steel, grade, or the quality of scrap. For this thesis, it is assumed that all heats contain exactly three charges which is a typical practice. . The melting cycle is as follows:

1. Furnace Turn Around (FTA): the furnace is prepared and checked over. The furnace roof swings aside to allow scrap to be dropped in.
2. Charge: Scrap is charged into the furnace
3. Melt: The roof and electrodes are lowered on to the furnace. The EAF is activated and the scrap melts until it is liquid enough to fit an additional charge in. This repeats for each additional charge. After the third charge is liquefied, the refining stage begins
4. Refine: Different metals and alloys are added to the steel to achieve the specific grade. The steel is also brought to tapping temperature
5. Tap: The steel is poured (tapped) into the ladle.

At this point the melting cycle is completed and the liquid steel moves on to be cast.

1.2.2 EAF Description

Wilton’s EAF has a 33.6 MVA transformer. It delivers the current through aluminum conducting arms, which is then transferred to the steel via 3 20 inch diameter graphite electrodes. There are also 4 sidewall burners to assist the melting process and the EAF is contained in a 16ft shell. Of the 4 sidewall burners, 2 use a gas-oxygen combination fuel and 2 use gas-oxygen with carbon injection. This EAF uses a balanced 3 phase system. Each electrode can have its characteristics individually altered by moving the electrodes up or down. The distance between the electrode and the steel determines the length of the arc, impedance and therefore current of the phase.

Currently there are two different control systems that the EAF electrodes utilize in Wilton. The first system does the following

- Moves the electrodes with large hydraulics controlled by a programmable logic controller. A current set-point is entered and the electrodes are moved in order to maintain that amount of current. It can change the current on any phase individually using a power program.
- This system starts the actual arc current and will restart the arc if it is ever extinguished by a cave-in of scrap or other complications.
- The system is also responsible for all electrical calculations (current, impedance, etc.)

The second system

- Can dynamically calculate current set-points
- Controls the sidewall burners using a burner program
- Logs all EAF data. This takes a sample about every 5 seconds

The electrodes in an EAF are clustered in the middle of the furnace. Due to this fact, the heat generated by the arc affects the center of the furnace much more than the outside. To compensate for this, sidewall burners are strategically placed on the outside of the furnace. This both melts some of the scrap on the outside and collapses some of the solid scrap towards the center where there is more heat. As mentioned previously, there are 4 sidewall burners, 2 using a gas-oxygen combination fuel and 2 using gas-oxygen with carbon injection. They are primarily controlled by the second electrode control system discussed above. However, they can also be controlled by a completely separate burner program or by hand. The flow rate of each individual burner is able to be controlled separately. In a typical heat, the burners come on right after the power is turned on and continues for about %60 of the melt cycle [1]. It is important to note that the arc current accounts for %70-80 of the overall melting, and the sidewall burners are supplemental [1].

The EAF also includes a manually controlled oxygen lance. This is to melt scrap that may be piled in front of the slag door, or also may be used to add oxygen if one of the sidewall burners fails.

1.3 Useful Definitions

- Charge – A charge is one bucket load of scrap steel. Only one full unmelted batch can fit in the furnace at a time. It must be melted to make space and then additional charges can be added. Typically there are three charges to make one heat of steel.
- Charging – The process of adding a charge to the furnace.
- EAF – Electric arc furnace
- FIR system – A Finite Impulse Response system
- Flat bath – A flat bath is when all the steel in the furnace is completely melted. This happens only when all three charges are in the furnace.
- Heat – A heat is one batch of melted steel. Only one full melted batch can fit in the furnace at a time. For this thesis, only heats that include 3 charges are used.
- IIR system – An Infinite Impulse Response system
- I^2h - This is short for electrode consumption.

- *KWH* - Kilowatt hours. An important metric for the EAF.
- *PM* - Percent melted. This generally indicates how much steel is liquid in the furnace at any time. *PM* may be better thought of as the percentage of progress as it reaches %100 when the charge is finished. When *PM* is at %100 there is a flat bath in the furnace.
- Refining – Refining is the process of adding different alloys and metals to the steel. This occurs after a flat bath is achieved and determines the grade of steel in the finished product.
- Tap – A tap is the process of pouring the melted steel out of the furnace and into the ladle.

1.4 Previous Work

Billy W. Bryant Jr, a previous master's student at Iowa, wrote his masters thesis on the same topic in 2006 [2]. Although his model and control techniques did improve performance of the EAF in Wilton, it is believed that it could be improved further. This thesis has a significantly different approach. The model greatly differs and the optimized control system is completely different. One of the key goals of this thesis is that the new schemes not only are different than the previous attempts, but that it also outperforms them.

1.5 Thesis Goals and Contributions

The goals of this thesis are to contribute the following:

1. Reproduce modeling results from previous optimization attempts

2. Produce a model that is more accurate and simpler than previous attempts. The model also should make physical and intuitive sense for a real-world application.
3. From the new model, contribute an optimized control profile that is reasonable and reduces cost.

CHAPTER 2

MODELING

There are very few modeling and system identification schemes available in the steel industry. This is especially true for the EAF, where many of the furnaces are unique. The randomness of the arc current, the grade of steel, size of furnace, and the type and quality of scrap also add to the difficulty of modeling the system. Therefore it is a goal of this thesis to make a new modeling scheme that is more accurate and less complex than the previous attempt. It is also a goal that while the model may be accurate, it also must make intuitive sense.

2.1 Data and Identification of Parameters

The data for this thesis was contributed by the Gerdau Ameristeel plant in Wilton, Iowa. Data from 1,791 heats was collected. These were not altered in any way for the purposes of this thesis; they were just day to day typical heats. The data was obtained in Microsoft Excel file form and manipulated in MatLab. Each file has a total of 95 different parameters that are recorded approximately every 5 seconds. Of these 95 parameters, 17 were actually used in the original model. This involves only the 11 quantities listed below, as I^2h and I_x have 3 phases and charge size has three columns to indicate each charge. A certain amount of the heats were outliers. This included incomplete heats, heats with more than 3 charges, and heats that were excessively long. These outliers tended to cluster near each other and were filtered out.

The main goal of the thesis is to minimize the cost in terms of energy and capital to make a heat of steel. The main costs in operating this particular furnace, if man-hours and scrap are not included, are

- Electrode Consumption (I^2h)
- Kilowatt hours (KWH)
- Oxygen (O_2)
- Carbon
- Gas

Each cost parameter has a known constant multiplier to find the cost per unit. However, due to the fact that this information is proprietary, general constants from the industry will be shown instead throughout the thesis. The largest percentage of the expense is created by kilowatt hours. In a typical EAF, this accounts for roughly %60 of costs. The next highest expense is electrode consumption with about %20. Oxygen, gas, and carbon make up the remaining %20 [7]. The EAF has many parameters that can be measured. Among these, the key quantities for the model are

- Kilo watt hours (KWH)
- Electrode Consumption (I^2h)
- Percent Melted (PM)
- Oxygen (O_2)
- Carbon
- Gas
- Power-on-time (P_{on})
- Power-off-time (P_{off})
- Charge Size
- Heat Size

- Phase Current (I_x)

Using these parameters, the melting process can be modeled to a reasonable accuracy. These quantities were chosen because they are the dominate factors in modeling. All had been provided in the data acquired and were used in the Mr. Bryant's thesis. The very first objective of the modeling was to produce a model similar to what was documented in the previous modeling attempt. The model consists of three systems

- KWH
- I^2h
- PM

This basic three system model will be used throughout the thesis. The underlying principle of this three system model is that the KWH and I^2h systems are used to determine the energy usage and costs, while PM is useful for finding when the charge is melted and how much each metric contributes to the melting process. Using these three, an accurate depiction of the melting process can be modeled.

2.2 Model Improvement

One critical component of the project is to understand and improve the models that had been developed previously. Improvement here implies accurate prediction ability of the models without substantially increasing the model complexity, which hinders real time applications of the models. Significant enhancements in each of the models have been achieved.

2.2.1 Kilowatt hours (KWH)

Kilowatt hours account for about %18.5 of a typical mini-mill's total budget [7]. It is one of the most important metrics in the furnace as it not only supplies a large portion of the cost, but also amounts to about %70-80 of the melting of the steel [1]. Its previous developed model is shown as

$$(1) \qquad K\hat{W}H(n) = \theta_1 + \theta_2 P_{on}(n-1)$$

[2] where the superscript "hat" indicates that the value is predicted by the model. All coefficients θ_j 's for any model in this thesis are calculated by minimizing the sum of squared errors. For example, in the instance of equation (1), coefficients are obtained by the least squares theory [4].

$$(2) \qquad \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \left(\begin{pmatrix} 1 & P_{on}(1) \\ 1 & P_{on}(2) \\ \mathrm{M} & \mathrm{M} \\ 1 & P_{on}(N-2) \\ 1 & P_{on}(N-1) \end{pmatrix}^T \begin{pmatrix} 1 & P_{on}(1) \\ 1 & P_{on}(2) \\ \mathrm{M} & \mathrm{M} \\ 1 & P_{on}(N-2) \\ 1 & P_{on}(N-1) \end{pmatrix} \right)^{-1} \begin{pmatrix} 1 & P_{on}(1) \\ 1 & P_{on}(2) \\ \mathrm{M} & \mathrm{M} \\ 1 & P_{on}(N-2) \\ 1 & P_{on}(N-1) \end{pmatrix}^T \begin{pmatrix} KWH(2) \\ KWH(3) \\ \mathrm{M} \\ KWH(N-1) \\ KWH(N) \end{pmatrix}$$

where the superscript "T" denotes the matrix transpose. The coefficients θ_j 's are calculated similarly for I^2h and PM . The previous model (1) and its estimate of KWH is shown in Figure 2.1 for heat file number 29965, which will be the heat file used for the rest of the thesis unless otherwise noted. Although it is very difficult to see the difference between the model $K\hat{W}H$ and the actual KWH because of the scale, it is more obvious when looking at the autocorrelation of the error. It is well known that if a model estimates well, the error between the estimated or predicted value and the true but unknown value behaves like white noise. In other words, the optimal autocorrelation is a

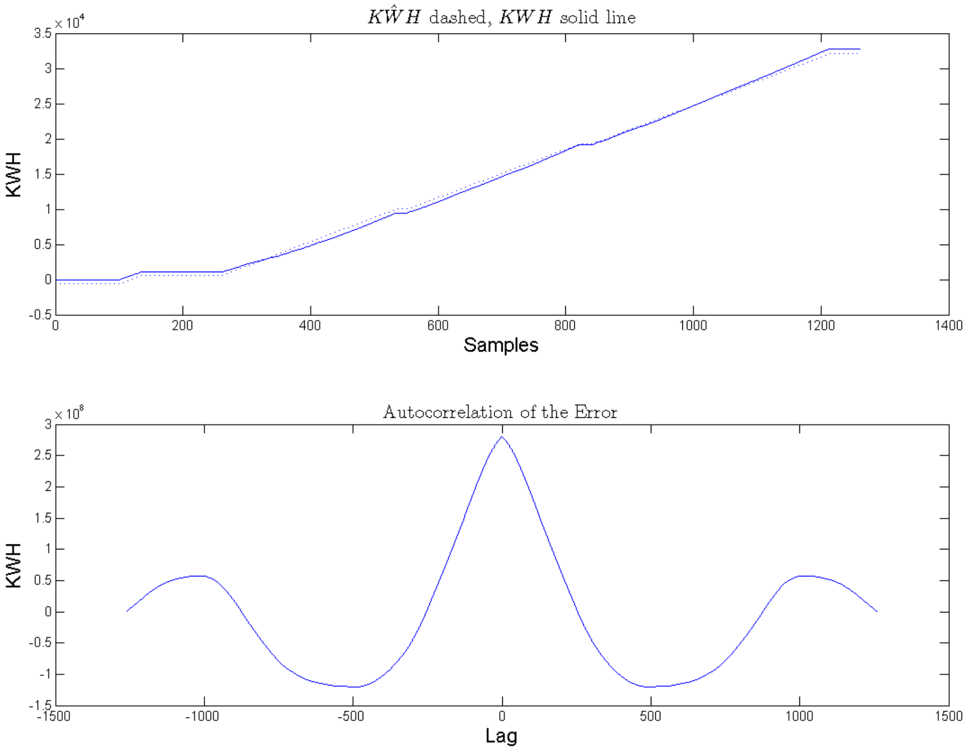
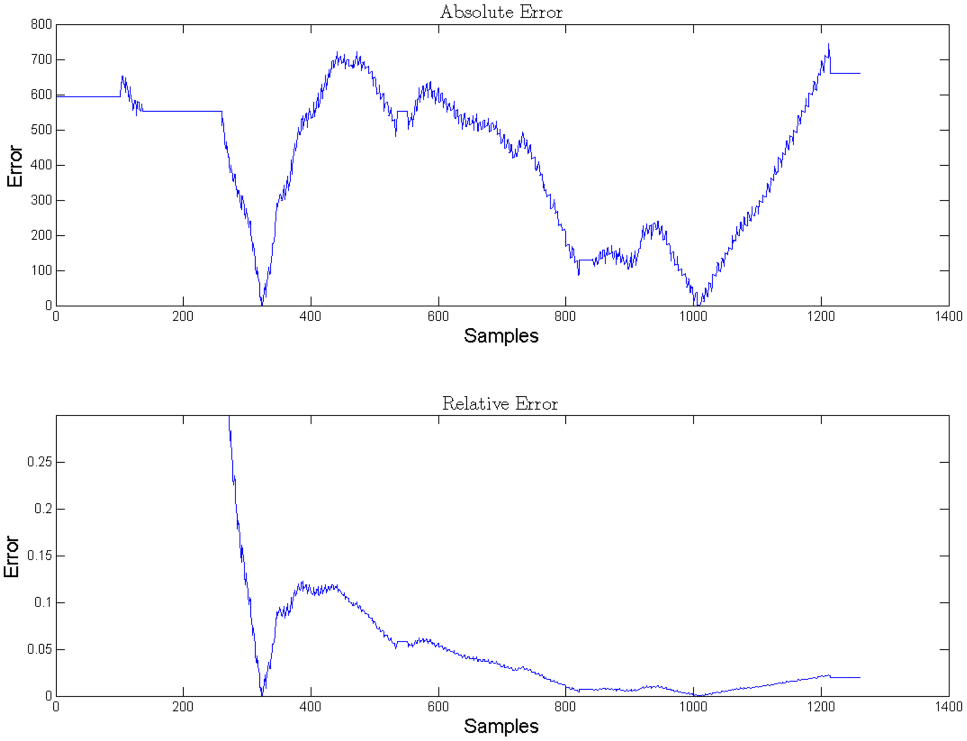


Figure 2.1: Actual *KWH* , its estimate and autocorrelation of model (1)


Absolute Error

Relative Error

Figure 2.2: Absolute and relative error of model (1)

single pulse at the origin. The smaller this white noise at non-origin and the higher the pulse at the origin, the more accurate the model. Autocorrelation is shown in the bottom half of Figure 2.1. Obviously, the system in Figure 2.1 does not behave quite like white noise. Room for improvement can also be shown by the absolute and relative errors of the model, shown in Figure 2.2. In this figure, absolute error refers to the raw difference between the model and actual data and relative error refers to the absolute error divided by the total actual KWH .

(3) $$abserr = \left| K\hat{W}H - KWH \right|$$

(4) $$relerr = \frac{\left| K\hat{W}H - KWH \right|}{KWH}$$

Another well known way to measure the performance of a model is by a quantity called Goodness of Fit (GoF). This is shown for the KWH model of (1).

(5) $$GoF = 1 - \sqrt{\sum_{n=1}^{N} \frac{\left(KWH(n) - K\hat{W}H(n)\right)^2}{\left(KWH(n) - avg\{KWH(n)\}\right)^2}}$$

Here N is the final sample in the heat. The GoF of the model for the heat file 29965 is 0.95647. Ideally the GoF should be as close to 1 as possible.

This shows that while the model is somewhat accurate, there is definitely room for absolute error improvement, and while the relative error tends to hang between one to ten percent off, that may be able to be improved as well.

There are now several choices that could possibly improve this old model. One solution would be to change the linear model to a non-linear model. To test whether a non-linear model is truly necessary, KWH was plotted versus its defining quantities. As P_{on} is the only quantity, it was plotted vs. KWH . The results presented in Figure 2.3

shows that although the KWH equation is not quite linear, it is very close. Therefore it eliminates the necessity of using a much more complicated non-linear equation.

One of the first and most significant improvements made to the model was changing it from a finite impulse response (FIR) model to an infinite impulse response (IIR) model. This is done by replacing the offset portion of the old model (the 1s) with the previous value of KWH . Therefore, the new model would be

(6) $$K\hat{W}H(n) = \theta_1 K\hat{W}H(n-1) + \theta_2 P_{on}(n-1)$$

This model has several advantages. Rather than needing to calculate the entire next step, the new model only needs to calculate the difference between the current step and the previous step. This should result in a reduction of error, as there is a much smaller range to be calculated each step. The actual data, estimate and autocorrelation of (6) is shown in Figure 2.4. Absolute error and relative error for (6) are shown in Figure 2.5. The autocorrelation of the new model looks significantly more like white noise. At first look, it may still seem poor, but upon closer inspection, the scaling on Figure 2.4 is much smaller than Figure 2.1, indicating much less variance in the noise outside zero.
As shown in Figures 2.4 and 2.5, there is a very significant improvement associated with the new model. In terms of prediction error, the improvement is approximately a factor of 6. In terms of GoF, the model goes from 0.95647 to 0.9988. It should also be noted that the model really does not increase in complexity, as it still only contains two parameters.

The next alteration to the KWH model involved both P_{on} and the quantity I_{avg} . The explanation for I_{avg} is described at the beginning of section 2.2.2, but for the purposes here, it can be described as average current, or

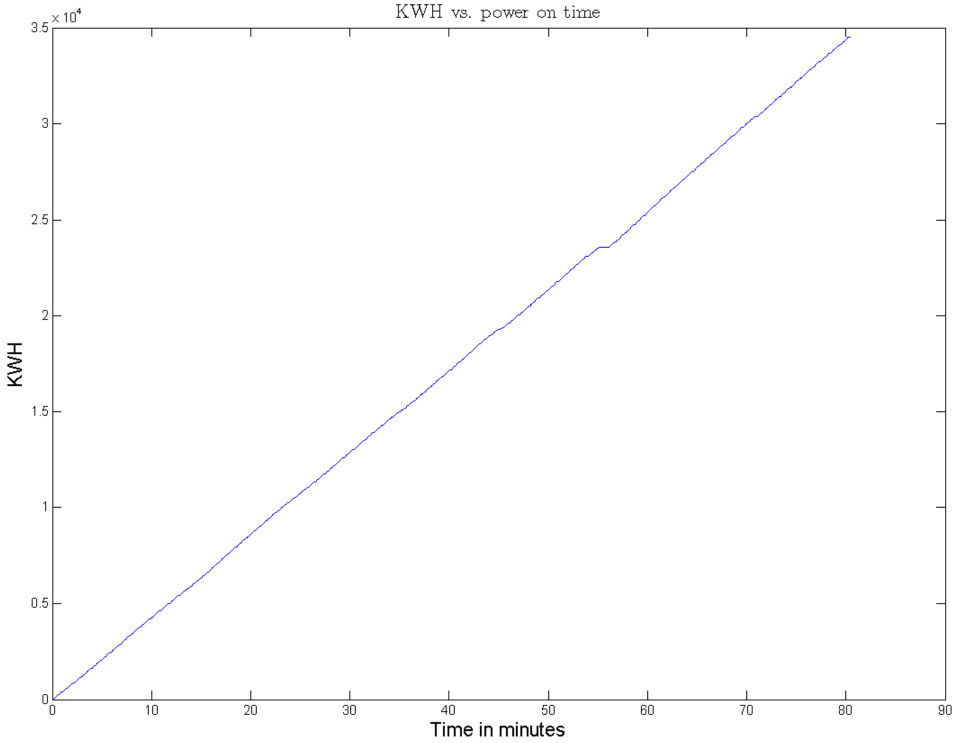


Figure 2.3: KWH vs. P_{on}

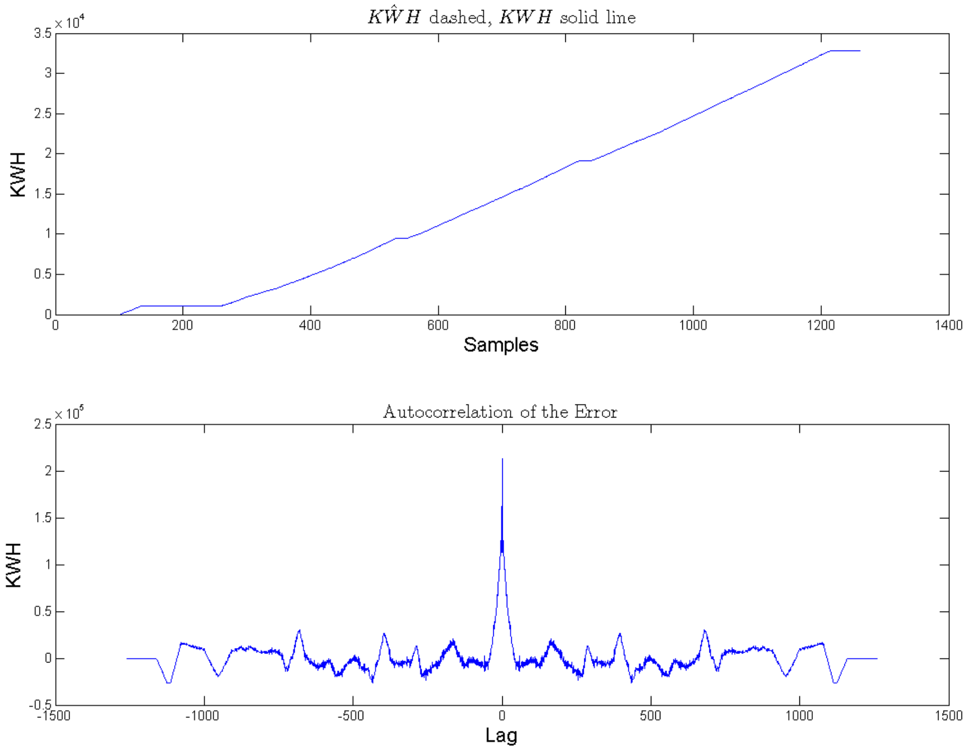


Figure 2.4: Actual KWH , its estimate and autocorrelation of model (6)

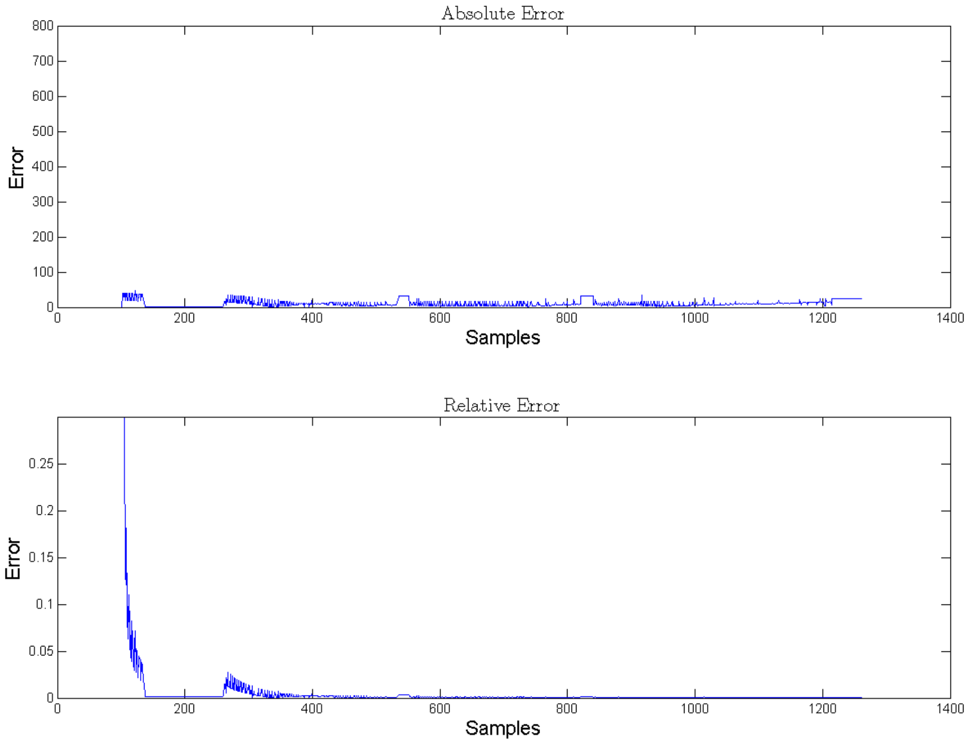


Figure 2.5: Absolute and relative error of model (6)

(7) $$I_{avg}(n) = \left(\frac{I_a(n) + I_b(n) + I_c(n)}{3} \right)$$

Notice that whenever the power is on in the EAF, current is flowing and melting the steel. Therefore, P_{on} can only be active when I_{avg} is active and vice versa. However, while P_{on} can only be on or off, I_{avg} can have many different values, better reflecting the total power being used. Therefore, it was decided to replace P_{on} with I_{avg}, in order to provide an even more descriptive identifier for KWH. Therefore, equation (8) is used

(8) $$K\hat{W}H(n) = \theta_1 K\hat{W}H(n-1) + \theta_2 \left(\frac{I_a(n-1) + I_b(n-1) + I_c(n-1)}{3} \right)$$

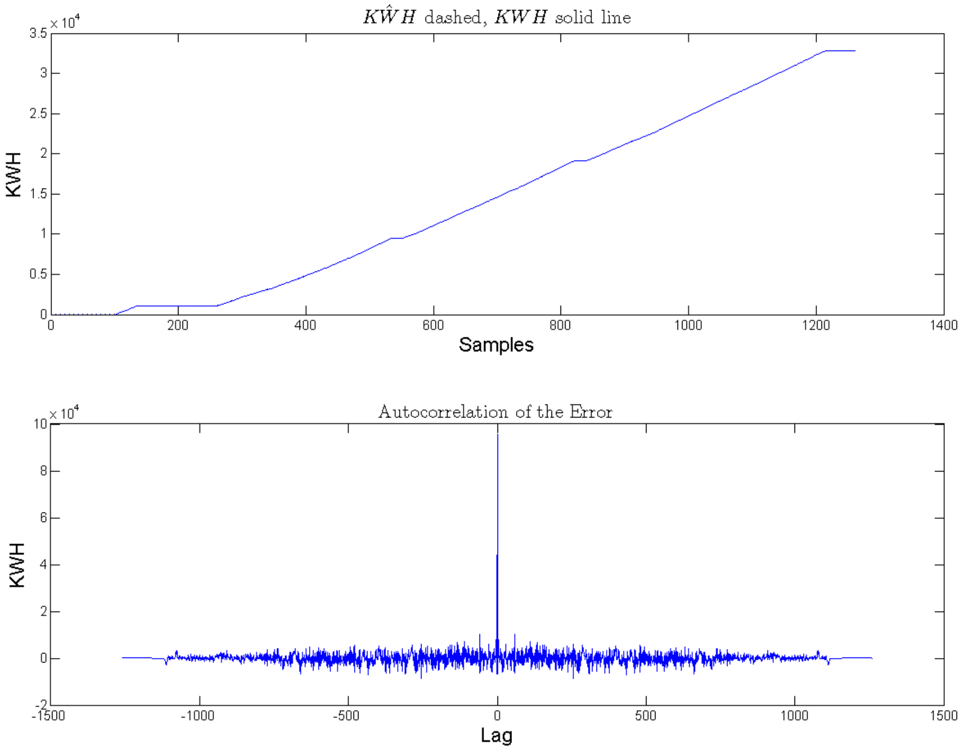
The actual data, estimate and autocorrelation of (8) is shown in Figure 2.6. Absolute error and relative error for (8) are shown in Figure 2.7. It is simple to see that the model is improved by this alteration. The GoF has further increased to 0.9992.

2.2.2 Electrode Consumption (I^2h)

Electrode consumption accounts for a surprising amount of an EAF's total costs. It is estimated by Small-Scale Steelmaking [6] that the electrode consumption can comprise of up to %20 of a typical heat's production expense, not including scrap and labor. Therefore, it is an important metric to model so that it can later be minimized. Its original model is shown as [2]

(9) $$I^2\hat{h}(n) = \theta_1 P_{on}(n-1) + \theta_2 I_a(n-1) + \theta_3 I_b(n-1) + \theta_4 I_c(n-1)$$

First, notice that the input to model (9) has three separate terms, I_a, I_b and I_c, which are the three phase currents. By checking the values of the three currents against each other, it was found that they were more or less the same. The three phase currents



$K\hat{W}H$ dashed, KWH solid line

Autocorrelation of the Error

Figure 2.6: Actual *KWH* , its estimate and autocorrelation of model (8)

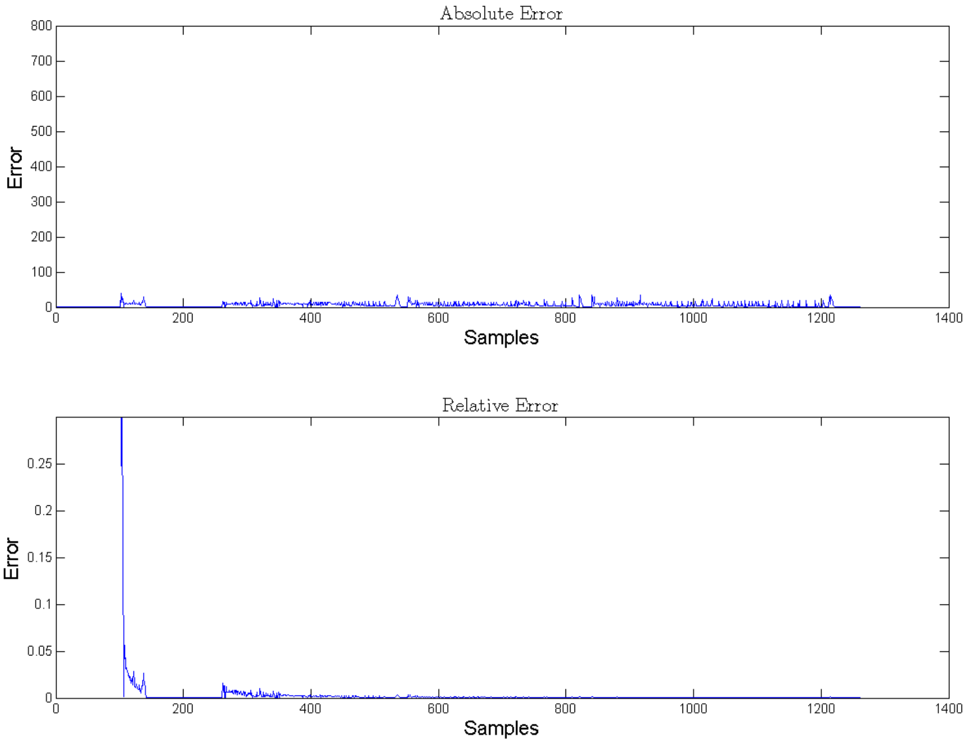


Figure 2.7: Absolute and relative error of model (8)

are shown in Figure 2.8. Unless it is desirable to control these currents separately, they can be grouped together. At this point, we assume that the three currents can be averaged into one parameter, which results in

$$(10) \qquad I^2\hat{h}(n) = \theta_1 P_{on}(n-1) + \theta_2\left(\frac{I_a(n-1) + I_b(n-1) + I_c(n-1)}{3}\right)$$

The actual data, estimate and autocorrelation of the previous model (9) is shown in Figure 2.9. Absolute and relative error for (9) are shown in Figure 2.10. Figures 2.11 and 2.12 show autocorrelation and errors likewise, but use model (10).

From these figures, it is safe to say that model (9) and model (10) perform similarly. The original model's (9) GoF is 0.98036, and model (10) has a GoF of 0.97996. The real benefit is that the complexity of the I^2h state equation is reduced from 4 parameters to 2.

Similarly to the KWH model, the I^2h model should be able to be further improved by changing the original model from an FIR system to an IIR system. This is achieved by adding $I^2\hat{h}(n-1)$ to model (10), which results in

$$(11) \qquad I^2\hat{h}(n) = \theta_1 I^2\hat{h}(n-1) + \theta_2 P_{on}(n-1) + \theta_3\left(\frac{I_a(n-1) + I_b(n-1) + I_c(n-1)}{3}\right)$$

This should decrease error by decreasing the range that needs to be calculated every step. The actual data, estimate and autocorrelation of (11) is shown in Figure 2.13. Absolute and relative errors for (11) are shown in Figure 2.14. As expected, this change is very visible if one compares autocorrelation or error between the IIR and FIR models. The alteration reduces error in the model by a factor of about 8. The GoF is greatly increased to 0.99938. The final adjustment to the I^2H model involved the correlation

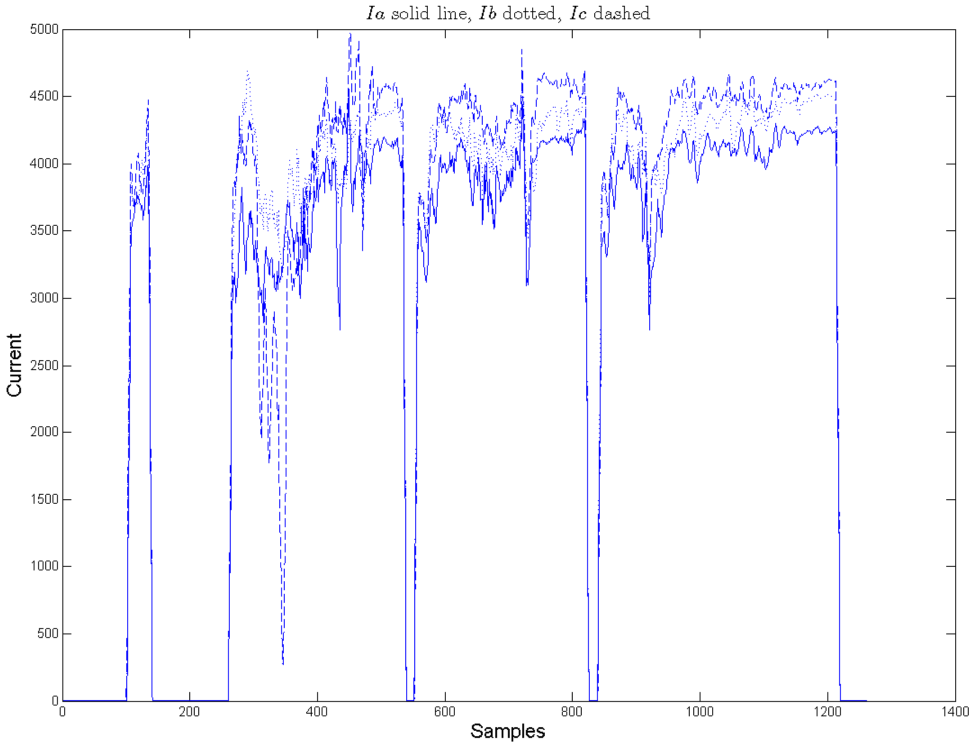


Figure 2.8: The three phase currents

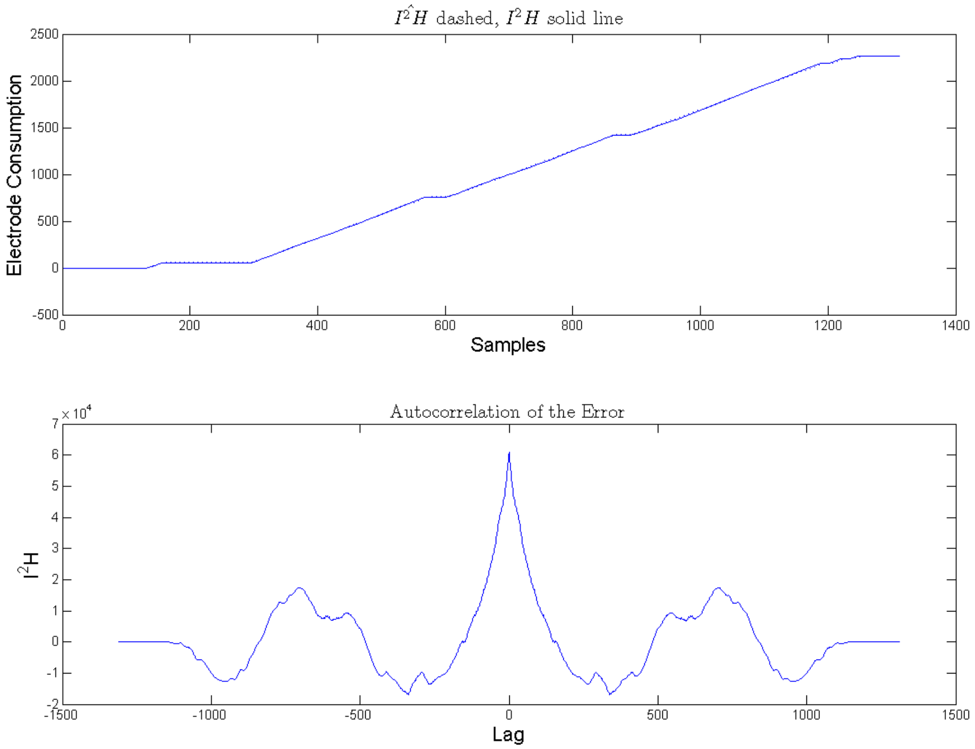


Figure 2.9: Actual I^2H , its estimate and autocorrelation of model (9)

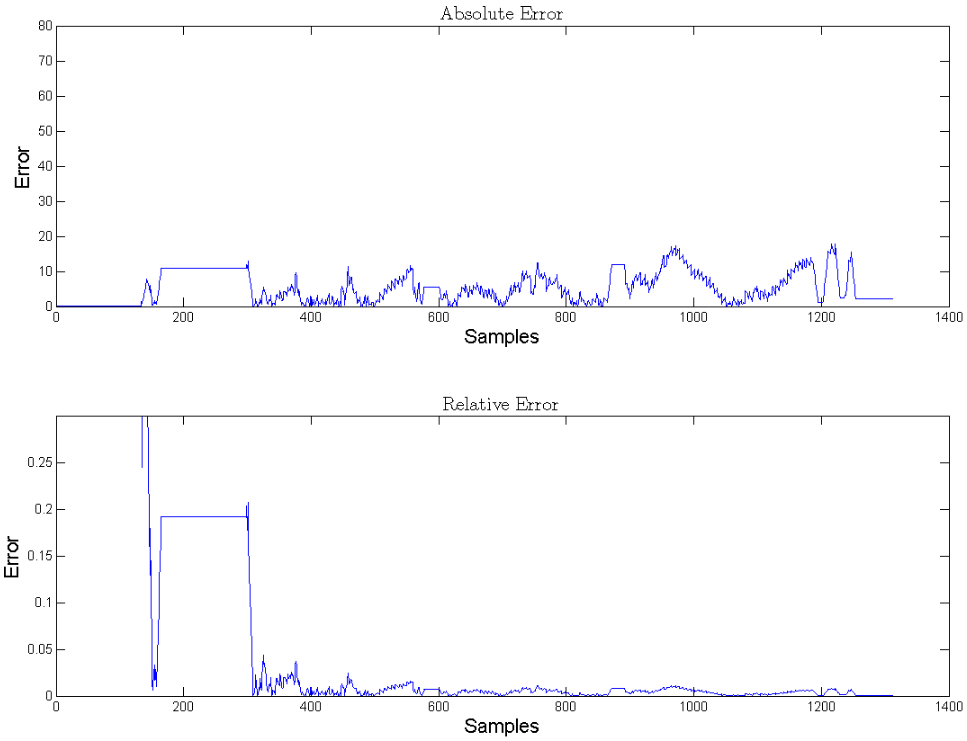
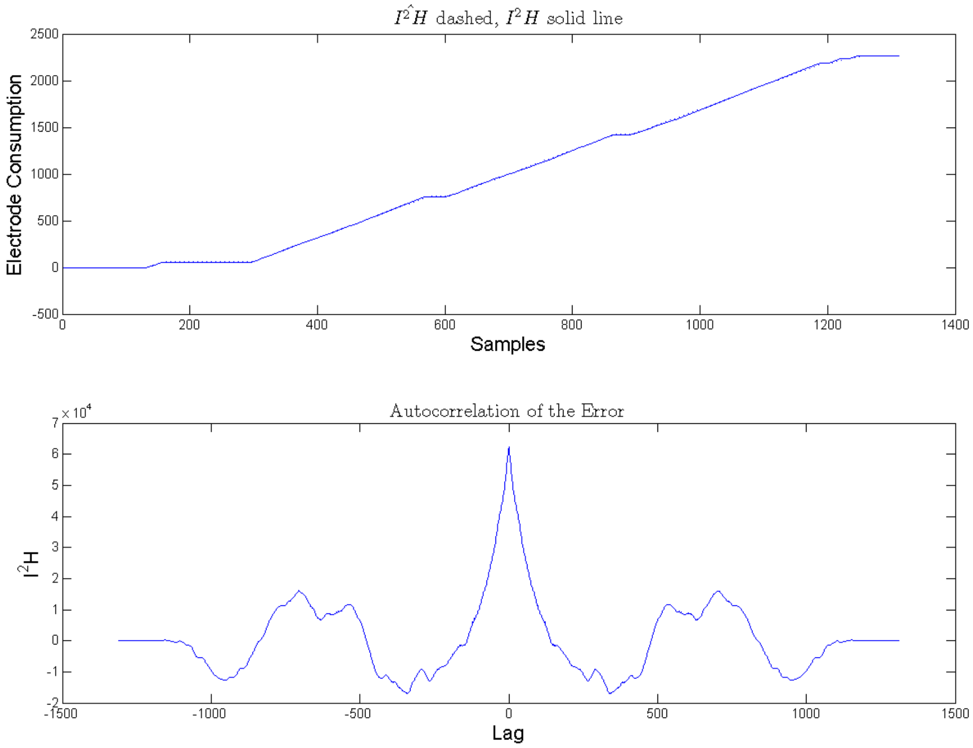


Figure 2.10: Absolute and relative error of model (9)


$\hat{I^2}H$ dashed, I^2H solid line

Autocorrelation of the Error

Figure 2.11: Actual I^2H , its estimate and autocorrelation of model (10)

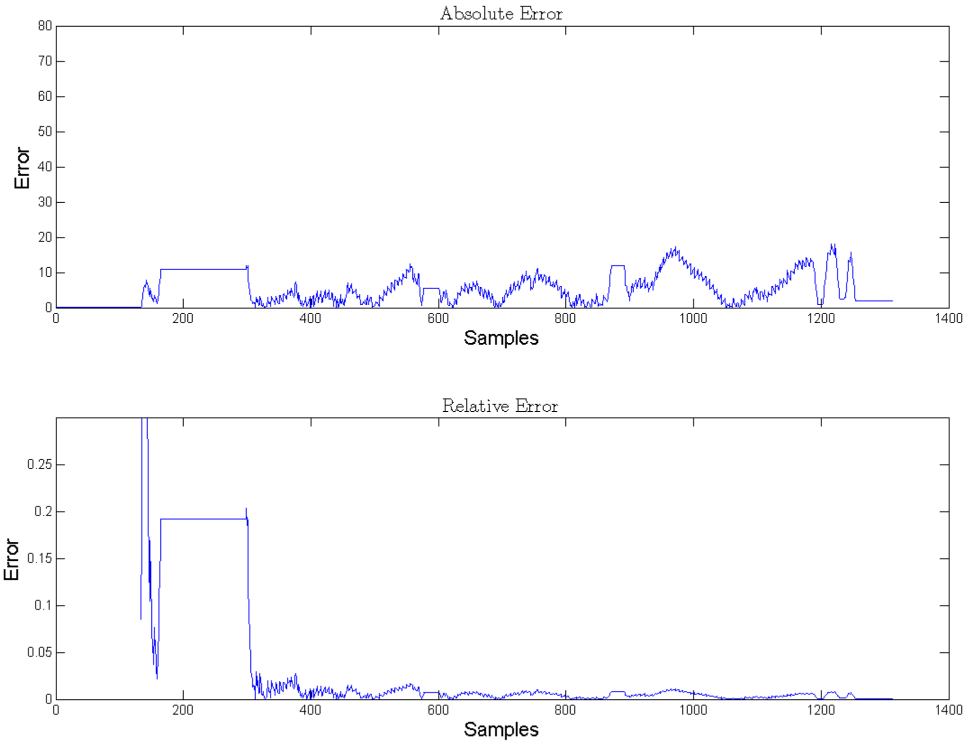


Figure 2.12: Absolute and relative error of model (10)

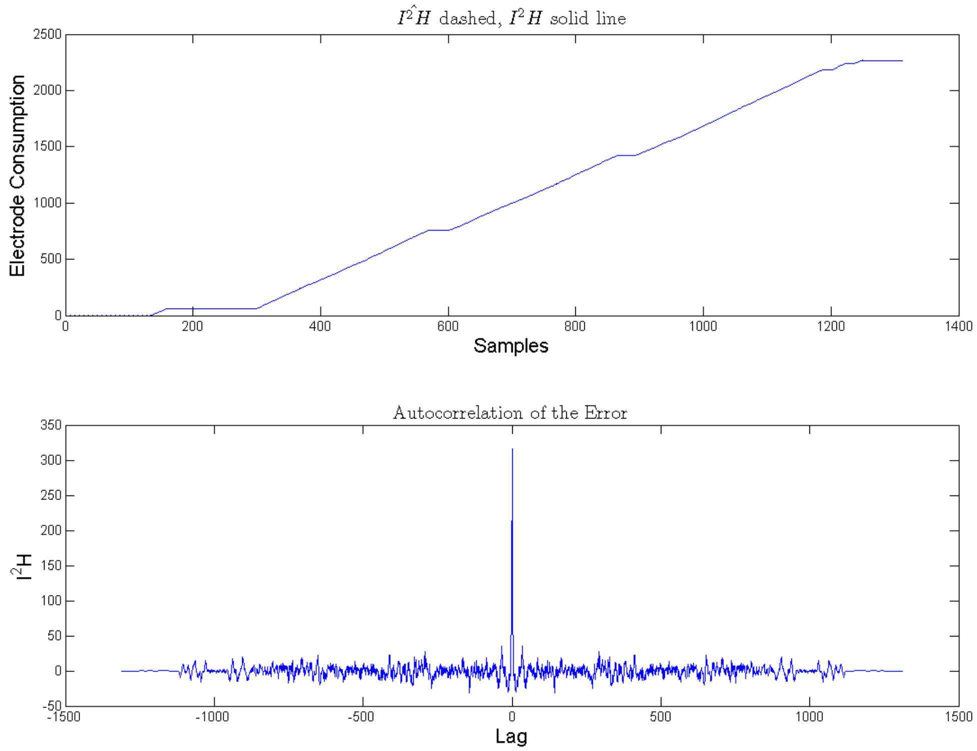


Figure 2.13: Actual I^2H , its estimate and autocorrelation of model (11)

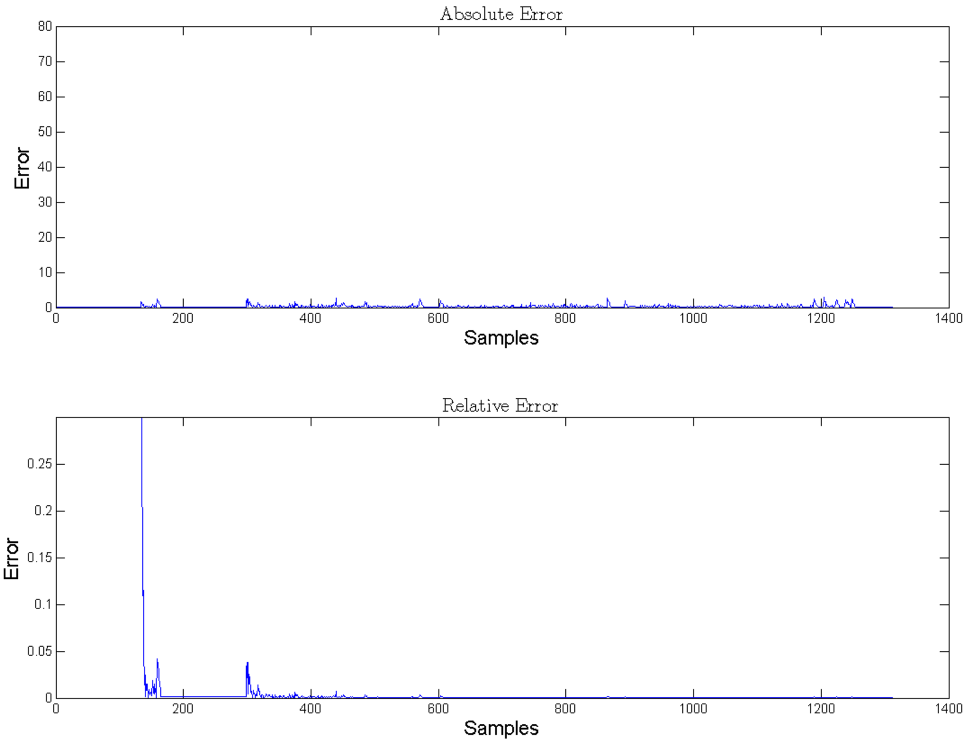


Figure 2.14: Absolute and relative error of model (11)

between P_{on} and I_{avg}. In the same manner and for the same reason as the KWH model, P_{on} was replaced with I_{avg}. As there is now 2 separate I_{avg} 's, only one is necessary. Therefore, the I^2H model was altered to be

$$I^2\hat{h}(n) = \theta_1 I^2\hat{h}(n-1) + \theta_2\left(\frac{I_a(n-1) + I_b(n-1) + I_c(n-1)}{3}\right) \tag{12}$$

The actual data, estimate and autocorrelation of (12) is shown in Figure 2.15. The GoF stayed at 0.99938. It should be emphasized that while the model has had its accuracy greatly improved, its complexity has actually decreased from the original model as well.

2.2.3 Percent Melted (PM)

The percent melted state equation is the most complex of the three equations. This is because its purpose is to calculate how much steel each quantity actually melts, and this can depend on a large number of factors. Clearly, percent melted does not have a cost associated with it, but determines how much of the other parameters are necessary to melt one charge of steel. It is important to realize that percent melted represents the percentage of a single particular charge that is melted. This means that if there is already one charge melted in the furnace, the second charge will be dropped in starting with a percent melted of 0, no matter the state of the first charge. This can become a bit confusing as the final charge may be completely melted for a while before it becomes %100. This is because there is a refining phase that occurs during the final charge of every heat that requires more heat than the previous two charges. Therefore, percent melted should be looked at in a sense more as progress rather than how liquid the steel is. The previous percent melted equation is shown as model (13) [2].

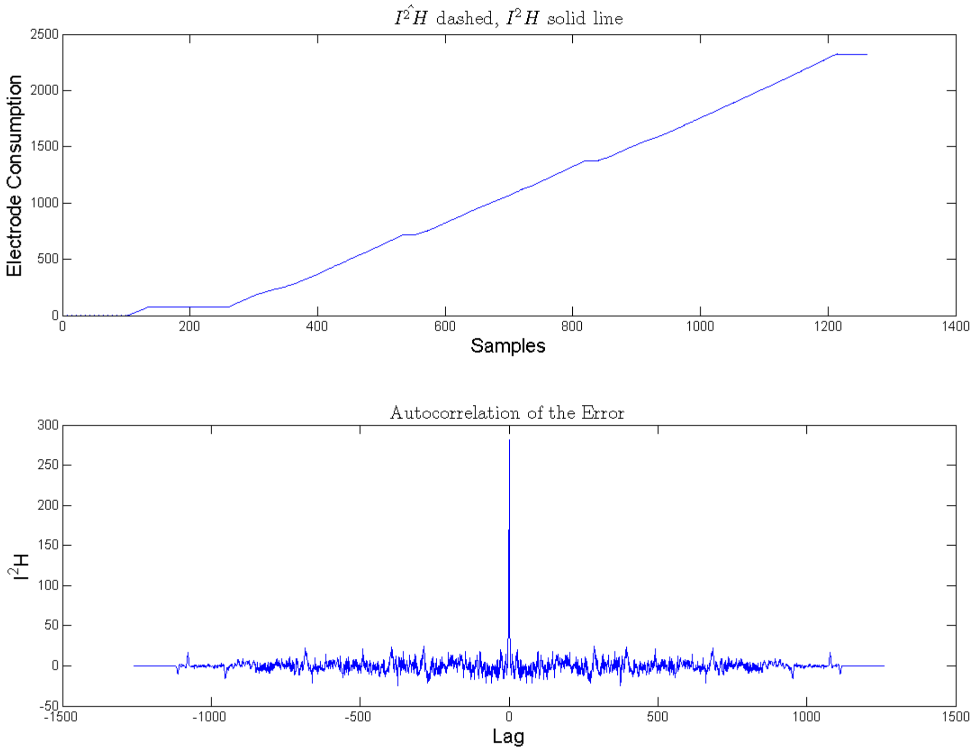


Figure 2.15: Actual I^2H , its estimate and autocorrelation of model (12)

$$
\begin{aligned}
P\hat{M}(n) = {} & \theta_1 \frac{ChargeKWH(n-1)}{ChargeSize} + \theta_2 \frac{HeatKWH(n-1)}{HeatSize} \\
& + \theta_3 \frac{ChargeO_2(n-1)}{ChargeSize} + \theta_4 \frac{HeatO_2(n-1)}{HeatSize} + \theta_5 \frac{ChargeGas(n-1)}{ChargeSize} \\
& + \theta_6 \frac{HeatGas(n-1)}{HeatSize} + \theta_7 \frac{ChargeCarbon(n-1)}{ChargeSize} + \theta_8 \frac{HeatCarbon(n-1)}{HeatSize} \\
& + \theta_9 P_{on}(n-1) + \theta_{10} P_{off}(n-1)
\end{aligned}
\tag{13}
$$

It is important to realize that in the percent melted case, there is no precise way to measure how much steel is actually melted at any time. The given data is estimating that a charge of steel is %100 melted when there has been 250 KWH used per charge ton. As noted before, KWH accounts for %70-80 of the melting cycle, so this is a good metric to base percent melted on. However, as there is still some percentage left unaccounted for. Nearly every charge is not considered melted by the EAF operators until the raw percent melted metric reaches somewhere above %100. Typically for charges 1 and 2, the raw percent melted metric reaches between %110 and %120. Charge 3 contains the refining phase, which means that while the charge may be entirely melted, the EAF will continue to operate until the steel has been alloyed and heated up to the specific grade standards. This usually brings the raw percent melted metric to around %250 and often times can get as high as %300. For this reason, the raw percent melted metric is normalized so that every charge is finished when the new percent melted metric reaches %100. A typical raw percent melted heat is shown in Figure 2.16. The normalized version is shown in Figure 2.17. Model (13) performed reasonably well for such a complex equation. The actual data, estimate and autocorrelation of (13) is shown in Figure 2.18. Absolute and relative errors for (13) are shown in Figure 2.19. The spikes in the error figures are due to the percent melted going directly from 100 to zero in one sample. Also, there should be two equidistant spikes expected in the autocorrelation, as there are 3 very similar,

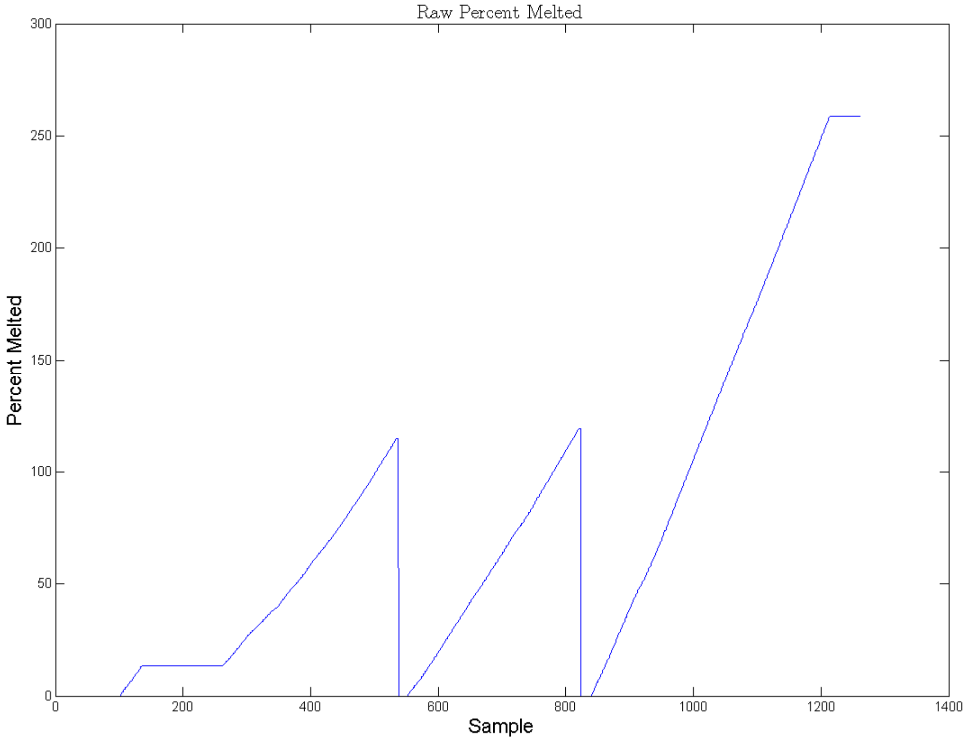


Figure 2.16: Raw percent melted

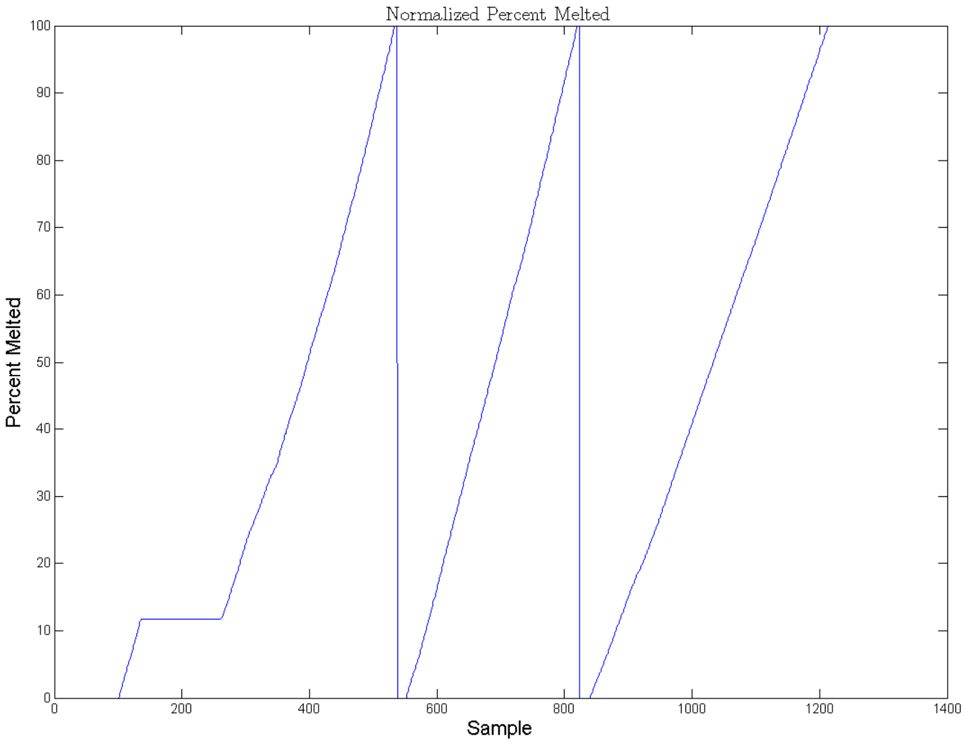
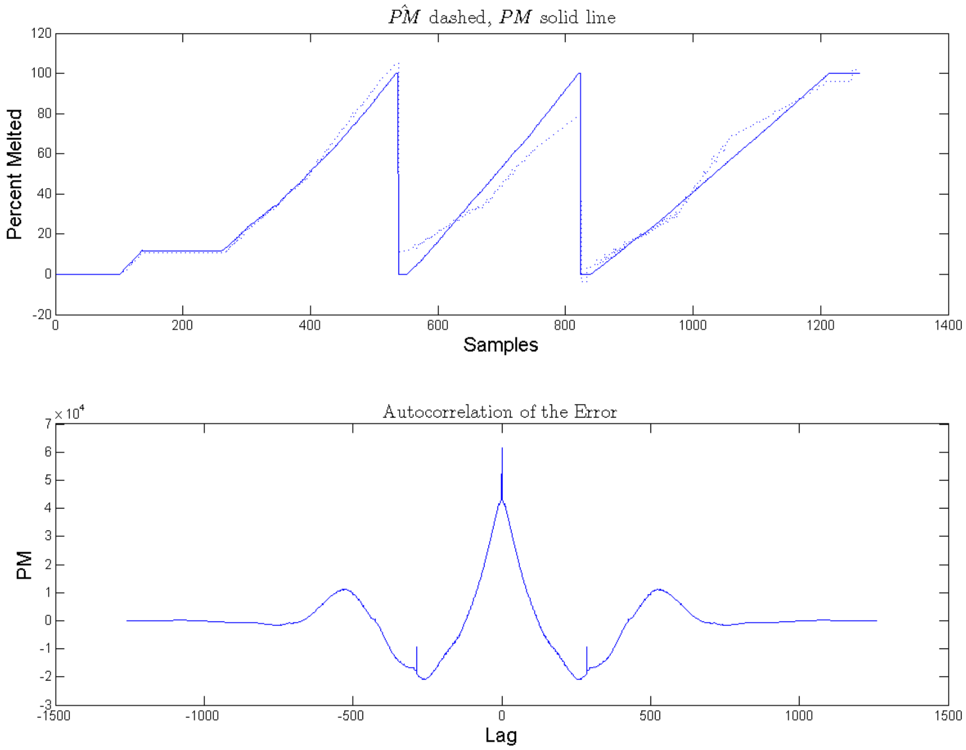


Figure 2.17: Normalized percent melted



$\hat{PM}$ dashed, PM solid line

Autocorrelation of the Error

Figure 2.18: Actual PM , its estimate and autocorrelation of model (13)

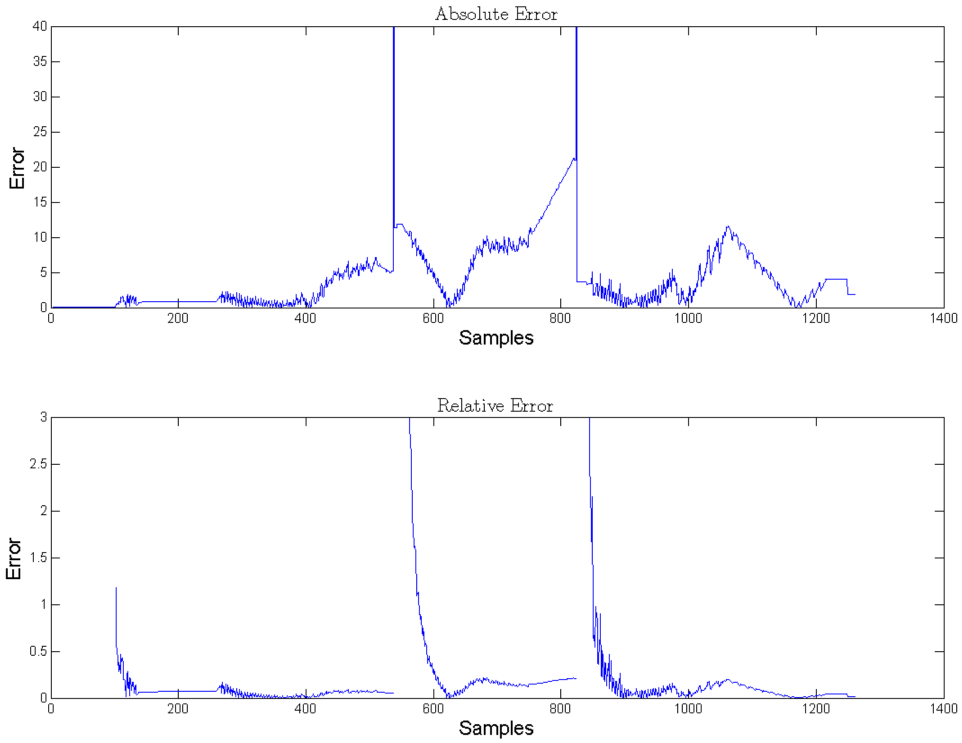


Figure 2.19: Absolute and relative error of model (13)

nearly periodic charges. As the error and autocorrelation show, the model has more room for improvement than the other 2 models. The first and most significant improvement, like the KWH and I^2h models, is changing the system from FIR to IIR. This results in

$$
\begin{aligned}
&P\hat{M}(n) = \theta_1 P\hat{M}(n-1) + \theta_2 \frac{Ch\arg eKWH(n-1)}{Ch\arg eSize} + \theta_3 \frac{HeatKWH(n-1)}{HeatSize} \\
(14)\qquad &+ \theta_4 \frac{Ch\arg eO_2(n-1)}{Ch\arg eSize} + \theta_5 \frac{HeatO_2(n-1)}{HeatSize} + \theta_6 \frac{Ch\arg eGas(n-1)}{Ch\arg eSize} \\
&+ \theta_7 \frac{HeatGas(n-1)}{HeatSize} + \theta_8 \frac{Ch\arg eCarbon(n-1)}{Ch\arg eSize} + \theta_9 \frac{HeatCarbon(n-1)}{HeatSize} \\
&+ \theta_{10} P_{on}(n-1) + \theta_{11} P_{off}(n-1)
\end{aligned}
$$

This model is a huge improvement over its predecessor. The GoF went from 0.77451 to 0.87897, a very significant improvement. The actual data, estimate and autocorrelation of (14) is shown in Figure 2.20. Absolute and relative errors for (14) are shown in Figure 2.21. As shown, the absolute error is reduced roughly by a factor of 10. At this point, the model performs very well. The problem is that it involves many variables and some are closely correlated. It is more or less a matter of simplifying it as much as possible while maintaining the accuracy. It is an easy assumption to say that $\frac{Ch\arg eO_2(n-1)}{Ch\arg eSize}$ and $\frac{HeatO_2(n-1)}{HeatSize}$ are closely correlated and so are $\frac{Ch\arg eGas(n-1)}{Ch\arg eSize}$ and $\frac{HeatGas(n-1)}{HeatSize}$, $\frac{Ch\arg eCarbon(n-1)}{Ch\arg eSize}$ and $\frac{HeatCarbon(n-1)}{HeatSize}$, and $\frac{Ch\arg eKWH(n-1)}{Ch\arg eSize}$ and $\frac{HeatKWH(n-1)}{HeatSize}$. Each physical variable is provided in both charge and heat forms. It seems that one or the other could be removed without greatly affecting the model. Therefore, was attempted as an alternative percent melted model.

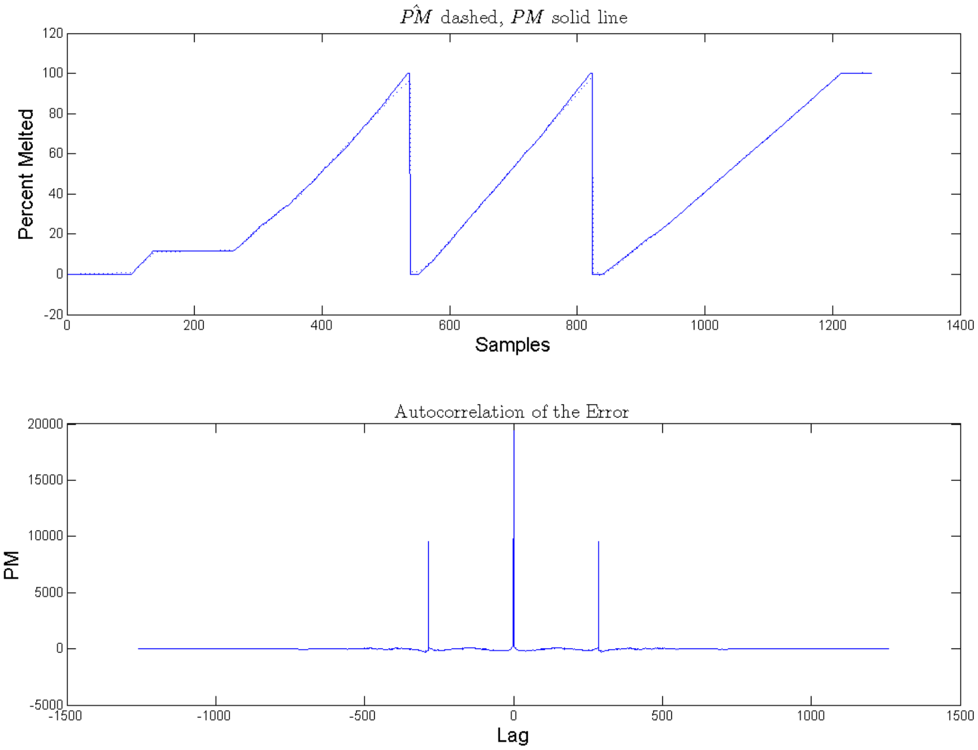


Figure 2.20: Actual PM , its estimate and autocorrelation of model (14)

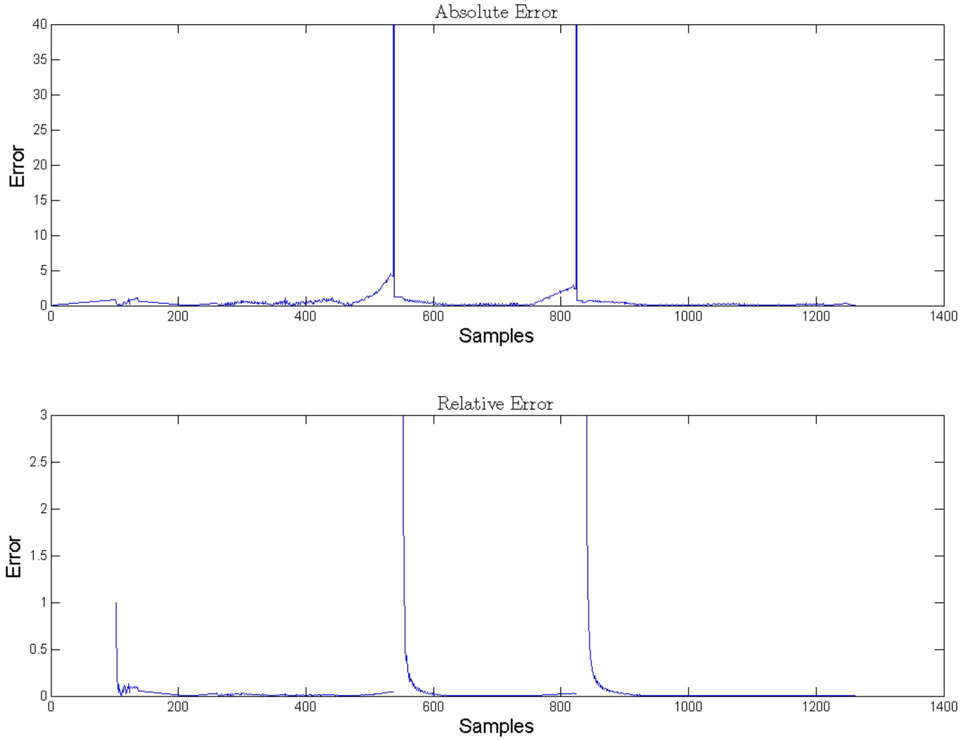


Figure 2.21: Absolute and relative error of model (14)

$$\begin{aligned}(15)\quad & P\hat{M}(n)=\theta_1 P\hat{M}(n-1)+\theta_2\frac{HeatKWH(n-1)}{HeatSize}+\theta_3\frac{HeatO_2(n-1)}{HeatSize}\\ & +\theta_4\frac{HeatGas(n-1)}{HeatSize}+\theta_5\frac{HeatCarbon(n-1)}{HeatSize}+\theta_6 P_{on}(n-1)+\theta_7 P_{off}(n-1)\end{aligned}$$

The actual data, estimate and autocorrelation of (15) is shown in Figure 2.22. Its GoF stays similar at 0.8785. When compared, the difference in the models is negligible, so the newer, simpler model is preferred.

One more simple alteration can be made to improve the percent melted model. Since the power of the EAF is either on or off, P_{on} and P_{off} can be calculated from each other. Further, P_{off} would not likely contribute much to the melting process. Therefore, the model was changed to

$$\begin{aligned}(16)\quad & P\hat{M}(n)=\theta_1 P\hat{M}(n-1)+\theta_2\frac{HeatKWH(n-1)}{HeatSize}+\theta_3\frac{HeatO_2(n-1)}{HeatSize}\\ & +\theta_4\frac{HeatGas(n-1)}{HeatSize}+\theta_5\frac{HeatCarbon(n-1)}{HeatSize}+\theta_6 P_{on}(n-1)\end{aligned}$$

The GoF stays constant at 0.8785. The model and autocorrelation is shown for the same heat as the previous percent melted models in Figure 2.20. The next alteration performed was changing P_{on} to be I_{avg}. This was done for the same reasons as explained in section 2.2.1.

$$\begin{aligned}(17)\quad & P\hat{M}(n)=\theta_1 P\hat{M}(n-1)+\theta_2\frac{HeatKWH(n-1)}{HeatSize}+\theta_3\frac{HeatO_2(n-1)}{HeatSize}\\ & +\theta_4\frac{HeatGas(n-1)}{HeatSize}+\theta_5\frac{HeatCarbon(n-1)}{HeatSize}+\theta_6 I_{avg}(n-1)\end{aligned}$$

The actual data, estimate and autocorrelation of (17) is shown in Figure 2.24. Model (17) performs similar to Model (16), as expected. There is a slight increase in accuracy, and the GoF increases to 0.87921.

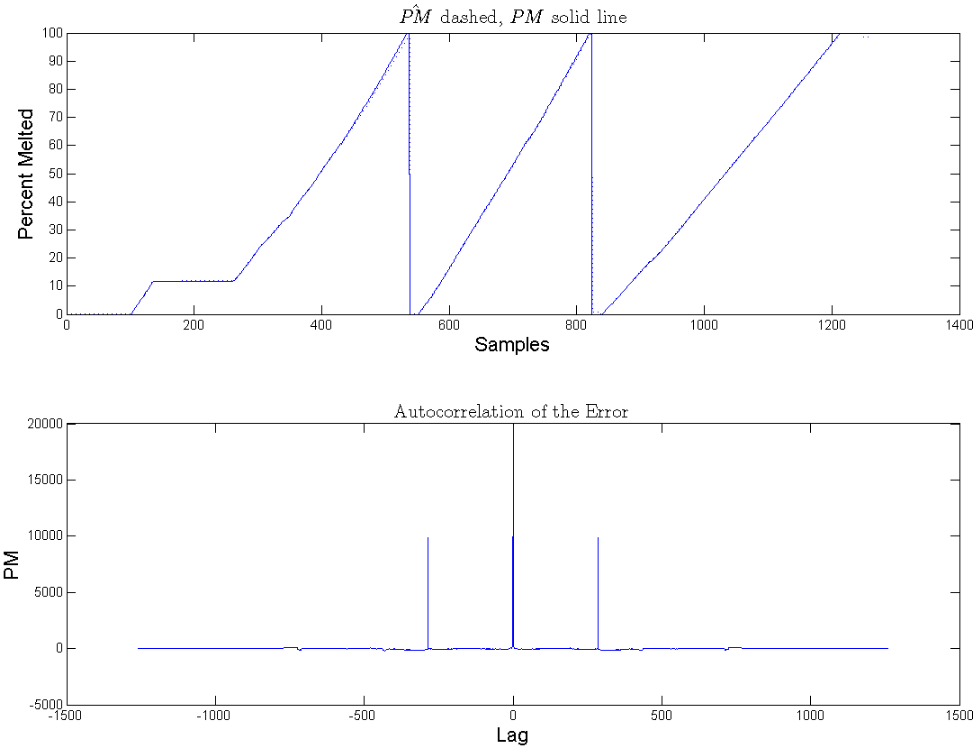


Figure 2.22: Actual PM , its estimate and autocorrelation of model (15)

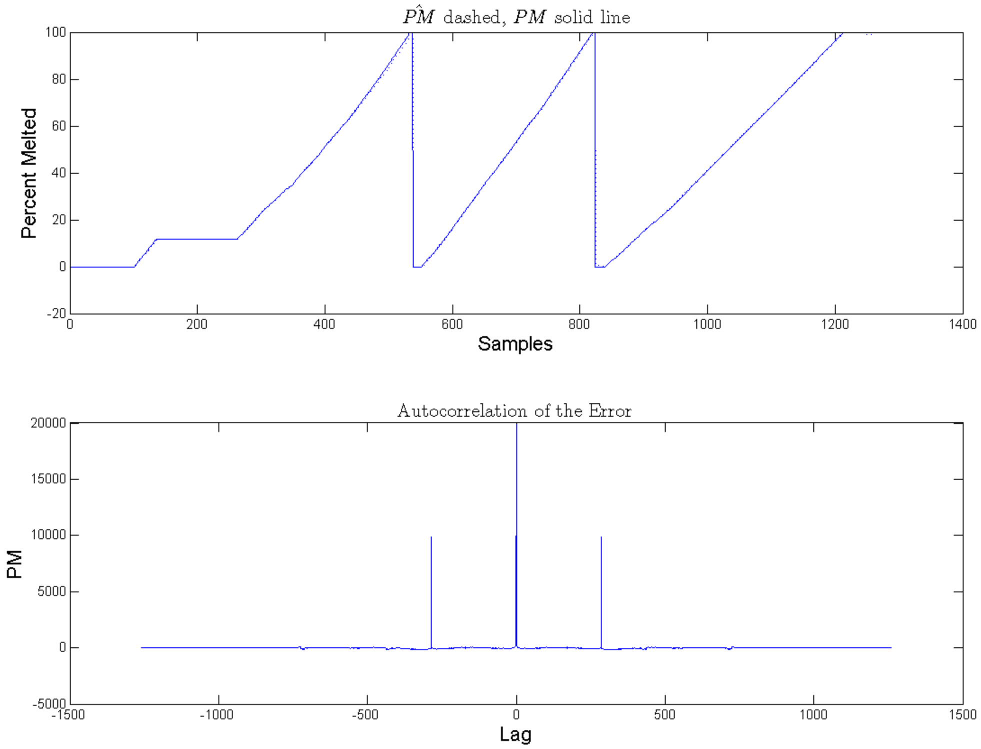
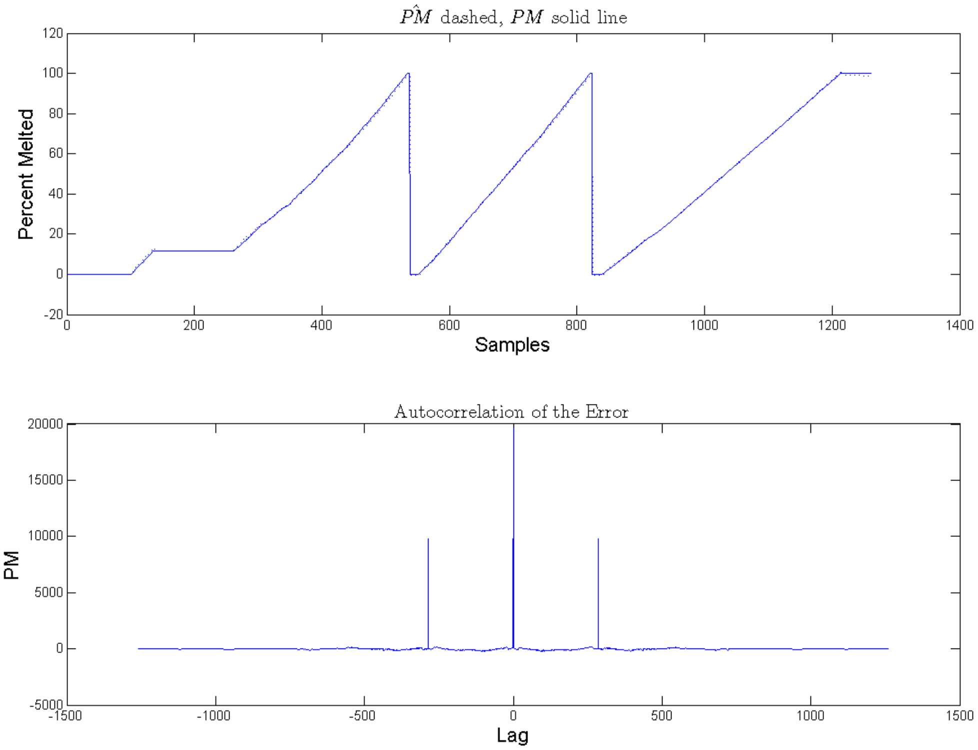


Figure 2.23: Actual PM , its estimate and autocorrelation of model (16)



$P\hat{M}$ dashed, PM solid line

Autocorrelation of the Error

Figure 2.24: Actual PM , its estimate and autocorrelation of model (17)

Again, there is very little difference if any, so the simpler model is preferred. Now the model is much more accurate, and uses only the 5 quantities listed earlier in the chapter that significantly affect the melting process.

2.3 Free Oxygen

One particular component of interest for the melting process is oxygen. The oxygen that is used to melt the steel comes from a number of places. The furnace is not air-tight, so some of the oxygen is naturally provided by air. There are large quantities of iron-oxide, or rust, contained in the scrap. Differences in amounts of iron-oxide can have a sizeable impact on the melting process. The last and most important provider of oxygen is the burners, which are controllable. Part of the injected oxygen is designed to directly assist in melting the steel. Another portion of the oxygen is injected with the intended purpose of completely combusting the gases and excess carbon so heat is not wasted by having gases burned in the exhaust shaft. Therefore, it was decided that a variable called free oxygen, or $FreeO_2$, should be calculated to find the amount of oxygen that directly melts the steel rather than is consumed by burning gas and carbon. $FreeO_2$ would then replace the regular oxygen quantity in the percent melted model. In previous work, it was suggested that that equation (18) be used, where θ_1 was suggested to be 2.3 and θ_2 was suggested to be 13 [1].

(18) $$FreeO_2(n) = TotalO_2(n) - \theta_1 TotalGas(n) - \theta_2 TotalCarbon(n)$$

Figure 2.25 shows a heat of unaltered oxygen, while Figure 2.26 shows the same heat's $FreeO_2$. This equation had some surprising effects. As shown, $FreeO_2$ is almost entirely negative. Since negative oxygen is clearly impossible, it can be inferred that the gas and carbon injected into the furnace are not getting enough oxygen to completely combust. If this were to happen, it would mean that some of the gas and carbon are being released into the exhaust shaft and combusted there, which would be wasteful. However, this assumption is complicated by the fact that each batch of steel carries a different and unknown amount of iron-oxide which would further alter the oxygen statistics. Many different solutions to this negative $FreeO_2$ were attempted, and eventually equation (19) was settled on as a reasonable solution. This equation is the same as equation (18), but α_1 and α_2 were assigned as 0.5 and 1.5, respectively.

(19) $FreeO_2(n) = TotalO_2(n) - \alpha_1\theta_1 TotalGas(n) - \alpha_2\theta_2 TotalCarbon(n)$

This new equation was created to better reflect the fact that gas likely escapes through the exhaust shaft more easily than the carbon dust. It also works much better with the percent melted model than the previous $FreeO_2$ equation. Figure 2.27 shows $FreeO_2$ for the same heat using equation (19). Notice that the oxygen is positive at all times. From here on in the thesis, the parameter O_2 is assumed to be the $FreeO_2$ used in equation (19).

2.4 Negative Coefficients

It was discovered when re-deriving the original model that there were a number of negative coefficients calculated by the least squares estimator. This was a problem for the previous optimization attempt, even as it used a genetic algorithm the corresponding control had to be manually adjusted. Since a goal of this thesis is to make an EAF model that makes physical sense, all coefficients should be positive. If one of the coefficients were to be negative in the percent melted model’s case, that would mean the parameter

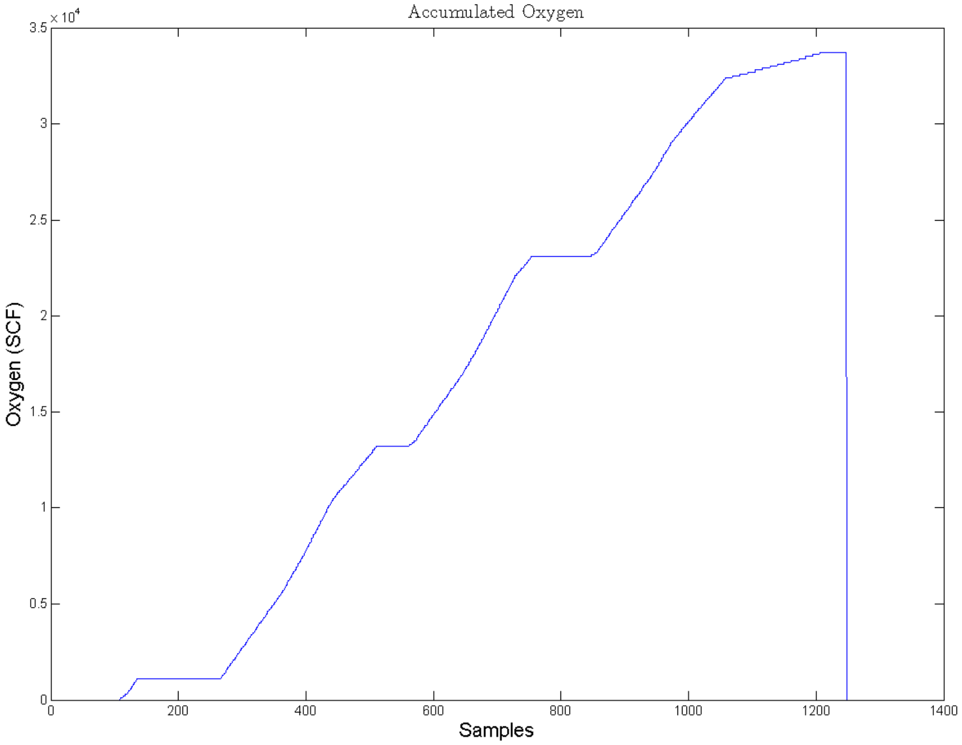


Figure 2.25: Accumulated oxygen

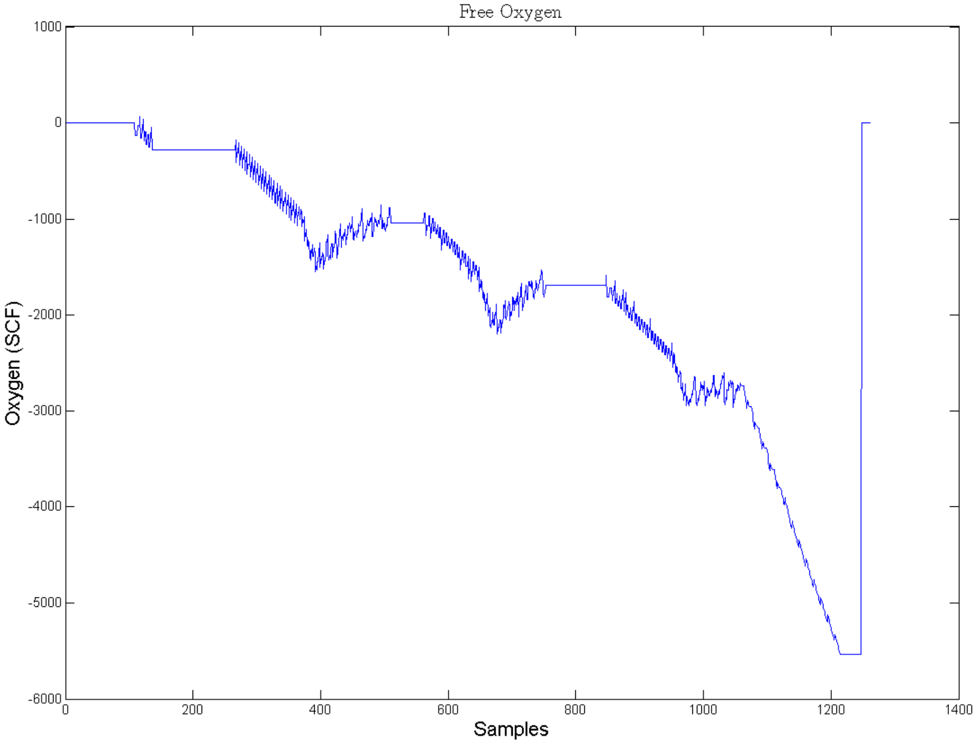


Figure 2.26: $FreeO_2$ using equation (18)

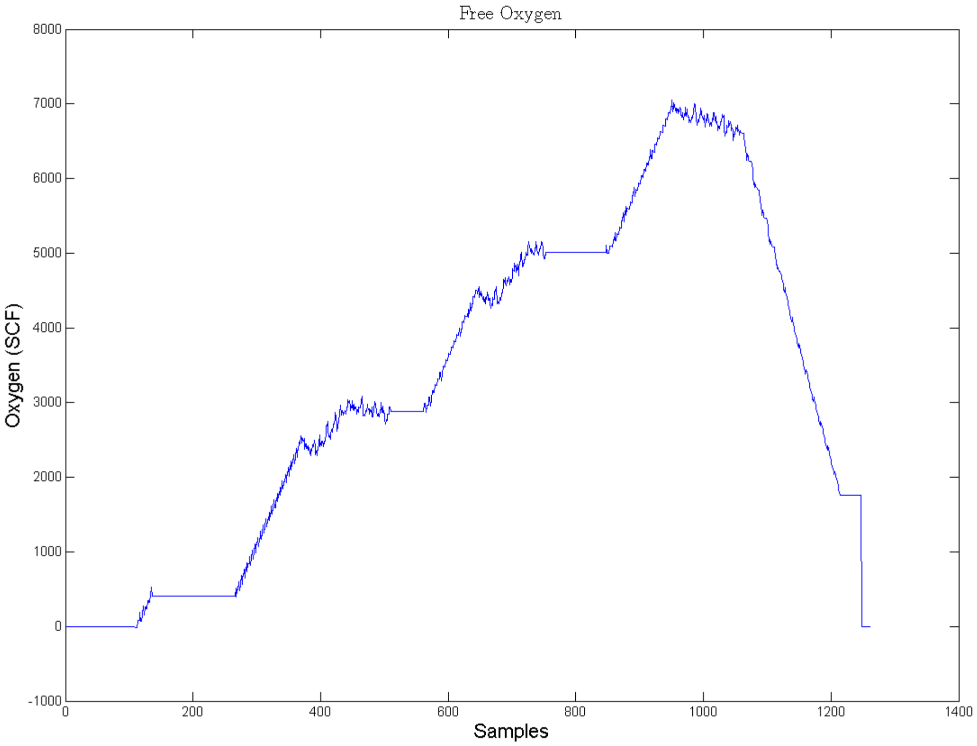


Figure 2.27: $FreeO_2$ using equation (19)

with the negative coefficient would actually decrease the overall percent melted. While this could make perfect mathematical sense and a good model, it makes no physical sense. Clearly, all the parameters in the percent melted model are there because they melt the steel, not cool it. The cause of the negative coefficients is not quite known. A good guess is that they are caused by the insufficient amounts of oxygen or the effects of non-linearity in the models. Whatever the cause, there cannot be any negative coefficients in a model that makes intuitive sense.

Several attempts were made to eliminate the negative coefficients. This included attempts by changing the basic parameters. All different combinations were tried of using heat parameters, charge parameters, and just basic parameters. For instance, $\frac{Ch\arg eKWH(n-1)}{Ch\arg eSize}$, $\frac{HeatO_2(n-1)}{HeatSize}$, $Gas(n-1)$, and $\frac{Ch\arg eCarbon(n-1)}{Ch\arg eSize}$ was attempted for the percent melted model. Some of these models had moderate success in eliminating the negative coefficients. However, the goal of eliminating the negative coefficients is to obtain a model that makes intuitive sense, and it does not make sense to just choose some charge parameters and some heat or other parameters. Therefore, a new method was attempted.

First, it should be noted that all the charges in the same heat do not necessarily have the same characteristics. As shown in previous figures, the last charge especially takes much more heat to achieve a percent melted of %100. For this reason, charge characteristics are desirable for the model. One way to achieve a very good model is to break the heat down into individual charges and model each charge individually. One of the advantages of modeling in this way is that there is no longer any need for keeping track of the charge and heat weights in the percent melted model. If each charge is individually modeled, it automatically takes that charge’s weight into account. After each charge has been modeled, they can be reunited as a heat. Now the percent melted model can be written as

$$\begin{aligned} P\hat{M}(n) &= \theta_1 P\hat{M}(n-1) + \theta_2 KWH(n-1) + \theta_3 O_2(n-1) \\ &+ \theta_4 Gas(n-1) + \theta_5 Carbon(n-1) + \theta_6 I_{avg}(n-1) \end{aligned} \tag{20}$$

The actual data, estimate and autocorrelation of (20) is shown in Figure 2.28. It again performs similarly to model (17).

The next change in the model was performed to simplify and ideally decrease negative coefficients. The θ_1’s, or the first coefficients from each model, were removed. This is the same as equating θ_1=1. This makes sense as each model is cumulative and should only increase. In this way, the models will include exactly the last step, with the estimated input added to it. Therefore, the models were altered to

$$K\hat{W}H(n) = K\hat{W}H(n-1) + \theta_1 I_{avg}(n-1) \tag{21}$$

$$I^2\hat{h}(n) = I^2\hat{h}(n-1) + \theta_1 I_{avg}(n-1) \tag{22}$$

$$\begin{aligned} P\hat{M}(n) &= P\hat{M}(n-1) + \theta_1 KWH(n-1) + \theta_2 O_2(n-1) \\ &+ \theta_3 Gas(n-1) + \theta_4 Carbon(n-1) + \theta_5 I_{avg}(n-1) \end{aligned} \tag{23}$$

To further prepare the model to be used in an optimal control situation, all the input parameters for each model were altered to be “delta” form. Most parameters from the data are recorded in a cumulative fashion. However, current is one major exception here, and instantaneous current is recorded rather than cumulative current. Therefore, I_{avg} does not need to be changed to delta form. Delta form means that instead of having cumulative parameters, the quantities are now being kept track of as the difference from one sample to the next.

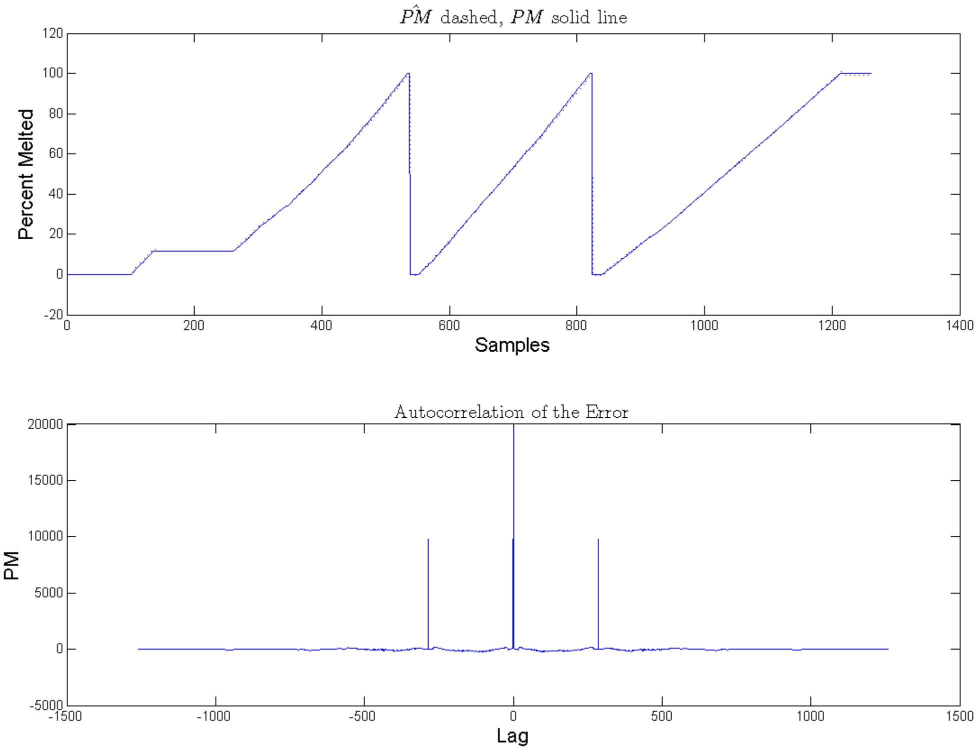


Figure 2.28: Actual PM , its estimate and autocorrelation of model (20)

(24) $$K\hat{W}H(n) = K\hat{W}H(n-1) + \theta_{1KWH} I_{avg}(n-1)$$

(25) $$I^2\hat{h}(n) = I^2\hat{h}(n-1) + \theta_1 I_{avg}(n-1)$$

(26) $$\begin{aligned} &P\hat{M}(n) = P\hat{M}(n-1) + \theta_{1PM}\Delta KWH(n-1) + \theta_2\Delta O_2(n-1) \\ &+ \theta_3\Delta Gas(n-1) + \theta_4\Delta Carbon(n-1) + \theta_5 I_{avg}(n-1) \end{aligned}$$

The final change to the model is a bit more complicated. It was observed that the first coefficient of the percent melted model, θ_1, was regularly negative. This last change is a method that changes the coefficient's sign. First, model (24) can be manipulated in a way as shown in (27)

(27) $$\Delta KWH(n) = K\hat{W}H(n) - K\hat{W}H(n-1) = \theta_1 I_{avg}(n-1)$$

Assuming that being off by one step will not significantly alter results, $\Delta KWH(n)$ can now be replaced with $\theta_1 I_{avg}(n-1)$ in the PM model. So now,

(28) $$\begin{aligned} &P\hat{M}(n) = P\hat{M}(n-1) + \theta_{1PM}\Delta KWH(n-1) + \theta_2\Delta O_2(n-1) \\ &+ \theta_3\Delta Gas(n-1) + \theta_4\Delta Carbon(n-1) \\ &= P\hat{M}(n-1) + \theta_{1PM}\theta_{1KWH} I_{avg}(n-1) + \theta_2\Delta O_2(n-1) \\ &+ \theta_3\Delta Gas(n-1) + \theta_4\Delta Carbon(n-1) \end{aligned}$$

The new coefficient of (29) was calculated using equations (24) and (26). Typically this resulted in a positive coefficient.

(29) $$\theta_{1pos} = \theta_{1PM}\theta_{1KWH}$$

Then the resulting new coefficient is inserted into the percent melted equation. Now all state models have achieved their final forms. To avoid confusion with coefficients, θ_1 for KWH and I^2H have been reassigned as φ_1 and ϕ_1, respectively.

(30) $$K\hat{W}H(n) = K\hat{W}H(n-1) + \varphi_1 I_{avg}(n-1)$$

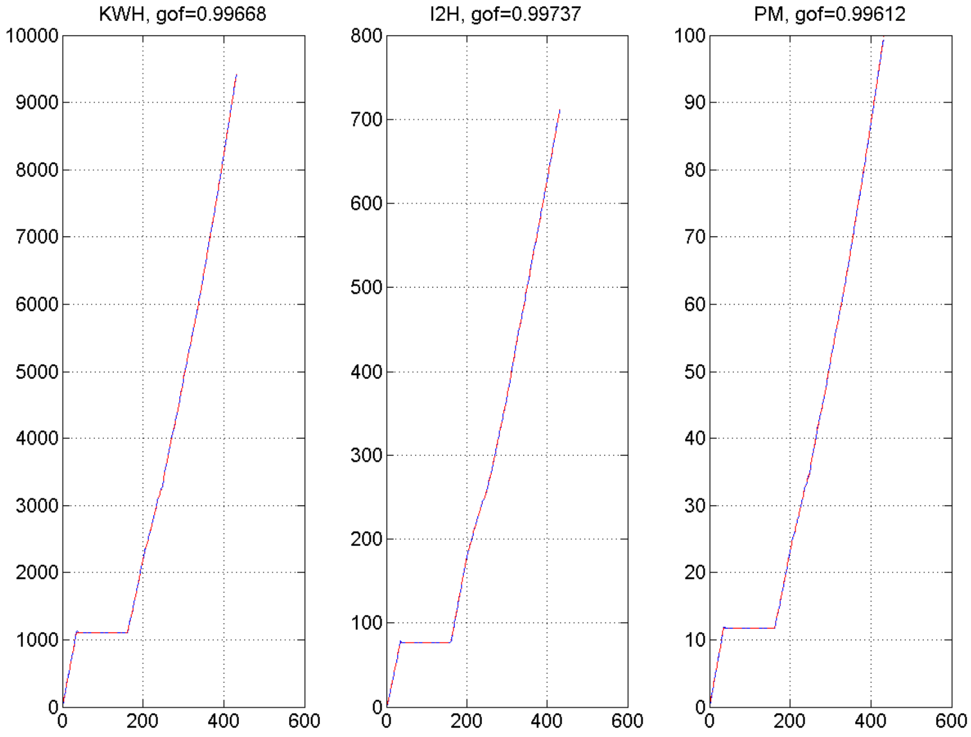


Figure 2.29: Charge 1 models and GoFs

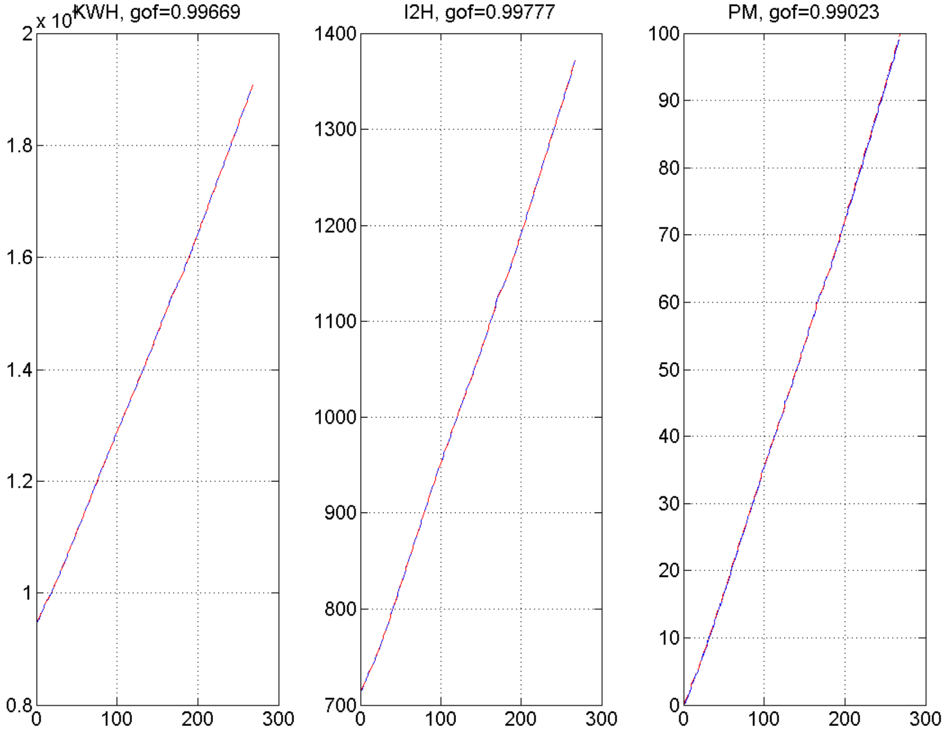


Figure 2.30: Charge 2 models and GoFs

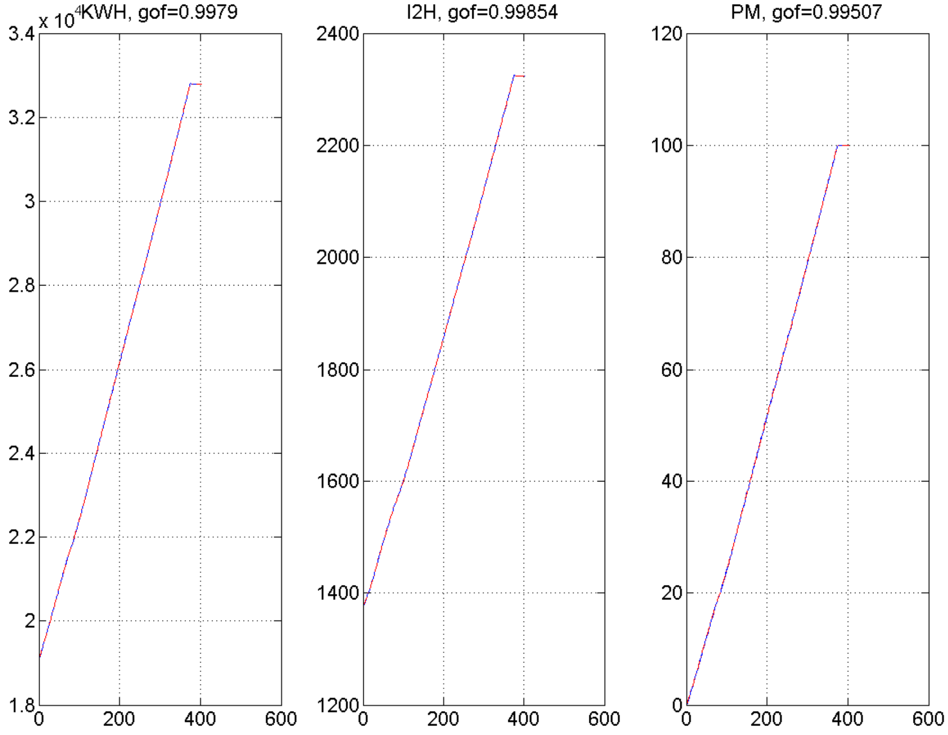


Figure 2.31: Charge 3 models and GoFs

(31) $$I^2\hat{h}(n) = I^2\hat{h}(n-1) + \phi_1 I_{avg}(n-1)$$

(32) $$\begin{aligned}&P\hat{M}(n) = P\hat{M}(n-1) + \theta_{1pos} I_{avg}(n-1) + \theta_2 \Delta O_2(n-1)\\ &+ \theta_3 \Delta Gas(n-1) + \theta_4 \Delta Carbon(n-1)\end{aligned}$$

As the fitting is now calculated for each individual charge, the GoF of each model was calculated for every charge in heat 29965. The actual data, estimate and GoF of models (30), (31), and (32) are shown Figures 2.29-2.31. As shown, the models perform very well. The additional changes to reduce negative coefficients, reflect free oxygen, and prepare for optimized control did not adversely effect accuracy in any of the models, and in the percent melted case, overall accuracy was greatly increased from a GoF of 0.87921 to 0.99404. There is no visible difference from the model and actual on the graphs, and each GoF is at least 0.99. This is typical for any charge. Additionally, out of a sample of 254 heats, only 16 were found to have any negative coefficients with these new models.

2.5 Model Validation

In order to properly show that the new modeling scheme would work in the real world, the model should be validated by comparing one model's parameters and fit it against a similar but separate heat. First, an algorithm was devised that determined similar heats in terms of charge weights. Each charge is given an identifier (1=low, 2=medium, 3=high) to reflect the amount of scrap put in the furnace in that particular charge. Charges were identified according to Figure 2.32. It is important to realize that a moderate amount of weight was typical. Once each charge has an identifier, they were reassembled into their heat and matched with similar heats. The models developed based on the heat file numbers 30217, 30012 and 30096 respectively were validated on 10 different files each of similar change weights. Figures 2.33-2.35 show three different comparisons between heats with similar charge weights. Figure 2.33 compares heat 30017 with heat 30205. Figure 2.34 compares heat 30041 and 30133. Figure 2.35 compares heat 30187 and 30288. A brief summary of results for 30 different heat comparisons is shown in Figure 2.36. As shown, the GoF never drops below 0.99 for any of the state models. This shows that the models developed on one file can be very well applied to predict on a similar file. This is essential because it is impossible to have a model made during the steel melting process; it must be known a priori. This indicates that the model will still perform well, even though it is calculated before hand based on a past heat file.

		Charge Weight (in tons)		
		1 (low)	2 (medium)	3(high)
Charge Number	1	CW<33	33<CW<35	CW>35
	2	CW<33	33<CW<35	CW>35
	3	CW<19	19<CW<22	CW>22

Figure 2.32: Charge identifying diagram

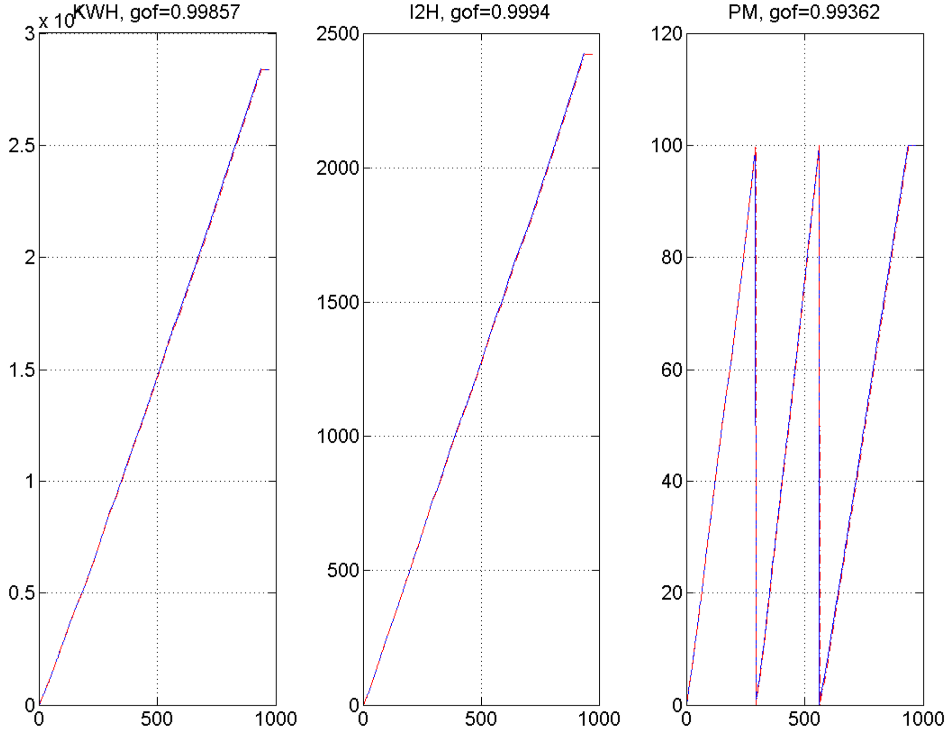


Figure 2.33: Model comparison of heat 30017 and 30205

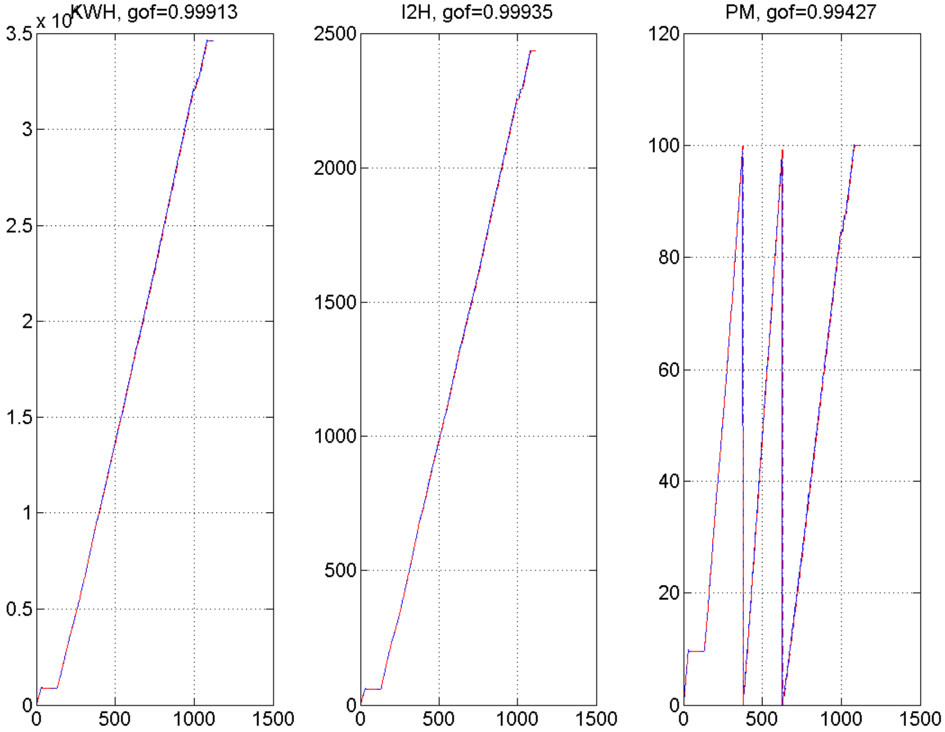


Figure 2.34: Model comparison of heat 30041 and 30133

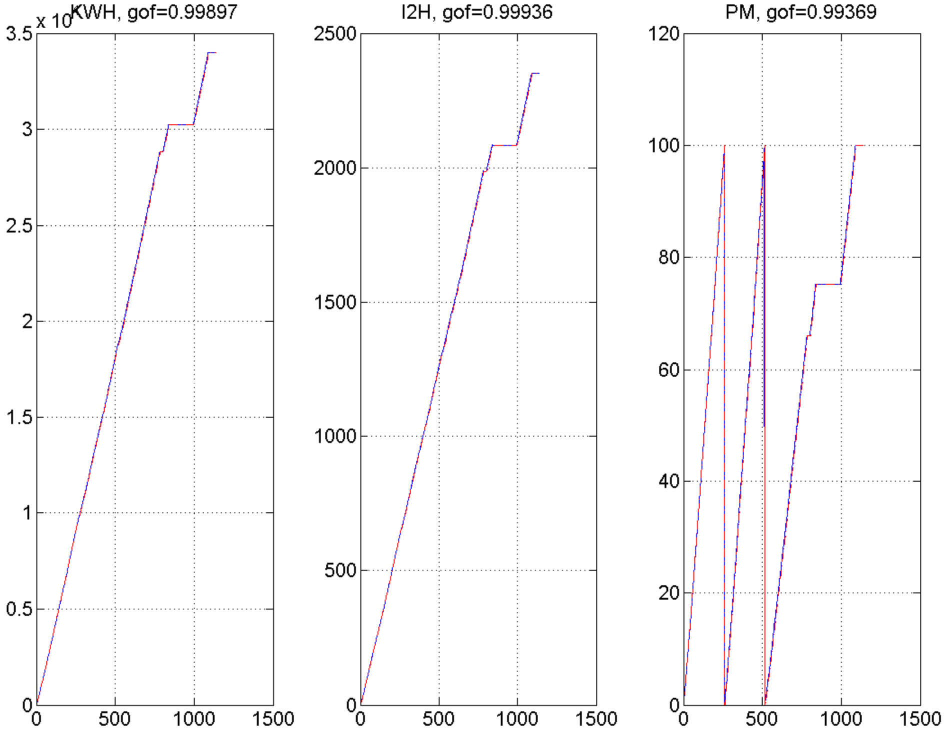
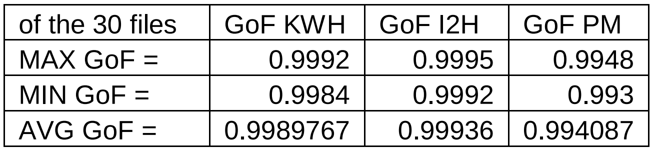


Figure 2.35: Model comparison of heat 30187 and 30288



of the 30 files	GoF KWH	GoF I2H	GoF PM
MAX GoF =	0.9992	0.9995	0.9948
MIN GoF =	0.9984	0.9992	0.993
AVG GoF =	0.9989767	0.99936	0.994087

Figure 2.36: Summary of model comparisons

CHAPTER 3

OPTIMAL CONTROL

This chapter describes the new optimal control system and its effects.

3.1 Final-State-Fixed LQ Regulator

Now that a simple, accurate, and physically viable model has been made, an optimal control system can be implemented to melt a heat of steel with the minimum cost of materials.

A variety of optimal control solutions could be used, but a final-state-fixed LQ regulator was selected. This particular solution was chosen because the overall most important factor in steel melting is forcing each charge to be %100 melted. The final state equations of (30), (31), and (32)

$$\text{(30)} \qquad K\hat{W}H(n) = K\hat{W}H(n-1) + \varphi_1 I_{avg}$$

$$\text{(31)} \qquad I^2\hat{h}(n) = I^2\hat{h}(n-1) + \phi_1 I_{avg}$$

$$\text{(32)} \qquad \begin{aligned} &P\hat{M}(n) = P\hat{M}(n-1) + \theta_{1pos} I_{avg}(n-1) + \theta_2 \Delta O_2(n-1) \\ &+ \theta_3 \Delta Gas(n-1) + \theta_4 \Delta Carbon(n-1) \end{aligned}$$

can be written in a matrix form as

$$\text{(33)} \qquad \begin{bmatrix} KWH(n+1) \\ I^2H(n+1) \\ PM(n+1) \end{bmatrix} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} KWH(n) \\ I^2H(n) \\ PM(n) \end{bmatrix} + \begin{bmatrix} \varphi_1\ 0\ 0\ 0 \\ \phi_1\ 0\ 0\ 0 \\ \theta_{1pos}\ \theta_2\ \theta_3\ \theta_4 \end{bmatrix} \begin{bmatrix} I_{avg}(n) \\ O_2(n) \\ Gas(n) \\ Carbon(n) \end{bmatrix},$$

or simply

$$\text{(34)} \qquad x(n+1) = Ax(n) + Bu(n) \qquad\qquad n \geq 0$$

with

$$x(n)=\begin{bmatrix} KWH(n) \\ I^2H(n) \\ PM(n) \end{bmatrix}, \quad A=\begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}, \quad B=\begin{bmatrix} \varphi_1\ 0\ 0\ 0 \\ \phi_1\ 0\ 0\ 0 \\ \theta_{1pos}\ \theta_2\ \theta_3\ \theta_4 \end{bmatrix}, \quad u(n)=\begin{bmatrix} I_{avg}(n) \\ O_2(n) \\ Gas(n) \\ Carbon(n) \end{bmatrix}$$

Here $x(n)$ stands for the 3 states and $u(n)$ stands for the 4 inputs. A is represented as an identity matrix because each previous step state variable has no coefficient. B represents the coefficients found from the 3 state models for their respective inputs. Now it is important to know that while (34) depicts how future values of the state equations are calculated, it is not necessarily what is optimized. Note that the goal here is to force PM to 100 while minimizing I^2H and KWH as well as the usage of oxygen, gas and carbon. Therefore, PM was used as the variable $x(n)$, and only its coefficients were used in the actual optimal control. For the cost function J, shown in equation (35), there were several options. The route chosen involved first

(35) $$J=\frac{1}{2}PM^T(N)S(N)PM(N)+\frac{1}{2}\sum_{k=0}^{N-1}\left[PM^T(n)QPM(n)+u^T(n)Ru(n)\right],$$

$$S(N)=0, \quad Q=0, \quad R>0$$

Or

$$J=\frac{1}{2}\sum_{k=0}^{N-1}\left[u^T(n)Ru(n)\right]$$

N always refers to the final charge sample number. By setting these weights to 0, this has the effect of putting the entire penalty on the inputs u and their weights R. This is a bit counterintuitive as the 2 largest costs of the furnace are the state equations I^2H and KWH. However, the penalty function can be used this way because both I^2H and KWH only depend on the inputs that already affect PM. In this way, I^2H and KWH are indirectly penalized by their input, I_{avg} , and its corresponding penalty R . Next, the fixed-final-state was fixed at 100 by setting $Cx(N) = r(N) = 100$ and $C = 1$ in equation (36).

(36) $Cx(N) = r(N)$

This was done so that as long as there are no additional constraints, each charge would always end once %100 melted.

The R matrices found in the optimal control solutions were found for each individual charge. The individual values were chosen because they result in both a decrease in cost, and a reasonable model. The model is reasonable if it is within rough physical constraints [1]. These included

- Current should be constrained to between 30000-50000 A
- Oxygen should be constrained to less than 1000 SCF per charge ton
- Gas should be constrained to a max of 300 SCF per charge ton
- Carbon should have a max of 2000 lbs per heat, meaning roughly less than 23 lbs per charge ton

Without these constraints, the optimal control can be solved using equations (37)-(42) from Frank L. Lewis's <u>Optimal Control</u> on page 133 [4].

(37) $K(n) = \left[B^T S(n+1)B + R\right]^{-1} B^T S(n+1)A$

(38) $S(n) = A^T S(n+1)[A - BK(n)] + Q$

(39) $V(n) = [A - BK(n)]^T V(n+1), \qquad V(N) = C^T$

(40) $P(n) = P(n+1) - V^T(n+1)B\left[B^T S(n+1)B + R\right]^{-1} B^T V(n+1), \ P(N) = 0$

(41) $K^u(n) = \left[B^T S(n+1)B + R\right]^{-1} B^T$

(42) $$u(n) = -\left[K(n) - K^{u}(n)V(n+1)P^{-1}(n)V^{T}(n)\right]x(n) - K^{u}(n)V(n+1)P^{-1}(n)r(N)$$

It is important to note that these constraints impose nonlinear conditions on the control so that the outcome may or may not be optimal. By carefully adjusting weighting, all the constraints except the first are actually automatically satisfied. The first one remains in effect. It is obvious that *KWH* is the most expensive parameter per percentage of steel melted. Therefore, the R weighting was selected so that the average current is as low as possible to save as much as possible. To find the values of R, the model was first attempted entirely unconstrained, and the cost parameters were adjusted by altering the values of R. Since the minimum current constraint is 30000 A, the R weights were first chosen so that the average current would typically end up right around 30000 A. Next, the other R values were adjusted such that the other cost parameters were reasonable. Then, the actual hard constraints were added to be certain the model stayed reasonable. As a hard constraint on the current to 30000 A could result in over-melting, a safe-guard was created that shut off the current if *PM* ever reached 100. R matrices are listed as follows:

$$R_{chrg1} = \begin{bmatrix} 0.0003 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 8500 \end{bmatrix}$$

$$R_{chrg2} = \begin{bmatrix} 1.2\text{e-}6 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 5000 \end{bmatrix}$$

$$R_{chrg3} = \begin{bmatrix} 0.4\mathrm{e}-7 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 2500 \end{bmatrix}$$

The cost J was found by using actual cost coefficients supplied by Wilton. These coefficients will not be revealed due to the competitiveness of the industry. These were simply multiplied by the final values of their respective cost parameters and summed.

(43) $J = \alpha_1 KWH(N) + \alpha_2 I^2 h(N) + \alpha_3 O_2(N) + \alpha_4 Gas(N) + \alpha_5 Carbon(N)$

Using this equation, the final costs of the actual data of previous heats as well as the new optimal cost models can be calculated and compared.

CHAPTER 4

RESULTS

The relevant results of this project are relatively simple. They can be separated into 2 categories, modeling and optimal control.

4.1 Modeling Results

The goals of the modeling portion of this thesis were to construct a model that has both improved accuracy and simplicity than previous attempts. The model also needs to make physical and intuitive sense.

First, the simplicity of the models was greatly reduced. The initial models (1),(9),(13) contained 16 coefficients that needed to be calculated. The final models (30),(31),(32) contained 7 coefficients that needed to be calculated.

Next, the model makes physical and intuitive sense. While the original models had many negative coefficients in each model, the final models typically contain none. Negative coefficients were found in modeling processes in just 16 out of the 254 heats studied for a correctness of %93.7.

Finally, the model has much improved accuracy. Figure 4.1 shows the actual data, estimate and the GoF of the initial models (1),(9),(13). Figure 4.2 shows the actual data, estimate and the GoF of the final models (30),(31),(32) for the same heat. As shown, the GoF increases for all three models. The GoF has a particularly large margin of improvement for the percent melted model. This is also true when the coefficients derived from one heat are used to model a similar heat. The GoF is still much better than previous models, and tends to average above 0.99. Results of the GoF where 10 similar

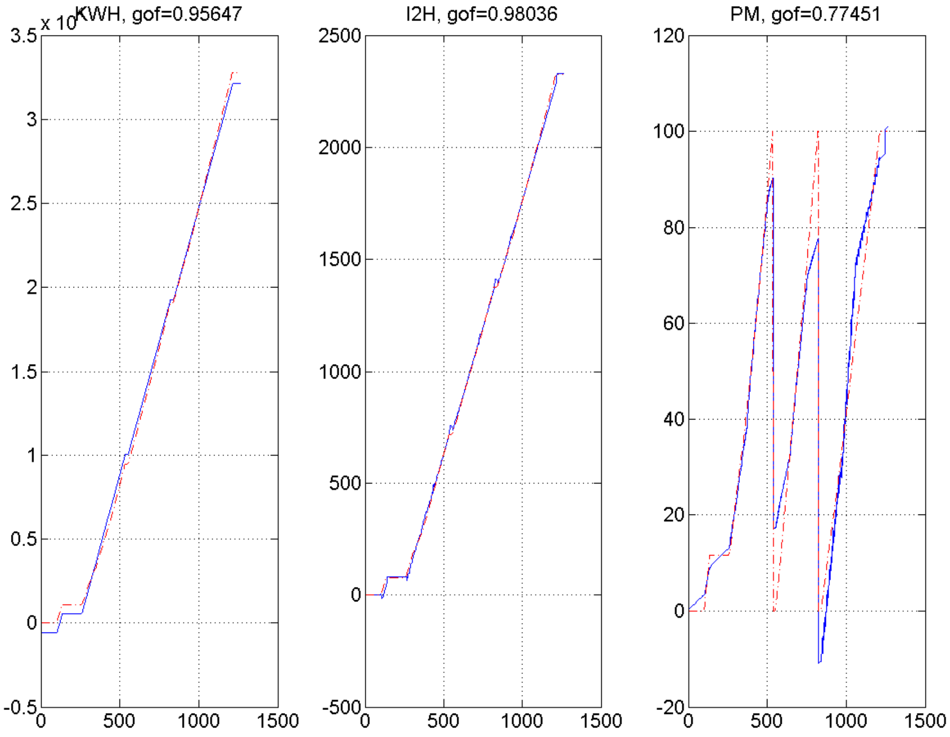


Figure 4.1: Actual data, its estimate and GoF of models (1),(9),(13)

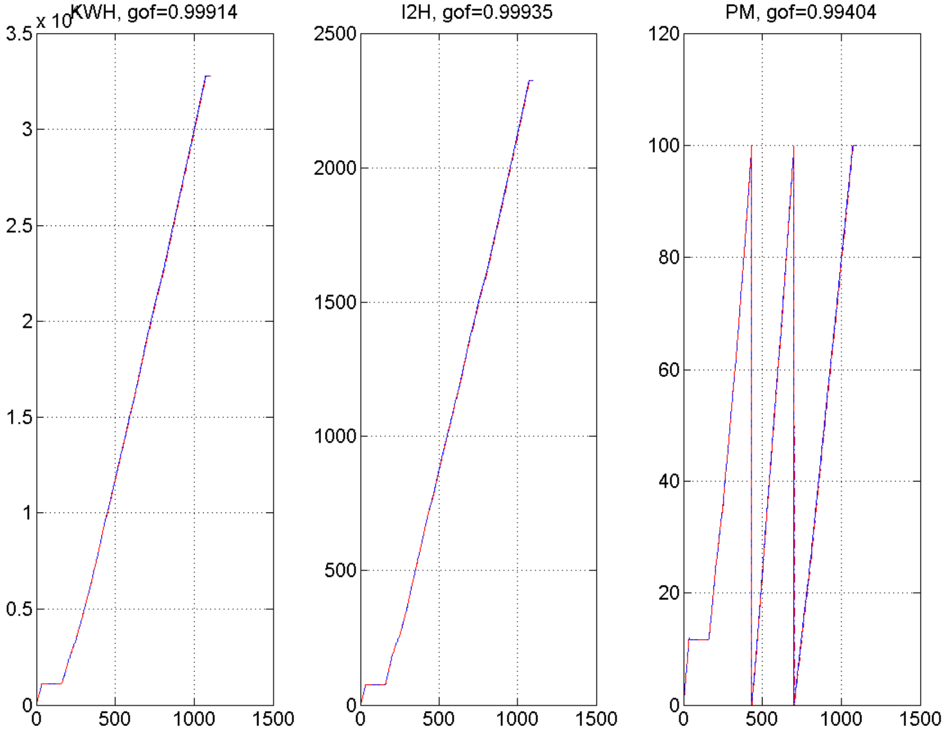


Figure 4.2: Actual data, its estimate and GoF of models (30),(31),(32)

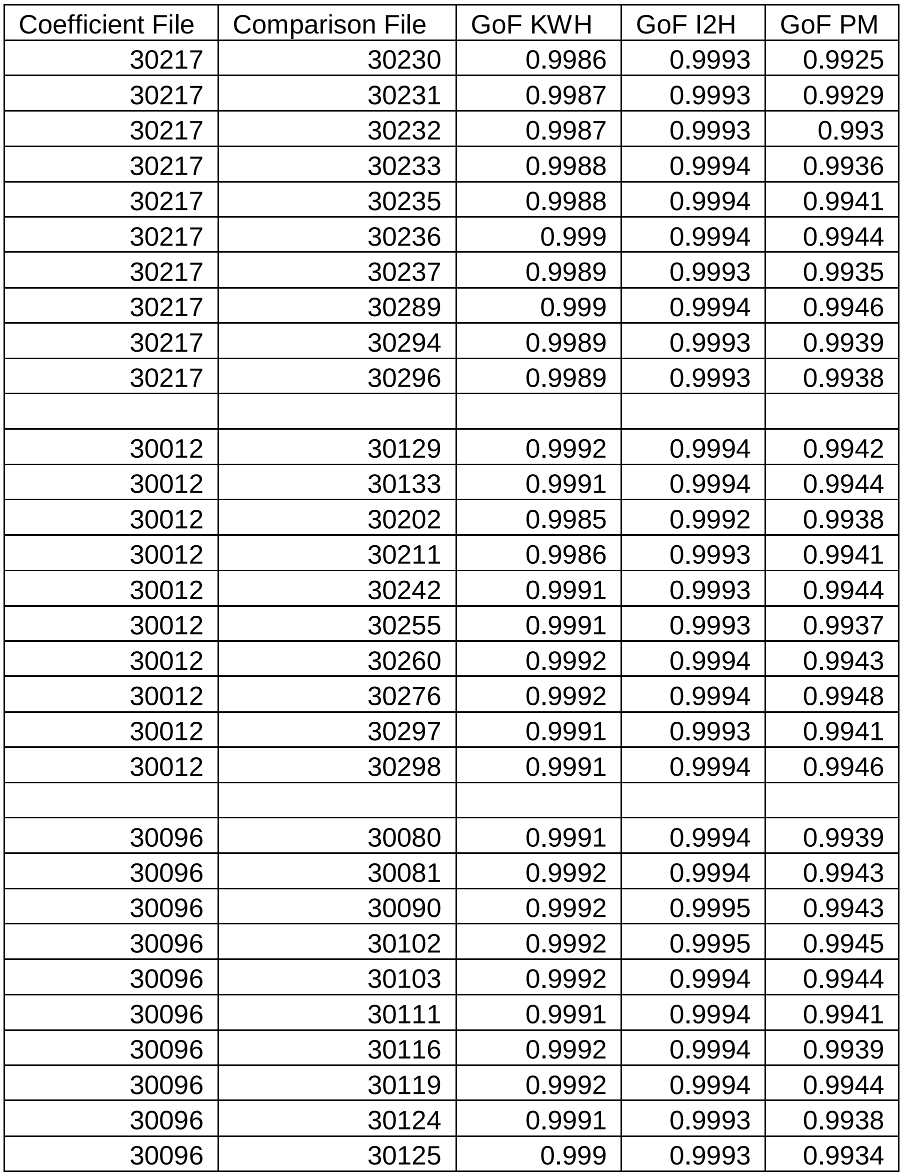
models were compared for 3 different groups of similar files are shown in Figure 4.3. General statistics of the comparisons are shown in Figure 4.4. As shown, the GoF when compared to a similar file is still always above 0.99 for all models. This proves that the model is vastly more accurate.

4.2 Optimal Control Results

There are 2 main goals that need to be achieved. First, the optimal control model must behave in a reasonable fashion. As stated earlier, this can be achieved by having its parameters observe certain constraints. Second and most importantly, the optimal control model should decrease overall cost.

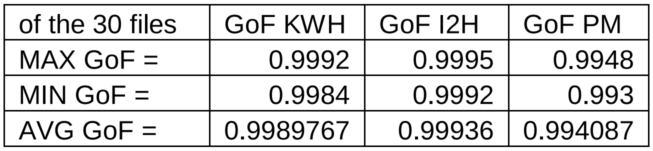
The first goal was achieved mainly by different weightings of R. As long as the charge currents were near 30000 A, the rest of the input parameters stayed reasonable. Note that after the weights of R were established, a hard constraint was placed on current so that it never went below 30000 A. If this manual constraint would cause the percent melted to exceed %100, then the current would shut off. Using these techniques, all 238 correct files tested were found to be reasonable for a %100 success rate.

The second goal was shown by two ways. First, the optimal solution is shown versus the original data. Next, the model is calculated for one heat, and then the model is applied to a new heat with similar characteristics and the optimal solution is calculated. This second comparison is better, as this is more similar to reality. In the real world, the model will have to be on hand before starting the heat that needs to be optimized. At any rate, both results are shown. The optimal profile versus the original data was tracked for a total of 300 heats. Note that this includes outliers. A summary of the optimal profile versus the original data is included in Figure 4.5. All ratios refer to the cost ratio of



Coefficient File	Comparison File	GoF KWH	GoF I2H	GoF PM
30217	30230	0.9986	0.9993	0.9925
30217	30231	0.9987	0.9993	0.9929
30217	30232	0.9987	0.9993	0.993
30217	30233	0.9988	0.9994	0.9936
30217	30235	0.9988	0.9994	0.9941
30217	30236	0.999	0.9994	0.9944
30217	30237	0.9989	0.9993	0.9935
30217	30289	0.999	0.9994	0.9946
30217	30294	0.9989	0.9993	0.9939
30217	30296	0.9989	0.9993	0.9938
30012	30129	0.9992	0.9994	0.9942
30012	30133	0.9991	0.9994	0.9944
30012	30202	0.9985	0.9992	0.9938
30012	30211	0.9986	0.9993	0.9941
30012	30242	0.9991	0.9993	0.9944
30012	30255	0.9991	0.9993	0.9937
30012	30260	0.9992	0.9994	0.9943
30012	30276	0.9992	0.9994	0.9948
30012	30297	0.9991	0.9993	0.9941
30012	30298	0.9991	0.9994	0.9946
30096	30080	0.9991	0.9994	0.9939
30096	30081	0.9992	0.9994	0.9943
30096	30090	0.9992	0.9995	0.9943
30096	30102	0.9992	0.9995	0.9945
30096	30103	0.9992	0.9994	0.9944
30096	30111	0.9991	0.9994	0.9941
30096	30116	0.9992	0.9994	0.9939
30096	30119	0.9992	0.9994	0.9944
30096	30124	0.9991	0.9993	0.9938
30096	30125	0.999	0.9993	0.9934

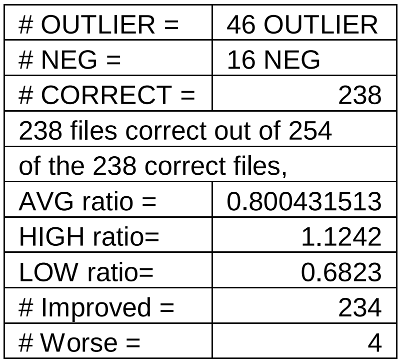
Figure 4.3: GoF’s for model comparisons



of the 30 files	GoF KWH	GoF I2H	GoF PM
MAX GoF =	0.9992	0.9995	0.9948
MIN GoF =	0.9984	0.9992	0.993
AVG GoF =	0.9989767	0.99936	0.994087

Figure 4.4: GoF modeling summary of results

$\frac{J_{new}}{J_{old}}$. 3 typical heats and their overall characteristics are shown in Figures 4.6-4.8. The real data is a solid line and its optimal profile is shown as a dotted line. The respective heats are 30025, 30144, and 30265. Next, the model from a previous heat is tested on a new heat that is similar. Each charge is given an identifier ((1)low, (2)medium, (3)high) to reflect the amount of scrap put in the furnace in that particular charge, similar to what was described in section 2.5 and Figure 2.32. Once each charge has an identifying number, they are put together into their heat and matched with similar heats. Results of 3 of these similar comparisons are shown. Figure 4.9 lists comparisons of heat 30017 to its 36 similar heats of the 300 heats tested. These heats fall into the 2,1,3 identifiers for their respective charges. This means that the first charges have medium weight, the second charges have low weights, and the third charges have high weights. A summary of these results is shown in Figure 4.10. Figure 4.11 lists comparisons of heat 30149 to its 28 similar heats of the 300 heats tested. These heats fall into the 2,2,2 identifiers for their respective charges. A summary of these results is shown in Figure 4.12. Figure 4.13 lists comparisons of heat 30222 to its 39 similar heats of the 300 heats tested. These heats fall into the 2,2,3 identifiers for their respective charges. A summary of these results is shown in Figure 4.14. All ratios again refer to the cost ratio of $\frac{J_{new}}{J_{old}}$. As shown, the optimized control profile still performs very well. In some of the cases, the average cost savings is greater than the optimal profile vs. actual average savings. This is because heats with certain charge identifiers seem to outperform others. Another benefit of using a previously calculated model is that now the files that had negative coefficients associated with them can now be used, as the coefficients are already calculated. Overall,



# OUTLIER =	46 OUTLIER
# NEG =	16 NEG
# CORRECT =	238
238 files correct out of 254	
of the 238 correct files,	
AVG ratio =	0.800431513
HIGH ratio=	1.1242
LOW ratio=	0.6823
# Improved =	234
# Worse =	4

Figure 4.5: Summary of results of optimal profiles vs. original data costs

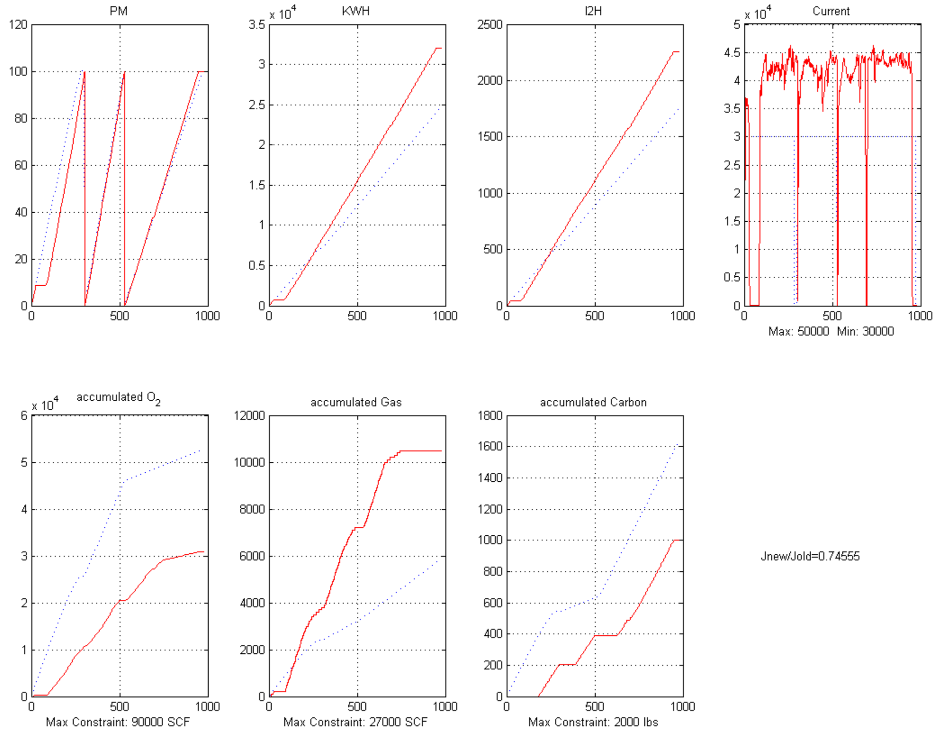


Figure 4.6: Actual data and optimal model

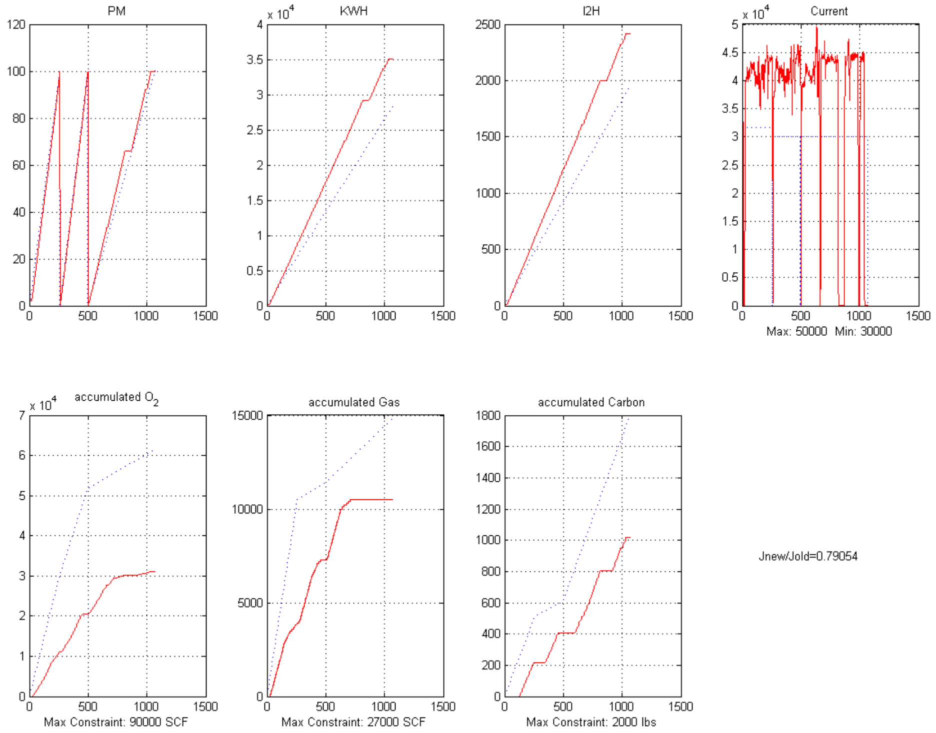


Figure 4.7: Actual data and optimal model

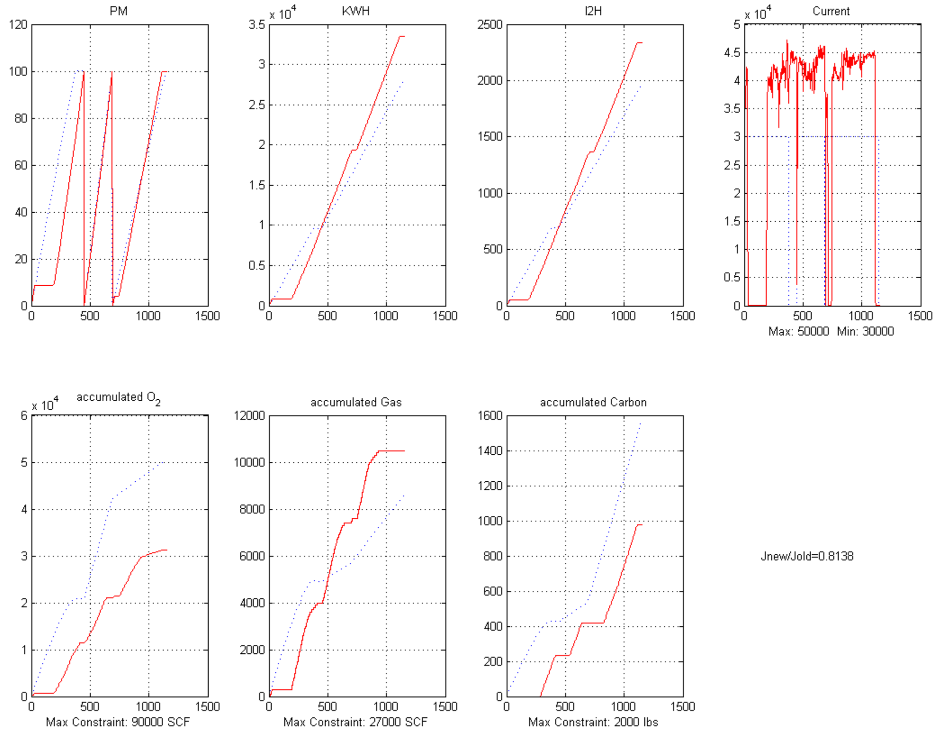
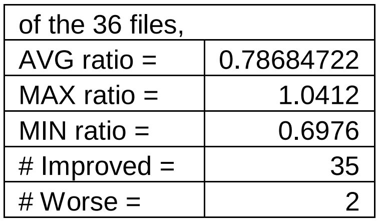


Figure 4.8: Actual data and optimal model

Coefficient File	Comparison File	Error Ratio
30017	30029	0.8008
30017	30055	0.723
30017	30056	0.7388
30017	30060	0.8337
30017	30062	0.754
30017	30066	0.7975
30017	30085	0.8239
30017	30091	0.727
30017	30092	0.7232
30017	30094	0.7665
30017	30095	0.7097
30017	30107	0.7143
30017	30138	0.7719
30017	30152	0.7646
30017	30165	0.6995
30017	30167	0.6976
30017	30168	0.8546
30017	30189	0.8958
30017	30197	0.7526
30017	30205	0.7193
30017	30209	0.7452
30017	30210	1.0412
30017	30217	1.0095
30017	30221	0.7683
30017	30225	0.7702
30017	30226	0.7917
30017	30230	0.7704
30017	30231	0.7847
30017	30232	0.758
30017	30233	0.7585
30017	30235	0.7981
30017	30236	0.9393
30017	30237	0.7302
30017	30289	0.8596
30017	30294	0.7433
30017	30296	0.79

Figure 4.9: Heat error ratios using 30017 and its 36 similar files

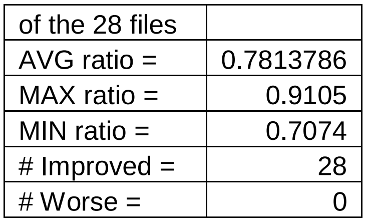


of the 36 files,	
AVG ratio =	0.78684722
MAX ratio =	1.0412
MIN ratio =	0.6976
# Improved =	35
# Worse =	2

Figure 4.10: Heat error ratio summary using 30017 and its 36 similar files

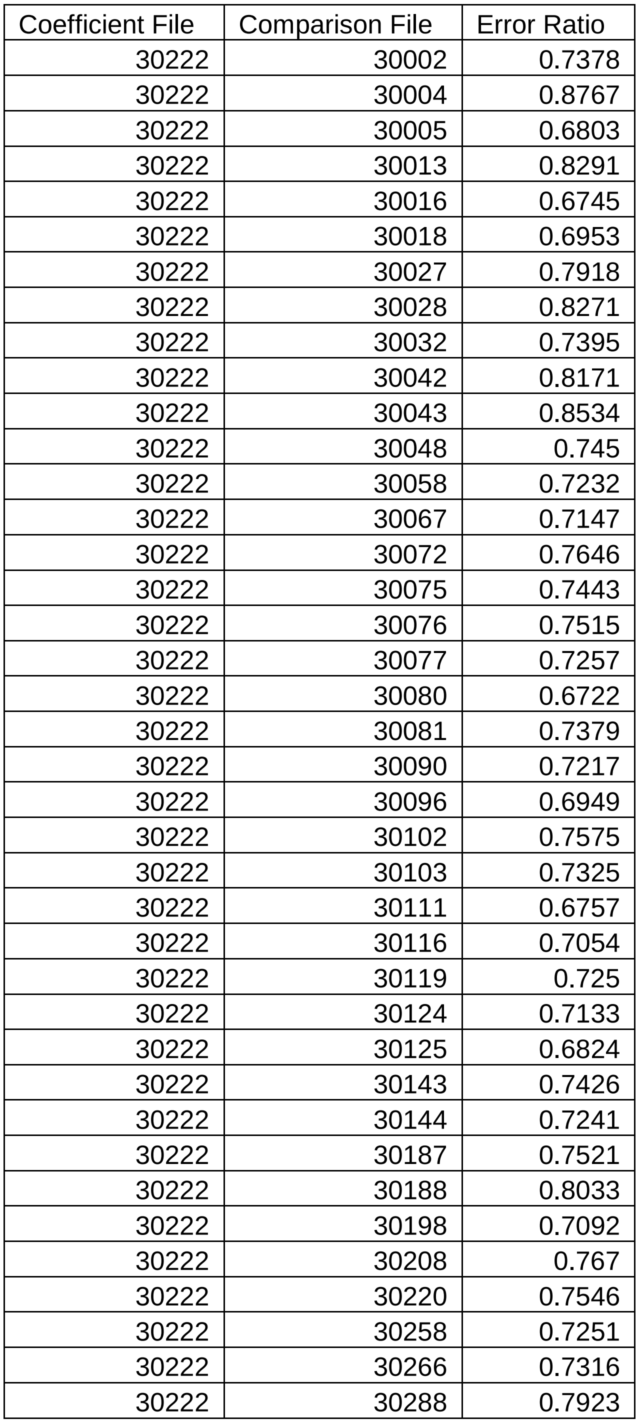
Coefficient File	Comparison File	Error Ratio
30149	30001	0.8171
30149	30008	0.7627
30149	30011	0.9105
30149	30012	0.7934
30149	30019	0.8223
30149	30025	0.735
30149	30031	0.722
30149	30034	0.7572
30149	30038	0.7952
30149	30039	0.8044
30149	30041	0.7535
30149	30045	0.7259
30149	30079	0.8182
30149	30082	0.7546
30149	30097	0.735
30149	30108	0.7714
30149	30113	0.7074
30149	30120	0.8202
30149	30129	0.7838
30149	30133	0.7533
30149	30202	0.7885
30149	30211	0.7797
30149	30242	0.8134
30149	30255	0.7528
30149	30260	0.766
30149	30276	0.8292
30149	30297	0.7776
30149	30298	0.8283

Figure 4.11: Heat error ratios using 30149 and its 28 similar files



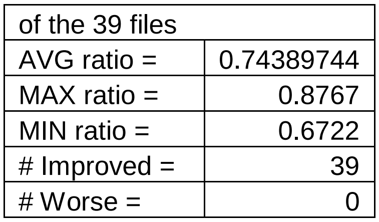
of the 28 files	
AVG ratio =	0.7813786
MAX ratio =	0.9105
MIN ratio =	0.7074
# Improved =	28
# Worse =	0

Figure 4.12: Heat error ratio summary using 30149 and its 28 similar files



Coefficient File	Comparison File	Error Ratio
30222	30002	0.7378
30222	30004	0.8767
30222	30005	0.6803
30222	30013	0.8291
30222	30016	0.6745
30222	30018	0.6953
30222	30027	0.7918
30222	30028	0.8271
30222	30032	0.7395
30222	30042	0.8171
30222	30043	0.8534
30222	30048	0.745
30222	30058	0.7232
30222	30067	0.7147
30222	30072	0.7646
30222	30075	0.7443
30222	30076	0.7515
30222	30077	0.7257
30222	30080	0.6722
30222	30081	0.7379
30222	30090	0.7217
30222	30096	0.6949
30222	30102	0.7575
30222	30103	0.7325
30222	30111	0.6757
30222	30116	0.7054
30222	30119	0.725
30222	30124	0.7133
30222	30125	0.6824
30222	30143	0.7426
30222	30144	0.7241
30222	30187	0.7521
30222	30188	0.8033
30222	30198	0.7092
30222	30208	0.767
30222	30220	0.7546
30222	30258	0.7251
30222	30266	0.7316
30222	30288	0.7923

Figure 4.13: Heat error ratios using 30222 and its 39 similar files



of the 39 files	
AVG ratio =	0.74389744
MAX ratio =	0.8767
MIN ratio =	0.6722
# Improved =	39
# Worse =	0

Figure 4.14: Heat error ratio summary using 30222 and its 39 similar files

all heats calculated had improved costs. The three particular similar heat categories above had average improvements of roughly 21, 22, and 26 percent. Figures 4.15-4.17 show overall results of the process for three different heat comparisons. The real data is a solid line and its optimal profile is shown as a dotted line.

4.3 Conclusions and Future Work

All the goals of this thesis have been met. The original model has been derived. A new model that makes sense has been found that is both simpler and far more accurate. A different optimized control profile has been devised in a reasonable way. This resulted in average cost savings of around %20. While this almost surely would decrease in the physical world, it still holds promise that it can improve cost.

Given time, a few things may be improved upon. The issues with free oxygen suggest that there is not enough oxygen being injected into the furnace normally. Perhaps the free oxygen equation could be better modeled, perhaps by a chemical process. Different metal consistencies in the scrap, grade of steel desired, and different scrap types could be included in the data and make for a more accurate model.

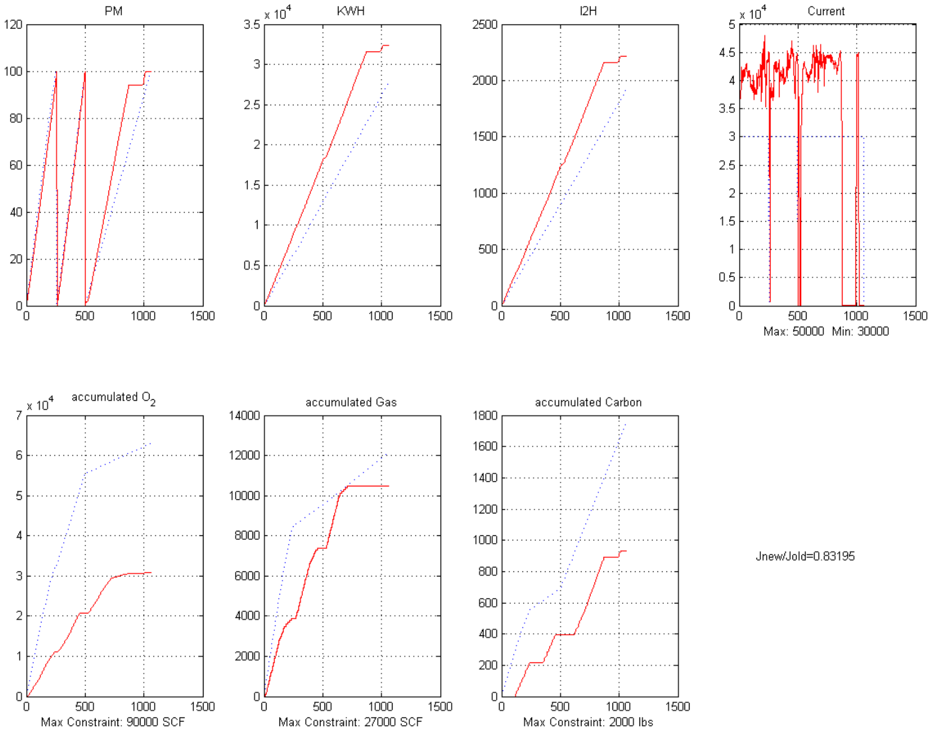


Figure 4.15: Actual data 30244 and optimal using 30124's coefficients

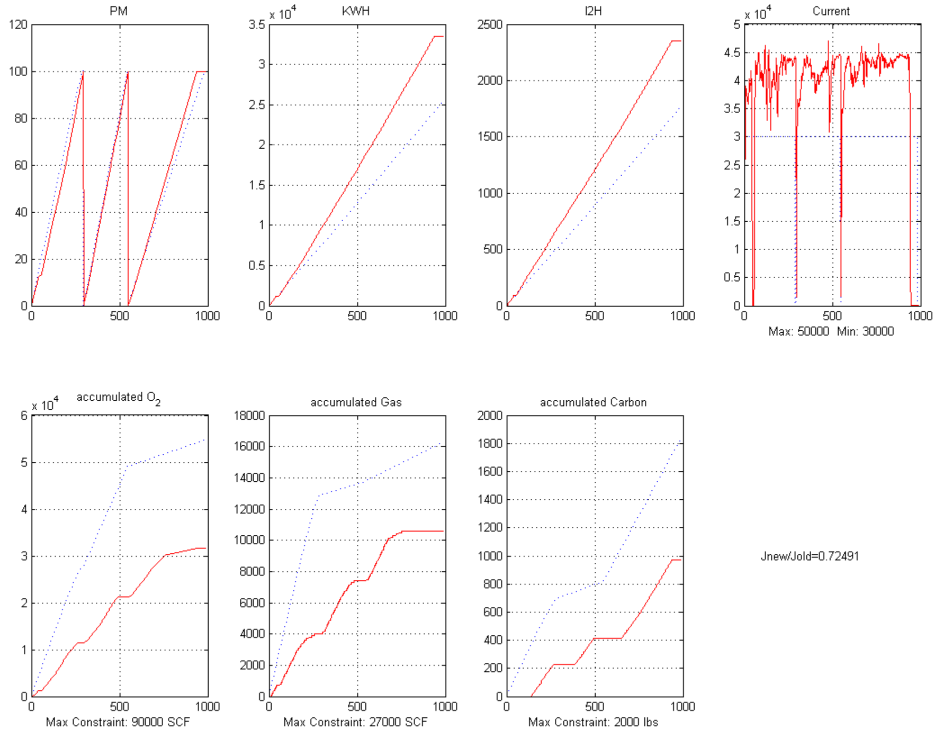


Figure 4.16: Actual data 30128 and optimal using 30082's coefficients

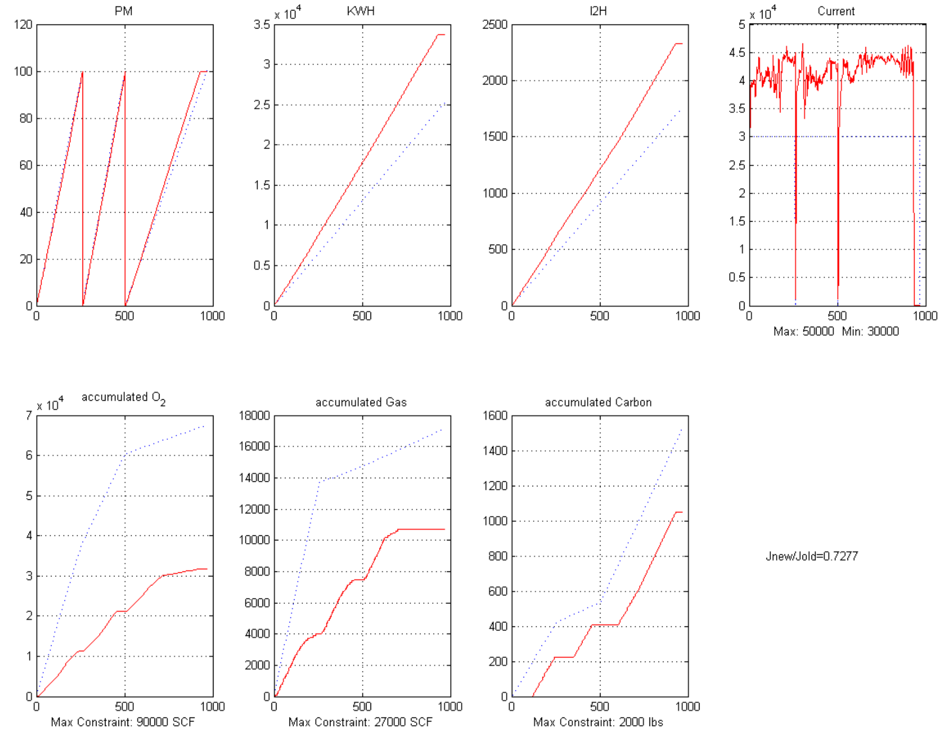


Figure 4.17: Actual data 30005 and optimal using 30050’s coefficients

APPENDIX A: RELEVANT RESULTS

A.1: Heats and Error Ratios of the optimal solution compared to the original

File Name	Error Ratio	Number Correct
30000	0.8514	1
30001	0.8336	1
30002	0.7989	1
30003	0.8666	1
30004	0.93	1
30005	0.7896	1
30006	0.7841	1
30007	0.8647	1
30008	NEG	
30009	0.8356	1
30010	0.9091	1
30011	NEG	
30012	0.7896	1
30013	0.9073	1
30014	OUTLIER	
30015	OUTLIER	
30016	0.7662	1
30017	0.7233	1
30018	0.739	1
30019	0.8889	1
30020	1.1242	1
30021	0.8743	1
30022	0.8416	1
30023	0.7633	1
30024	0.9628	1
30025	0.7456	1
30026	0.7495	1
30027	0.8277	1
30028	0.8864	1
30029	NEG	
30030	OUTLIER	
30031	NEG	
30032	0.8395	1
30033	0.9072	1
30034	0.7667	1
30035	0.9049	1
30036	0.818	1
30037	0.8934	1
30038	0.7992	1
30039	0.8217	1
30040	0.8529	1

30041	0.7593	1
30042	0.9015	1
30043	0.9709	1
30044	0.8991	1
30045	0.7335	1
30046	0.7593	1
30047	0.8998	1
30048	0.8364	1
30049	0.7194	1
30050	0.7254	1
30051	OUTLIER	
30052	OUTLIER	
30053	0.7713	1
30054	OUTLIER	
30055	0.7271	1
30056	0.7549	1
30057	0.6986	1
30058	0.7613	1
30059	OUTLIER	
30060	0.8331	1
30061	0.7553	1
30062	0.7497	1
30063	0.8048	1
30064	0.7545	1
30065	0.82	1
30066	0.8224	1
30067	0.7888	1
30068	0.7396	1
30069	0.8101	1
30070	0.7582	1
30071	0.8128	1
30072	0.8004	1
30073	NEG	
30074	1.021	1
30075	NEG	
30076	0.8066	1
30077	0.7767	1
30078	0.7283	1
30079	0.8055	1
30080	0.72	1
30081	0.8618	1
30082	0.768	1
30083	0.8344	1
30084	0.7503	1
30085	0.8104	1
30086	0.7893	1
30087	0.7738	1
30088	0.7591	1
30089	OUTLIER	

30090	0.7827	1
30091	0.724	1
30092	0.72	1
30093	0.7888	1
30094	0.7765	1
30095	0.7497	1
30096	0.7328	1
30097	0.7844	1
30098	0.7389	1
30099	0.8086	1
30100	0.7555	1
30101	OUTLIER	
30102	0.8023	1
30103	0.7568	1
30104	NEG	
30105	0.8081	1
30106	0.7546	1
30107	0.7086	1
30108	0.7797	1
30109	0.8345	1
30110	NEG	
30111	0.7727	1
30112	0.7784	1
30113	NEG	
30114	0.8301	1
30115	0.8929	1
30116	0.7418	1
30117	0.8374	1
30118	0.8374	1
30119	0.7741	1
30120	0.8336	1
30121	0.767	1
30122	0.8129	1
30123	0.7291	1
30124	0.762	1
30125	0.7603	1
30126	0.8142	1
30127	NEG	
30128	0.7327	1
30129	0.7942	1
30130	0.7766	1
30131	0.7858	1
30132	0.8784	1
30133	0.7741	1
30134	OUTLIER	
30135	OUTLIER	
30136	0.7509	1
30137	0.7661	1
30138	0.7835	1

30139	0.8634	1
30140	0.8115	1
30141	0.7387	1
30142	0.7436	1
30143	0.7887	1
30144	0.7905	1
30145	0.7415	1
30146	0.9726	1
30147	0.8395	1
30148	0.8149	1
30149	0.7228	1
30150	0.7634	1
30151	0.7593	1
30152	0.763	1
30153	0.7482	1
30154	0.7565	1
30155	OUTLIER	
30156	0.7354	1
30157	0.816	1
30158	OUTLIER	
30159	NEG	
30160	NEG	
30161	0.8833	1
30162	0.8892	1
30163	0.7952	1
30164	0.848	1
30165	0.7601	1
30166	0.7767	1
30167	0.7148	1
30168	0.9015	1
30169	0.7586	1
30170	0.8302	1
30171	0.8035	1
30172	0.827	1
30173	0.7949	1
30174	0.7989	1
30175	0.7972	1
30176	0.7813	1
30177	0.7509	1
30178	0.7843	1
30179	OUTLIER	
30180	OUTLIER	
30181	0.8001	1
30182	0.7376	1
30183	0.9245	1
30184	0.891	1
30185	0.8416	1
30186	0.7926	1
30187	0.7613	1

30188	0.7953	1
30189	0.8411	1
30190	0.7932	1
30191	OUTLIER	
30192	OUTLIER	
30193	0.7214	1
30194	OUTLIER	
30195	0.7877	1
30196	OUTLIER	
30197	0.715	1
30198	0.7	1
30199	OUTLIER	
30200	OUTLIER	
30201	0.7729	1
30202	0.7456	1
30203	OUTLIER	
30204	OUTLIER	
30205	0.688	1
30206	0.7464	1
30207	0.7989	1
30208	0.767	1
30209	0.7099	1
30210	0.9939	1
30211	0.7373	1
30212	OUTLIER	
30213	OUTLIER	
30214	OUTLIER	
30215	OUTLIER	
30216	OUTLIER	
30217	0.9623	1
30218	0.757	1
30219	0.7918	1
30220	0.7579	1
30221	0.7288	1
30222	0.7329	1
30223	0.7108	1
30224	OUTLIER	
30225	0.7338	1
30226	0.767	1
30227	OUTLIER	
30228	OUTLIER	
30229	OUTLIER	
30230	0.7297	1
30231	0.7416	1
30232	0.7222	1
30233	0.722	1
30234	0.8028	1
30235	0.772	1
30236	0.9405	1

30237	0.7006	1
30238	NEG	
30239	0.7485	1
30240	0.7397	1
30241	0.8282	1
30242	0.8629	1
30243	0.7988	1
30244	NEG	
30245	0.748	1
30246	NEG	
30247	0.7839	1
30248	0.8623	1
30249	0.7365	1
30250	0.7689	1
30251	0.9627	1
30252	OUTLIER	
30253	OUTLIER	
30254	OUTLIER	
30255	0.7633	1
30256	OUTLIER	
30257	0.7519	1
30258	0.7674	1
30259	0.7418	1
30260	0.777	1
30261	OUTLIER	
30262	0.7747	1
30263	0.7705	1
30264	0.7906	1
30265	0.8138	1
30266	0.7714	1
30267	0.732	1
30268	OUTLIER	
30269	OUTLIER	
30270	0.8459	1
30271	OUTLIER	
30272	OUTLIER	
30273	0.8183	1
30274	1.0723	1
30275	0.9167	1
30276	0.9097	1
30277	0.8846	1
30278	1.0338	1
30279	0.7907	1
30280	OUTLIER	
30281	OUTLIER	
30282	OUTLIER	
30283	OUTLIER	
30284	OUTLIER	
30285	0.6823	1

30286	0.7531	1
30287	0.7935	1
30288	0.8956	1
30289	0.8833	1
30290	0.8564	1
30291	0.7535	1
30292	0.9349	1
30293	0.7619	1
30294	0.7504	1
30295	0.8209	1
30296	0.8136	1
30297	0.7801	1
30298	0.8331	1
30299	NEG	

APPENDIX B: SOURCE CODE

B.1 Source Listing: Billyspmmodel.m

```
%this file is used to replicate, graph and show early
%thesis results
clear all

%initialize variables for for loop
totI1 = [];
totI2 = [];
totI3 = [];
totI2H = [];
totAccumulatedKWH = [];
totPowerOffTime = [];
totPowerOnTime = [];
totChargeKWHpt = [];
totHeatKWHpt = [];
totChargeO2pt = [];
totHeatO2pt = [];
totChargeGaspt = [];
totHeatGaspt = [];
totChargeCarbonpt = [];
totHeatCarbonpt = [];
totPercentMelted = [];



for i=5:5 %(just 150, say)
    if i<10
        x=['-2996',num2str(i),'h.csv'];
        temp = xlsread(x);
        if temp(:,66)>0
            %if there is a 4th charge, disregard the heat
        else %script
            %load in and set the core variables
            AccumulatedKWH = 1000*(temp(:,1));
            PowerOnTime = temp(:,3);
            PowerOffTime = temp(:,4);
            I1 = temp(:,5);
            I2 = temp(:,6);
            I3 = temp(:,7);
            I2H1 = temp(:,44);
            I2H2 = temp(:,45);
            I2H3 = temp(:,46);
            TotGas = temp(:,60)*100;
            RegO2 = temp(:,61)*100;
            TotCarbon = temp(:,62);
            ChWt1 = temp(:,63);
            ChWt2 = temp(:,64);
            ChWt3 = temp(:,65);
            HeatEnd = temp(:,69);
            PercentMelted = temp(:,91);



            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            ChargeLast = find(ChWt1,1,'last')   %the last cell # that
has charge weight
            ChargeLast2 = find(ChWt2,1,'last')

            if ChargeLast == ChargeLast2   %if there is no
discrepancies, operate

                %I2H is the average of the 3 phases
                I2H = I2H1;
                I2H(:) = (I2H1(:)+I2H2(:)+I2H3(:))/3;
                O2Last = find(RegO2,1,'last')           %O2,carbon, and
gas reset early, so find their last position
                %Get the cutoffs for the charges
                SecondChargeSrt = find(ChWt2,1)     %find where the
second charge starts
                ThirdChargeSrt = find(ChWt3,1)      %find where the
third charge starts
                %Find charge weight sizes
                ChWt1Size = ChWt1(ChargeLast-1)       %the total weight
of charge 1
                ChWt2Size = ChWt2(ChargeLast-1)       %the total weight
of charge 2
                ChWt3Size = ChWt3(ChargeLast-1)       %the total weight
of charge 3
                %find charge weights
                chwt1 = ChWt1(SecondChargeSrt-10)
                chwt2 = ChWt2(SecondChargeSrt+10)
                chwt3 = ChWt3(ThirdChargeSrt+10)
                i


                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


                %put all the parameters in their per ton form

                TotO2 = RegO2 - 0.5*2.3*TotGas - 1.5*13*TotCarbon;
%                 HeatKWHpt = AccumulatedKWH./(ChWt1+ChWt2+ChWt3);
%                 HeatO2pt = TotO2./(ChWt1+ChWt2+ChWt3);
%                 HeatO2pt = max(HeatO2pt,0);
%                 HeatGaspt = TotGas./(ChWt1+ChWt2+ChWt3);
%                 HeatCarbonpt = TotCarbon./(ChWt1+ChWt2+ChWt3);
```

```
                HeatKWHpt = AccumulatedKWH;
                HeatO2pt = TotO2;
                HeatO2pt = max(HeatO2pt,0);
                HeatGaspt = TotGas;
                HeatCarbonpt = TotCarbon;




                %put all the parameters in their per ton form
                Charge1KWH = AccumulatedKWH(1:SecondChargeSrt-1);
                Charge1KWHpt = Charge1KWH./ChWt1(1:SecondChargeSrt-1);
                Charge2KWH =
AccumulatedKWH(SecondChargeSrt:ThirdChargeSrt-1)-
AccumulatedKWH(SecondChargeSrt);
                Charge2KWHpt =
Charge2KWH./ChWt2(SecondChargeSrt:ThirdChargeSrt-1);
                Charge3KWH = AccumulatedKWH(ThirdChargeSrt:ChargeLast)-
AccumulatedKWH(ThirdChargeSrt);
                Charge3KWHpt =
Charge3KWH./ChWt3(ThirdChargeSrt:ChargeLast);

                ChargeKWHpt = [Charge1KWHpt;Charge2KWHpt;Charge3KWHpt];

                Charge1O2 = TotO2(1:SecondChargeSrt-1);
                Charge1O2pt = Charge1O2./ChWt1(1:SecondChargeSrt-1);
                Charge2O2 = TotO2(SecondChargeSrt:ThirdChargeSrt-1)-
TotO2(SecondChargeSrt);
                Charge2O2pt =
Charge2O2./ChWt2(SecondChargeSrt:ThirdChargeSrt-1);
                Charge3O2 = TotO2(ThirdChargeSrt:ChargeLast)-
TotO2(ThirdChargeSrt);
                Charge3O2pt =
Charge3O2./ChWt3(ThirdChargeSrt:ChargeLast);

                ChargeO2pt = [Charge1O2pt;Charge2O2pt;Charge3O2pt];

                Charge1Gas = TotGas(1:SecondChargeSrt-1);
                Charge1Gaspt = Charge1Gas./ChWt1(1:SecondChargeSrt-1);
                Charge2Gas = TotGas(SecondChargeSrt:ThirdChargeSrt-1)-
TotGas(SecondChargeSrt);
                Charge2Gaspt =
Charge2Gas./ChWt2(SecondChargeSrt:ThirdChargeSrt-1);
                Charge3Gas = TotGas(ThirdChargeSrt:ChargeLast)-
TotGas(ThirdChargeSrt);
                Charge3Gaspt =
Charge3Gas./ChWt3(ThirdChargeSrt:ChargeLast);

                ChargeGaspt = [Charge1Gaspt;Charge2Gaspt;Charge3Gaspt];

                Charge1Carbon = TotCarbon(1:SecondChargeSrt-1);
                Charge1Carbonpt =
Charge1Carbon./ChWt1(1:SecondChargeSrt-1);
```

```
                Charge2Carbon =
TotCarbon(SecondChargeSrt:ThirdChargeSrt-1)-TotCarbon(SecondChargeSrt);
                Charge2Carbonpt =
Charge2Carbon./ChWt2(SecondChargeSrt:ThirdChargeSrt-1);
                Charge3Carbon = TotCarbon(ThirdChargeSrt:ChargeLast)-
TotCarbon(ThirdChargeSrt);
                Charge3Carbonpt =
Charge3Carbon./ChWt3(ThirdChargeSrt:ChargeLast);

                ChargeCarbonpt =
[Charge1Carbonpt;Charge2Carbonpt;Charge3Carbonpt];



                %normalize percent melted

                PercentMelted1 = (PercentMelted(1:SecondChargeSrt-
1)/PercentMelted(SecondChargeSrt-1))*100;
                PercentMelted2 =
(PercentMelted(SecondChargeSrt:ThirdChargeSrt-
1)/PercentMelted(ThirdChargeSrt-1))*100;
                PercentMelted3 =
(PercentMelted(ThirdChargeSrt:ChargeLast)/PercentMelted(ChargeLast))*10
0;
                PercentMeltedNorm = [PercentMelted1; PercentMelted2;
PercentMelted3];


                %sum up all the parameters
                totI1 = [totI1; I1];
                totI2 = [totI2; I2];
                totI3 = [totI3; I3];
                totI2H = [totI2H; I2H];
                totAccumulatedKWH  = [totAccumulatedKWH; AccumulatedKWH];
                totPowerOffTime = [totPowerOffTime; PowerOffTime];
                totPowerOnTime = [totPowerOnTime; PowerOnTime];
                totHeatKWHpt = [totHeatKWHpt; HeatKWHpt];
                totHeatO2pt = [totHeatO2pt; HeatO2pt];
                totHeatGaspt = [totHeatGaspt; HeatGaspt];
                totHeatCarbonpt = [totHeatCarbonpt; HeatCarbonpt];
                totPercentMelted = [totPercentMelted;
PercentMeltedNorm];
                totChargeKWHpt = [totChargeKWHpt; ChargeKWHpt];
                totChargeO2pt = [totChargeO2pt; ChargeO2pt];
                totChargeGaspt = [totChargeGaspt; ChargeGaspt];
                totChargeCarbonpt  = [totChargeCarbonpt; ChargeCarbonpt];

            else
            end
        end
    elseif i>9 && i<100
        if i == 15 || i == 54 || i==92 || i==92 || i==93 || i==94 ||
i==95
            %these ones are outliers
        else
```

```
            x=['-310',num2str(i),'h.csv'];
            temp = xlsread(x);
            if temp(:,66) > 0
                %if there is a 4th charge or no 3rd charge, disregard
the heat
            else %script
                %load in and set the core variables
                AccumulatedKWH = 1000*(temp(:,1));
                PowerOnTime = temp(:,3);
                PowerOffTime = temp(:,4);
                I1 = temp(:,4);
                I2 = temp(:,5);
                I3 = temp(:,6);
                I2H1 = temp(:,44);
                I2H2 = temp(:,45);
                I2H3 = temp(:,46);
                TotGas = temp(:,60)*100;
                TotO2 = temp(:,61)*100;
                TotCarbon = temp(:,62);
                ChWt1 = temp(:,63);
                ChWt2 = temp(:,64);
                ChWt3 = temp(:,65);
                HeatEnd = temp(:,69);
                PercentMelted = temp(:,91);



                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                ChargeLast = find(ChWt1,1,'last')   %the last cell #
that has charge weight
                ChargeLast2 = find(ChWt2,1,'last')

                if ChargeLast == ChargeLast2   %if there is no
discrepancies, operate

                    %I2H is the average of the 3 phases
                    I2H = I2H1;
                    I2H(:) = (I2H1(:)+I2H2(:)+I2H3(:))/3;
                    O2Last = find(TotO2,1,'last')           %O2,carbon,
and gas reset early, so find their last position
                    %Get the cutoffs for the charges
                    SecondChargeSrt = find(ChWt2,1)     %find where the
second charge starts
                    ThirdChargeSrt = find(ChWt3,1)      %find where the
third charge starts
                    %Find charge weight sizes
                    ChWt1Size = ChWt1(ChargeLast-1)       %the total
weight of charge 1
                    ChWt2Size = ChWt2(ChargeLast-1)       %the total
weight of charge 2
                    ChWt3Size = ChWt3(ChargeLast-1)       %the total
weight of charge 3
```

```
                    %find charge weights
                    chwt1 = ChWt1(SecondChargeSrt-10)
                    chwt2 = ChWt2(SecondChargeSrt+10)
                    chwt3 = ChWt3(ThirdChargeSrt+10)
                    i


                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



                    %put all the parameters in their per ton form

%                     TotO2 = TotO2 - 2.3.*TotGas - 13*TotCarbon;
                    HeatKWHpt = AccumulatedKWH./(ChWt1+ChWt2+ChWt3);
                    HeatO2pt = TotO2./(ChWt1+ChWt2+ChWt3);
                    HeatGaspt = TotGas./(ChWt1+ChWt2+ChWt3);
                    HeatCarbonpt = TotCarbon./(ChWt1+ChWt2+ChWt3);

                    %normalize percent melted

                    PercentMelted1 = (PercentMelted(1:SecondChargeSrt-
1)/PercentMelted(SecondChargeSrt-1))*100;
                    PercentMelted2 =
(PercentMelted(SecondChargeSrt:ThirdChargeSrt-
1)/PercentMelted(ThirdChargeSrt-1))*100;
                    PercentMelted3 =
(PercentMelted(ThirdChargeSrt:ChargeLast)/PercentMelted(ChargeLast))*10
0;
                    PercentMeltedNorm = [PercentMelted1; PercentMelted2;
PercentMelted3];



                    %sum up all the parameters
                    totI1 = [totI1; I1];
                    totI2 = [totI2; I2];
                    totI3 = [totI3; I3];
                    totI2H = [totI2H; I2H];
                    totAccumulatedKWH = [totAccumulatedKWH;
AccumulatedKWH];
                    totPowerOffTime = [totPowerOffTime; PowerOffTime];
                    totPowerOnTime = [totPowerOnTime; PowerOnTime];
                    totHeatKWHpt = [totHeatKWHpt; HeatKWHpt];
                    totHeatO2pt = [totHeatO2pt; HeatO2pt];
                    totHeatGaspt = [totHeatGaspt; HeatGaspt];
                    totHeatCarbonpt = [totHeatCarbonpt; HeatCarbonpt];
                    totPercentMelted = [totPercentMelted;
PercentMeltedNorm];
                else
                end
            end
        end
    else
```

```
        x=['-31',num2str(i),'h.csv'];
        if i == 196 || i == 195
            %these ones are outliers
        else

            temp = xlsread(x);

            if ANY(temp(:,66))
                %if there is a 4th charge or no 3rd charge, disregard
the heat

            else %script

                %load in and set the core variables
                AccumulatedKWH = 1000*(temp(:,1));
                PowerOnTime = temp(:,3);
                PowerOffTime = temp(:,4);
                I1 = temp(:,4);
                I2 = temp(:,5);
                I3 = temp(:,6);
                I2H1 = temp(:,44);
                I2H2 = temp(:,45);
                I2H3 = temp(:,46);
                TotGas = temp(:,60)*100;
                TotO2 = temp(:,61)*100;
                TotCarbon = temp(:,62);
                ChWt1 = temp(:,63);
                ChWt2 = temp(:,64);
                ChWt3 = temp(:,65);
                HeatEnd = temp(:,69);
                PercentMelted = temp(:,91);



                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                ChargeLast = find(ChWt1,1,'last')   %the last cell #
that has charge weight
                ChargeLast2 = find(ChWt2,1,'last')

                if ChargeLast == ChargeLast2   %if there is no
discrepancies, operate

                    %I2H is the average of the 3 phases
                    I2H = I2H1;
                    I2H(:) = (I2H1(:)+I2H2(:)+I2H3(:))/3;
                    O2Last = find(TotO2,1,'last')          %O2,carbon,
and gas reset early, so find their last position
                    %Get the cutoffs for the charges
                    SecondChargeSrt = find(ChWt2,1)     %find where the
second charge starts
```

```
                    ThirdChargeSrt = find(ChWt3,1)      %find where the
third charge starts
                    %Find charge weight sizes
                    ChWt1Size = ChWt1(ChargeLast-1)       %the total
weight of charge 1
                    ChWt2Size = ChWt2(ChargeLast-1)       %the total
weight of charge 2
                    ChWt3Size = ChWt3(ChargeLast-1)       %the total
weight of charge 3
                    %find charge weights
                    chwt1 = ChWt1(SecondChargeSrt-10)
                    chwt2 = ChWt2(SecondChargeSrt+10)
                    chwt3 = ChWt3(ThirdChargeSrt+10)
                    i


                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                    %put all the parameters in their per ton form

                    TotO2 = TotO2 - 2.3.*TotGas - 13*TotCarbon;
                    HeatKWHpt = AccumulatedKWH./(ChWt1+ChWt2+ChWt3);
                    HeatO2pt = TotO2./(ChWt1+ChWt2+ChWt3);
                    HeatGaspt = TotGas./(ChWt1+ChWt2+ChWt3);
                    HeatCarbonpt = TotCarbon./(ChWt1+ChWt2+ChWt3);

                    %normalize percent melted

                    PercentMelted1 = (PercentMelted(1:SecondChargeSrt-
1)/PercentMelted(SecondChargeSrt-1))*100;
                    PercentMelted2 =
(PercentMelted(SecondChargeSrt:ThirdChargeSrt-
1)/PercentMelted(ThirdChargeSrt-1))*100;
                    PercentMelted3 =
(PercentMelted(ThirdChargeSrt:ChargeLast)/PercentMelted(ChargeLast))*10
0;
                    PercentMeltedNorm = [PercentMelted1; PercentMelted2;
PercentMelted3];


                    %sum up all the parameters
                    totI1 = [totI1; I1];
                    totI2 = [totI2; I2];
                    totI3 = [totI3; I3];
                    totI2H = [totI2H; I2H];
                    totAccumulatedKWH = [totAccumulatedKWH;
AccumulatedKWH];
                    totPowerOffTime = [totPowerOffTime; PowerOffTime];
                    totPowerOnTime = [totPowerOnTime; PowerOnTime];
                    totHeatKWHpt = [totHeatKWHpt; HeatKWHpt];
                    totHeatO2pt = [totHeatO2pt; HeatO2pt];
                    totHeatGaspt = [totHeatGaspt; HeatGaspt];
                    totHeatCarbonpt = [totHeatCarbonpt; HeatCarbonpt];
                    totPercentMelted = [totPercentMelted;
PercentMeltedNorm];

                else
                end
            end
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%define the overall length of the sum of vectors
[vectorlength width] = size(totPowerOnTime)

%KWH least squares model
totPowerOnTime2 = [0; totPowerOnTime(1:vectorlength-1)];
itot = totI1+totI2+totI3;
itot2 = [0; itot(1:vectorlength-1)];

% %the new model
% KWH_y = totAccumulatedKWH;
% KWH_y2 = [0; KWH_y(1:vectorlength-1)];
% KWH_X = [KWH_y2 totPowerOnTime2];
% KWH_theta_hat = (((KWH_X')*KWH_X)^-1)*(KWH_X')*KWH_y

%the old model
totPowerOnTime2 = [0; totPowerOnTime(1:vectorlength-1)];
KWH_X = [ones(size(totAccumulatedKWH)) totPowerOnTime2];
KWH_y = totAccumulatedKWH;
KWH_theta_hat = (((KWH_X')*KWH_X)^-1)*(KWH_X')*KWH_y
KWH_hat = KWH_X*KWH_theta_hat;




%i2h least squares model
%old model
totI12=[0; totI1(1:vectorlength-1)];
totI22=[0; totI2(1:vectorlength-1)];
totI32=[0; totI3(1:vectorlength-1)];
i2h_X = [totPowerOnTime2 totI12 totI22 totI32];
i2h_y = totI2H;
i2h_theta_hat = (((i2h_X')*i2h_X)^-1)*(i2h_X')*i2h_y
i2h_hat = i2h_X*i2h_theta_hat;
% %new model sum of currents
% itot = totI1+totI2+totI3;
% itot2 = [0; itot(1:vectorlength-1)];
% i2h_X = [totPowerOnTime2 itot2];
% i2h_y = totI2H;
% i2h_theta_hat = (((i2h_X')*i2h_X)^-1)*(i2h_X')*i2h_y
% %new model using previous i2h out
% i2h_y = totI2H;
% i2h_y2 = [0; i2h_y(1:vectorlength-1)];
% i2h_X = [i2h_y2 itot2];
% i2h_theta_hat = (((i2h_X')*i2h_X)^-1)*(i2h_X')*i2h_y

% %new model with i2h_y2 limited
% itot = totI1+totI2+totI3;
% itot2 = [0; itot(1:vectorlength-1)];
% i2h_y = totI2H;
% i2h_y2 = [0; i2h_y(1:vectorlength-1)];
% % i2h_newy = i2h_y - i2h_y2*0.98;
% i2h_X = [totPowerOnTime2 totI1 totI2 totI3];
% i2h_theta_hat = (((i2h_X')*i2h_X)^-1)*(i2h_X')*i2h_y;
% % i2h_theta_hat = [0.98; i2h_theta_hat]
% % i2h_X = [i2h_y2 totPowerOnTime2 itot2];






%Percent melted least squares model
totHeatKWHpt2 = [0; totHeatKWHpt(1:vectorlength-1)];
totHeatO2pt2 = [0; totHeatO2pt(1:vectorlength-1)];
totHeatGaspt2 = [0; totHeatGaspt(1:vectorlength-1)];
totHeatCarbonpt2 = [0; totHeatCarbonpt(1:vectorlength-1)];
totPowerOnTime2 = [0; totPowerOnTime(1:vectorlength-1)];
totPowerOffTime2 = [0; totPowerOffTime(1:vectorlength-1)];
totChargeKWHpt2 = [0; totChargeKWHpt(1:vectorlength-1)];
totChargeO2pt2 = [0; totChargeO2pt(1:vectorlength-1)];
totChargeGaspt2 = [0; totChargeGaspt(1:vectorlength-1)];
totChargeCarbonpt2 = [0; totChargeCarbonpt(1:vectorlength-1)];

%finds Pmeltd_hat with Pmeltd_y2 unconstrained
Pmeltd_y = totPercentMelted;
Pmeltd_y2 = [0; Pmeltd_y(1:vectorlength-1)];
%Pmeltd_X = [totHeatKWHpt2 totHeatO2pt2 totHeatGaspt2 totHeatCarbonpt2
totPowerOnTime2];
%Pmeltd_X = [Pmeltd_y2 totHeatKWHpt2 totHeatO2pt2 totHeatGaspt2
totHeatCarbonpt2 itot2];
Pmeltd_X = [totChargeKWHpt2 totHeatKWHpt2 totChargeO2pt2 totHeatO2pt2
totChargeGaspt2 totHeatGaspt2 totChargeCarbonpt2 totHeatCarbonpt2
totPowerOnTime2 totPowerOffTime2];
Pmeltd_theta_hat = (((Pmeltd_X')*Pmeltd_X)^-1)*(Pmeltd_X')*Pmeltd_y
Pmeltd_hat = Pmeltd_X*Pmeltd_theta_hat;

% %finds Pmeltd_hat with Pmeltd_y2 constrained
% Pmeltd_y = totPercentMelted;
% Pmeltd_y2 = [0; Pmeltd_y(1:vectorlength-1)];
% Pmeltd_newy = Pmeltd_y - Pmeltd_y2*0.95;
% Pmeltd_X = [totHeatKWHpt2 totHeatO2pt2 totHeatGaspt2 totHeatCarbonpt2
totPowerOnTime2];
```

```
% %Pmeltd_X = [totHeatKWHpt2 totHeatO2pt2 totHeatGaspt2
totHeatCarbonpt2 totPowerOnTime2 totPowerOffTime2];
% Pmeltd_theta_hat = (((Pmeltd_X')*Pmeltd_X)^-
1)*(Pmeltd_X')*Pmeltd_newy
% Pmeltd_theta_hat = [0.95; Pmeltd_theta_hat]
% Pmeltd_X = [Pmeltd_y2 totHeatKWHpt2 totHeatO2pt2 totHeatGaspt2
totHeatCarbonpt2 totPowerOnTime2];
% Pmeltd_hat = Pmeltd_X*Pmeltd_theta_hat;


% %start the fmincon stuff
% beta = abs(random('Normal',0,1,1,5))';
% alpha = abs(random('Normal',0,1,1,1));
% yhat = zeros(1,vectorlength);
% Pmeltd_X = [totHeatKWHpt totHeatO2pt totHeatGaspt totHeatCarbonpt
totPowerOnTime];
% Pmeltd_y = totPercentMelted;
%
% for k=2:vectorlength
% yhat(k) = alpha*yhat(k-1) + Pmeltd_X(k-1,:)*beta;
% end
% J = (yhat' - Pmeltd_y)'*(yhat' - Pmeltd_y)




GFKWH =1-(sum((KWH_hat-KWH_y).^2)/(sum((KWH_y-
mean(KWH_y)).^2))).^(0.5);
GFI2H=1-(sum((i2h_hat-i2h_y).^2)/(sum((i2h_y-  mean(i2h_y)).^2))).^(0.5);
GFPM=1-(sum((Pmeltd_hat-Pmeltd_y).^2)/(sum((Pmeltd_y-
mean(Pmeltd_y)).^2))).^(0.5);

close all
%plot gof figure
figure
subplot(131)
plot(KWH_hat)
hold on
plot(KWH_y(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('KWH, gof=',num2str(GFKWH)))
grid
subplot(132)
plot(i2h_hat)
hold on
plot(i2h_y(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('I2H, gof=',num2str(GFI2H)))
grid
subplot(133)
plot(Pmeltd_hat)
hold on
plot(Pmeltd_y(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('PM, gof=',num2str(GFPM)))
```

```
grid

% figure
% subplot(2,1,1)
% plot(Pmeltd_y)
% hold on
% plot(Pmeltd_hat,':')
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Percent Melted')
% title('$\hat{PM}$ dashed, ${PM}$ solid line','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% err=Pmeltd_hat-Pmeltd_y;
% C = xcorr(err);
% plot([-length(Pmeltd_hat)+1:length(Pmeltd_hat)-1],C)
% set(gca,'FontSize',16)
% xlabel('Lag')
% ylabel('PM')
% title('Autocorrelation of the Error','Interpreter','latex')
% set(gca,'FontSize',11)
%
%
%
%
%
%
%
% %
% figure
% i2h_hat = i2h_X*i2h_theta_hat;
% subplot(2,1,1)
% plot(i2h_y)
% hold on
% plot(i2h_hat,':')
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Electrode Consumption')
% title('$\hat{I^2H}$ dashed, ${I^2H}$ solid
line','Interpreter','latex')
% set(gca,'FontSize',11)
%
% subplot(2,1,2)
% err=i2h_hat-i2h_y;
% % relerr = abs(err./i2h_y);
% % plot(relerr)
% C = xcorr(err);
% plot([-length(i2h_hat)+1:length(i2h_hat)-1],C)
% set(gca,'FontSize',16)
% xlabel('Lag')
% ylabel('I^2H')
% title('Autocorrelation of the Error','Interpreter','latex')
% set(gca,'FontSize',11)
% %
% %
% figure
% subplot(2,1,1)
% i2h_hat = i2h_X*i2h_theta_hat;
% err=i2h_hat-i2h_y;
% abserr=abs(i2h_hat-i2h_y);
% relerr = abs(err./i2h_y);
% plot(abserr)
% axis([0 1400 0 80])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Absolute Error','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% plot(relerr)
% axis([0 1400 0 0.3])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Relative Error','Interpreter','latex')
% set(gca,'FontSize',11)


% figure
% subplot(2,1,1)
% KWH_hat = KWH_X*KWH_theta_hat;
% plot(KWH_y)
% hold on
% plot(KWH_hat,':')
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('KWH')
% title('$\hat{KWH}$ dashed, ${KWH}$ solid line','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% err=KWH_hat-KWH_y;
% % relerr = abs(err./KWH_y);
% % plot(relerr)
% C = xcorr(err);
% plot([-length(KWH_hat)+1:length(KWH_hat)-1],C)
% set(gca,'FontSize',16)
% xlabel('Lag')
% ylabel('KWH')
% title('Autocorrelation of the Error','Interpreter','latex')
% set(gca,'FontSize',11)
%
% figure
% subplot(2,1,1)
% KWH_hat = KWH_X*KWH_theta_hat;
% err=KWH_hat-KWH_y;
% abserr=abs(KWH_hat-KWH_y);
% relerr = abs(err./KWH_y);
% plot(abserr)
% axis([0 1400 0 800])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Absolute Error','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% plot(relerr)
% axis([0 1400 0 0.3])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Relative Error','Interpreter','latex')
% set(gca,'FontSize',11)

%
%
%
%
%
%
% figure
% plot(totPowerOnTime2,KWH_y)
% set(gca,'FontSize',16)
% xlabel('Time in minutes')
% ylabel('KWH')
% title('KWH vs. power on time','Interpreter','latex')
% set(gca,'FontSize',11)



% figure
% subplot(2,1,1)
% err=Pmeltd_hat-Pmeltd_y;
% abserr=abs(Pmeltd_hat-Pmeltd_y);
% relerr = abs(err./Pmeltd_y);
% plot(abserr)
% axis([0 1400 0 40])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Absolute Error','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% plot(relerr)
% axis([0 1400 0 3])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Relative Error','Interpreter','latex')
% set(gca,'FontSize',11)


% figure
% subplot(2,1,1)
% plot(Pmeltd_y)
% hold on
% plot(Pmeltd_hat,':')
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Percent Melted')
% title('$\hat{PM}$ dashed, ${PM}$ solid line','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% err=Pmeltd_hat-Pmeltd_y;
% C = xcorr(err);
% plot([-length(Pmeltd_hat)+1:length(Pmeltd_hat)-1],C)
% set(gca,'FontSize',16)
% xlabel('Lag')
% ylabel('PM')
% title('Autocorrelation of the Error','Interpreter','latex')
% set(gca,'FontSize',11)
```

B.2: Source Listing: HeatTotalsMasterFile.m

```
%this file compiles all the parameters from the three optimal control
%charge files.  It then compares the optimal solution to the original
%It also does the graphing.
%choose your target file
TargetFile='-29965h.csv';

%read in plot and cost variables
```

```
[Jo1,Jn1,x1,xKWH1,xI2H1,PM1,KWH1,I2H1,Itot1,accO21,accGas1,accCarbon1,D
I1,O21,Gas1,Carbon1,coePM1,coeKWH1,coeI2H1,preKWH1,preI2H1,prePM1]=firs
tchargefunction(TargetFile);
initKWH=xKWH1(find(xKWH1,1,'last'));
initI2H=xI2H1(find(xI2H1,1,'last'));
initO2=accO21(find(accO21,1,'last'));
initGas=accGas1(find(accGas1,1,'last'));
initCarbon=accCarbon1(find(accCarbon1,1,'last'));
[Jo2,Jn2,x2,xKWH2,xI2H2,PM2,KWH2,I2H2,Itot2,accO22,accGas2,accCarbon2,D
I2,O22,Gas2,Carbon2,coePM2,coeKWH2,coeI2H2,preKWH2,preI2H2,prePM2]=seco
ndchargefunction(TargetFile,initKWH,initI2H,initO2,initGas,initCarbon);
initKWH2=xKWH2(find(xKWH2,1,'last'));
initI2H2=xI2H2(find(xI2H2,1,'last'));
initO22=accO22(find(accO22,1,'last'));
initGas2=accGas2(find(accGas2,1,'last'));
initCarbon2=accCarbon2(find(accCarbon2,1,'last'));
[Jo3,Jn3,x3,xKWH3,xI2H3,PM3,KWH3,I2H3,Itot3,accO23,accGas3,accCarbon3,D
I3,O23,Gas3,Carbon3,coePM3,coeKWH3,coeI2H3,preKWH3,preI2H3,prePM3]=thir
dchargefunction(TargetFile,initKWH2,initI2H2,initO22,initGas2,initCarbo
n2);

%Total up quantities
TotalJn = Jn3;
TotalJo = Jo3;
x = [x1 x2 x3];
xKWH = [xKWH1 xKWH2 xKWH3];
xI2H = [xI2H1 xI2H2 xI2H3];
PM = [PM1;PM2;PM3]';
KWH = [KWH1;KWH2;KWH3]';
I2H = [I2H1;I2H2;I2H3]';
Itot = [Itot1 Itot2 Itot3];
accO2 = [accO21 accO22+initO2 accO23+initO2+initO22];
accGas = [accGas1 accGas2+initGas accGas3+initGas+initGas2];
accCarbon = [accCarbon1 accCarbon2+initCarbon
accCarbon3+initCarbon+initCarbon2];
DI = [DI1;DI2;DI3]';
O2 = [O21;O22;O23]';
Gas = [Gas1;Gas2;Gas3]';
Carbon = [Carbon1;Carbon2;Carbon3]';
preKWH = [preKWH1;preKWH2;preKWH3];
preI2H = [preI2H1;preI2H2;preI2H3];
prePM = [prePM1;prePM2;prePM3];
altPM = [PM1(2:end-1);PM2(2:end-1);PM3(2:end-1)];
altKWH = [KWH1(2:end-1);KWH2(2:end-1);KWH3(2:end-1)];
altI2H = [I2H1(2:end-1);I2H2(2:end-1);I2H3(2:end-1)];
GFKWH =1-(sum((preKWH-altKWH).^2)/(sum((altKWH-
mean(altKWH)).^2))).^(0.5);
GFI2H=1-(sum((preI2H-altI2H).^2)/(sum((altI2H-
mean(altI2H)).^2))).^(0.5);
GFPM=1-(sum((prePM-altPM).^2)/(sum((altPM- mean(altPM)).^2))).^(0.5);
ErrRatio=TotalJn/TotalJo

close all
%plot gof figure
figure
subplot(131)
```

```
plot(preKWH)
hold on
plot(KWH(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('KWH, gof=',num2str(GFKWH)))
grid
subplot(132)
plot(preI2H)
hold on
plot(I2H(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('I2H, gof=',num2str(GFI2H)))
grid
subplot(133)
plot(prePM)
hold on
plot(PM(2:end-1),'r-.')
set(gca,'FontSize',16)
title(strcat('PM, gof=',num2str(GFPM)))
grid


% %plot big picture figure
% figure;
% %title(TargetFile,'Interpreter','latex')
% subplot(2,4,1);plot(x,':');grid;title('PM');
% hold on; plot(PM,'r')
% subplot(2,4,2);plot(xKWH,':');grid;title('KWH');
% hold on; plot(KWH,'r')
% subplot(2,4,3);plot(xI2H,':');grid;title('I2H');
% hold on; plot(I2H,'r')
% subplot(2,4,4);plot(Itot,':');grid;title('Current');
% hold on; plot(DI,'r')
% subplot(2,4,5);plot(accO2,':');grid;title('accumulated O_2');
% hold on; plot(O2,'r')
% subplot(2,4,6);plot(accGas,':');grid;title('accumulated Gas');
% hold on; plot(Gas,'r')
% subplot(2,4,7);plot(accCarbon,':');grid;title('accumulated Carbon');
% hold on; plot(Carbon,'r')
% subplot(2,4,8);
% set(gca,'FontSize',16)
% text(0.1,0.8,strcat('Jnew= ',num2str(TotalJn)))
% text(0.1,0.5,strcat('Jold= ',num2str(TotalJo)))
% text(0.1,0.2,strcat('Jnew/Jold= ',num2str(TotalJn/TotalJo)))
% axis off
```

B.3: Source Listing: HeatComparison2.m

```
%this file compiles all the coefficents from the TargetFile
%Then it gathers the parameters from the three optimal control
%charge files for CompFile.  It then compares the optimal solution to
the original
%It also does the graphing.
%choose your target file (coefficents will be calculated from this file)
TargetFile='-30050h.csv';
```

```
%choose your comparison file
CompFile='-30005h.csv';

%read in plot and cost variables for target file to find the 3 sets of
%coefficents from the 3 different charges
[Jo1,Jn1,x1,xKWH1,xI2H1,PM1,KWH1,I2H1,Itot1,accO21,accGas1,accCarbon1,D
I1,O21,Gas1,Carbon1,coePM1,coeKWH1,coeI2H1]=firstchargefunction(TargetF
ile);
[Jo2,Jn2,x2,xKWH2,xI2H2,PM2,KWH2,I2H2,Itot2,accO22,accGas2,accCarbon2,D
I2,O22,Gas2,Carbon2,coePM2,coeKWH2,coeI2H2]=secondchargefunction(Target
File,initKWH,initI2H,initO2,initGas,initCarbon);
[Jo3,Jn3,x3,xKWH3,xI2H3,PM3,KWH3,I2H3,Itot3,accO23,accGas3,accCarbon3,D
I3,O23,Gas3,Carbon3,coePM3,coeKWH3,coeI2H3]=thirdchargefunction(TargetF
ile,initKWH2,initI2H2,initO22,initGas2,initCarbon2);

%read in plot and cost variables
[Jo1,Jn1,x1,xKWH1,xI2H1,PM1,KWH1,I2H1,Itot1,accO21,accGas1,accCarbon1,D
I1,O21,Gas1,Carbon1,coePM1,coeKWH1,coeI2H1]=firstchargefunctioncompare(
CompFile,coePM1,coeKWH1,coeI2H1);
initKWH=xKWH1(find(xKWH1,1,'last'));
initI2H=xI2H1(find(xI2H1,1,'last'));
initO2=accO21(find(accO21,1,'last'));
initGas=accGas1(find(accGas1,1,'last'));
initCarbon=accCarbon1(find(accCarbon1,1,'last'));
[Jo2,Jn2,x2,xKWH2,xI2H2,PM2,KWH2,I2H2,Itot2,accO22,accGas2,accCarbon2,D
I2,O22,Gas2,Carbon2,coePM2,coeKWH2,coeI2H2]=secondchargefunctioncompare
(CompFile,initKWH,initI2H,initO2,initGas,initCarbon,coePM2,coeKWH2,coeI
2H2);
initKWH2=xKWH2(find(xKWH2,1,'last'));
initI2H2=xI2H2(find(xI2H2,1,'last'));
initO22=accO22(find(accO22,1,'last'));
initGas2=accGas2(find(accGas2,1,'last'));
initCarbon2=accCarbon2(find(accCarbon2,1,'last'));
[Jo3,Jn3,x3,xKWH3,xI2H3,PM3,KWH3,I2H3,Itot3,accO23,accGas3,accCarbon3,D
I3,O23,Gas3,Carbon3,coePM3,coeKWH3,coeI2H3]=thirdchargefunctioncompare(
CompFile,initKWH2,initI2H2,initO22,initGas2,initCarbon2,coePM3,coeKWH3,
coeI2H3);


%Total up quantities
TotalJn = Jn3
TotalJo = Jo3
x = [x1 x2 x3];
xKWH = [xKWH1 xKWH2 xKWH3];
xI2H = [xI2H1 xI2H2 xI2H3];
PM = [PM1;PM2;PM3]';
KWH = [KWH1;KWH2;KWH3]';
I2H = [I2H1;I2H2;I2H3]';
Itot = [Itot1 Itot2 Itot3];
accO2 = [accO21 accO22+initO2 accO23+initO2+initO22];
accGas = [accGas1 accGas2+initGas accGas3+initGas+initGas2];
accCarbon = [accCarbon1 accCarbon2+initCarbon
accCarbon3+initCarbon+initCarbon2];
DI = [DI1;DI2;DI3]';
O2 = [O21;O22;O23]';
Gas = [Gas1;Gas2;Gas3]';
```

```
Carbon = [Carbon1;Carbon2;Carbon3]';
ErrRatio=TotalJn/TotalJo

%plot
close all
figure;
%title(TargetFile,'Interpreter','latex')
subplot(2,4,1);plot(x,':');grid;title('PM');
hold on; plot(PM,'r')
subplot(2,4,2);plot(xKWH,':');grid;title('KWH');
hold on; plot(KWH,'r')
subplot(2,4,3);plot(xI2H,':');grid;title('I2H');
hold on; plot(I2H,'r')
subplot(2,4,4);plot(Itot,':');grid;title('Current');
hold on; plot(DI,'r')
subplot(2,4,5);plot(accO2,':');grid;title('accumulated O_2');
hold on; plot(O2,'r')
subplot(2,4,6);plot(accGas,':');grid;title('accumulated Gas');
hold on; plot(Gas,'r')
subplot(2,4,7);plot(accCarbon,':');grid;title('accumulated Carbon');
hold on; plot(Carbon,'r')
subplot(2,4,8);
text(0.1,0.8,strcat('Jnew= ',num2str(TotalJn)))
text(0.1,0.5,strcat('Jold= ',num2str(TotalJo)))
text(0.1,0.2,strcat('Jnew/Jold= ',num2str(TotalJn/TotalJo)))
axis off
```

B.4: Source Listing: firstchargefunction.m

```
%this file calculates the coefficents and then finds the optimal
solution
%for the first charge of the file given by HeatTotalsMasterFile.m and
%HeatComparison2.m
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H,preKWH,preI2H,prePM] =
firstchargefunction(xx)
```

```
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(1);
ke = KE(1);
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*temp(kf:ke,60);
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*temp(kf:ke,61);
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

%                  %find the free oxygen
%                  FreeO2 = TotO2 - 2.3*TotGas - 13*TotCarbon;
%                  %penalize gas and carbon only if there is not enough
free
%                  %oxygen
%                  alpha1=0.5;
%                  alpha2=2-alpha1;
%                  for h=1:ChargeLast
%                      if FreeO2(h) < 0
%                      beta(h) = TotO2(h)/(alpha1*2.3*TotGas(h) +
alpha2*13*TotCarbon(h));
%                      FreeO2(h) = 0;
%                      TotGas(h) = beta(h)*TotGas(h);
%                      TotCarbon(h) = beta(h)*TotCarbon(h);
%                      end
%                  end


PM = temp(kf:ke,91);
PM = PM/max(PM)*100;
%PM = (1-PM)*100;

%estimate of coefficients of KWH(k)=a KWH(k-1)+b DI(k-1)
%a is fixed at 1
YKWH=KWH(2:kl)-KWH(1:kl-1);
AKWH=DI(1:kl-1);
coeKWH=inv(AKWH'*AKWH)*AKWH'*YKWH;
preKWH=KWH(1:kl-1)+coeKWH*DI(1:kl-1);
GFKWH =1-(sum((preKWH-KWH(2:kl)).^2)/(sum((KWH(2:kl)-
mean(KWH(2:kl))).^2))).^(0.5);
% subplot(131)
% plot(preKWH)
% hold on
% plot(KWH(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('KWH, gof=',num2str(GFKWH)))
% grid

%estimate of coefficients of I2H(k)=a I2H(k-1)+b DI(k-1)
%a is fixed at 1
YI2H=I2H(2:kl)-I2H(1:kl-1);
AI2H=DI(1:kl-1);
coeI2H=inv(AI2H'*AI2H)*AI2H'*YI2H;
preI2H=I2H(1:kl-1)+coeI2H*DI(1:kl-1);
GFI2H=1-(sum((preI2H-I2H(2:kl)).^2)/(sum((I2H(2:kl)-
mean(I2H(2:kl))).^2))).^(0.5);
% subplot(132)
% plot(preI2H)
% hold on
% plot(I2H(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('I2H, gof=',num2str(GFI2H)))
% grid

%estimate of coefficients of
%PM(k)=a PM(k-1)+b DI(k-1)+c O2(k-1)+d Gas(k-1)+e Carbon(k-1)
%a is fixed at 1
YPM=PM(2:kl)-PM(1:kl-1);
APM=[KWH(1:kl-1)- [0;KWH(1:kl-2)] DO2(1:kl-1) DGas(1:kl-1)
DCarbon(1:kl-1)];
APM1=[DI(1:kl-1) DO2(1:kl-1) DGas(1:kl-1) DCarbon(1:kl-1)];

coePM=inv(APM'*APM)*APM'*YPM;
coePM(1)=coePM(1)*coeKWH;
prePM=PM(1:kl-1)+APM1*coePM;
```

```
GFPM=1-(sum((prePM-PM(2:kl)).^2)/(sum((PM(2:kl)-
mean(PM(2:kl))).^2))).^(0.5);
% subplot(133)
% plot(prePM)
% hold on
% plot(PM(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('PM, gof=',num2str(GFPM)))
% grid

%optimal control with end constraint
F =1;

G =coePM'

SN =0;

R = [0.0003 0 0 0;0 6 0 0;0 0 14 0;0 0 0 8500];
Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end

xKWH(1)=0;
xI2H(1)=0;
for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;

    %     u(:,kk) = max(u(:,kk),0);
    %
    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
```

```
        u(1,kk)=0;
    end
%     u(2,kk) = max(0,min(u(2,kk),(ChWt1*1000)/kl));
%     if kk > 0.6*kl
%         u(3,kk) = 0;
%     else
%         u(3,kk) = max(0,min(u(3,kk),(ChWt1*300)/kl));
%     end
%     u(4,kk) = max(0,min(u(4,kk),(ChWt1*23)/kl));
    %     u(3,kk) = max(0,min(u(3,kk),100/kl));
    %     u(4,kk) = max(0,min(u(4,kk),2000/kl));
    %     u(2,kk) = max(0,min(u(2,kk),350/kl));
    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end

% figure;
% subplot(2,4,1);plot(x);grid;title('PM');
% hold on; plot(PM,'r')
% subplot(2,4,2);plot(xKWH);grid;title('KWH');
% hold on; plot(KWH,'r')
% subplot(2,4,3);plot(xI2H);grid;title('I2H');
% hold on; plot(I2H,'r')

Itot=u(1,:);
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));

% subplot(2,4,4);plot(Itot);grid;title('Current');
% hold on; plot(DI,'r')
% subplot(2,4,5);plot(accO2);grid;title('accumulated O_2');
% hold on; plot(O2,'r')
% subplot(2,4,6);plot(accGas);grid;title('accumulated Gas');
% hold on; plot(Gas,'r')
% subplot(2,4,7);plot(accCarbon);grid;title('accumulated Carbon');
% hold on; plot(Carbon,'r')

nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;

% figure
% subplot(2,1,1)
% err=preI2H-I2H;
% abserr=abs(preI2H-I2H);
% relerr = abs(err./I2H);
% plot(abserr)
% axis([0 1400 0 80])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Absolute Error','Interpreter','latex')
% set(gca,'FontSize',11)
% subplot(2,1,2)
% plot(relerr)
% axis([0 1400 0 0.3])
% set(gca,'FontSize',16)
% xlabel('Samples')
% ylabel('Error')
% title('Relative Error','Interpreter','latex')
% set(gca,'FontSize',11)

% subplot(2,4,8);
% text(0.1,0.8,strcat('Jnew= ',num2str(Jnew)))
% text(0.1,0.5,strcat('Jold= ',num2str(Jold)))
% text(0.1,0.2,strcat('Jnew/Jold= ',num2str(Jnew/Jold)))
% axis off
```

B.5: Source Listing: secondchargefunction.m

```
%this file calculates the coefficents and then finds the optimal
solution
%for the second charge of the file given by HeatTotalsMasterFile.m and
%HeatComparison2.m
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H,preKWH,preI2H,prePM] =
secondchargefunction(xx,initKWH,initI2H,initO2,initGas,initCarbon)
```

```
% temp = xlsread('-30025h.csv');
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(2);
ke = KE(2);
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*(temp(kf:ke,60));
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*(temp(kf:ke,61));
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

PM = temp(kf:ke,91);
PM = PM/max(PM)*100;
%PM = (1-PM)*100;

%estimate of coefficients of KWH(k)=a KWH(k-1)+b DI(k-1)
%a is fixed at 1
YKWH=KWH(2:kl)-KWH(1:kl-1);
AKWH=DI(1:kl-1);
coeKWH=inv(AKWH'*AKWH)*AKWH'*YKWH;
```

```
preKWH=KWH(1:kl-1)+coeKWH*DI(1:kl-1);
GFKWH =1-(sum((preKWH-KWH(2:kl)).^2)/(sum((KWH(2:kl)-
mean(KWH(2:kl))).^2))).^(0.5);
% figure
% subplot(131)
% plot(preKWH)
% hold on
% plot(KWH(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('KWH, gof=',num2str(GFKWH)))
% grid

%estimate of coefficients of I2H(k)=a I2H(k-1)+b DI(k-1)
%a is fixed at 1
YI2H=I2H(2:kl)-I2H(1:kl-1);
AI2H=DI(1:kl-1);
coeI2H=inv(AI2H'*AI2H)*AI2H'*YI2H;
preI2H=I2H(1:kl-1)+coeI2H*DI(1:kl-1);
GFI2H=1-(sum((preI2H-I2H(2:kl)).^2)/(sum((I2H(2:kl)-
mean(I2H(2:kl))).^2))).^(0.5);
% subplot(132)
% plot(preI2H)
% hold on
% plot(I2H(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('I2H, gof=',num2str(GFI2H)))
% grid

%estimate of coefficients of
%PM(k)=a PM(k-1)+b DI(k-1)+c O2(k-1)+d Gas(k-1)+e Carbon(k-1)
%a is fixed at 1
YPM=PM(2:kl)-PM(1:kl-1);
APM=[KWH(1:kl-1)- [0;KWH(1:kl-2)] DO2(1:kl-1) DGas(1:kl-1)
DCarbon(1:kl-1)];
APM1=[DI(1:kl-1) DO2(1:kl-1) DGas(1:kl-1) DCarbon(1:kl-1)];

coePM=inv(APM'*APM)*APM'*YPM;
coePM(1)=coePM(1)*coeKWH;
prePM=PM(1:kl-1)+APM1*coePM;
GFPM=1-(sum((prePM-PM(2:kl)).^2)/(sum((PM(2:kl)-
mean(PM(2:kl))).^2))).^(0.5);
% subplot(133)
% plot(prePM)
% hold on
% plot(PM(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('PM, gof=',num2str(GFPM)))
% grid


%optimal control with end constraint
F =1;

G =coePM'
```

```
SN =0;

R = [0.0000012 0 0 0;0 1 0 0;0 0 20 0;0 0 0 5000];

Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end
xKWH(1)=initKWH;
xI2H(1)=initI2H;
%xKWH(2)=initKWH;
%xI2H(2)=initI2H;
%u(2,1)=initO2;
%u(3,1)=initGas;
%u(4,1)=initCarbon;
%  xKWH(1)=0;
%  xI2H(1)=0;

for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;

    %     u(:,kk) = max(u(:,kk),0);
    %
    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
        u(1,kk)=0;
    end
%     u(2,kk) = max(0,min(u(2,kk),(ChWt2*1000)/kl));
%     if kk > 0.6*kl
%         u(3,kk) = 0;
%     else
%         u(3,kk) = max(0,min(u(3,kk),(ChWt2*300)/kl));
%     end
%     u(4,kk) = max(0,min(u(4,kk),(ChWt2*23)/kl));
%     %     u(3,kk) = max(0,min(u(3,kk),100/kl));
%     %     u(4,kk) = max(0,min(u(4,kk),2000/kl));
%     %     u(2,kk) = max(0,min(u(2,kk),350/kl));
    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end

% figure;
% subplot(2,4,1);plot(x);grid;title('PM');
% hold on; plot(PM,'r')
% subplot(2,4,2);plot(xKWH);grid;title('KWH');
% hold on; plot(KWH,'r')
% subplot(2,4,3);plot(xI2H);grid;title('I2H');
% hold on; plot(I2H,'r')

Itot=u(1,:);
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));

% subplot(2,4,4);plot(Itot);grid;title('Current');
% hold on; plot(DI,'r')
% subplot(2,4,5);plot(accO2);grid;title('accumulated O_2');
% hold on; plot(O2,'r')
% subplot(2,4,6);plot(accGas);grid;title('accumulated Gas');
% hold on; plot(Gas,'r')
% subplot(2,4,7);plot(accCarbon);grid;title('accumulated Carbon');
% hold on; plot(Carbon,'r')

nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;

% subplot(2,4,8);
% text(0.1,0.8,strcat('Jnew= ',num2str(Jnew)))
% text(0.1,0.5,strcat('Jold= ',num2str(Jold)))
% text(0.1,0.2,strcat('Jnew/Jold= ',num2str(Jnew/Jold)))
% axis off
```

B.6: Source Listing: thirdchargefunction.m

```
%this file calculates the coefficents and then finds the optimal
solution
%for the third charge of the file given by HeatTotalsMasterFile.m and
%HeatComparison2.m
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H,preKWH,preI2H,prePM] =
thirdchargefunction(xx,initKWH,initI2H,initO2,initGas,initCarbon)
```

```
% temp = xlsread('-30025h.csv');
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(3);
ke = FinalValue-20;
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*(temp(kf:ke,60));
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*(temp(kf:ke,61));
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

PM = temp(kf:ke,91);
PM = PM/max(PM)*100;
%PM = (1-PM)*100;

%estimate of coefficients of KWH(k)=a KWH(k-1)+b DI(k-1)
%a is fixed at 1
YKWH=KWH(2:kl)-KWH(1:kl-1);
```

```
AKWH=DI(1:kl-1);
coeKWH=inv(AKWH'*AKWH)*AKWH'*YKWH;
preKWH=KWH(1:kl-1)+coeKWH*DI(1:kl-1);
GFKWH =1-(sum((preKWH-KWH(2:kl)).^2)/(sum((KWH(2:kl)-
mean(KWH(2:kl))).^2))).^(0.5);
% figure
% subplot(131)
% plot(preKWH)
% hold on
% plot(KWH(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('KWH, gof=',num2str(GFKWH)))
% grid

%estimate of coefficients of I2H(k)=a I2H(k-1)+b DI(k-1)
%a is fixed at 1
YI2H=I2H(2:kl)-I2H(1:kl-1);
AI2H=DI(1:kl-1);
coeI2H=inv(AI2H'*AI2H)*AI2H'*YI2H;
preI2H=I2H(1:kl-1)+coeI2H*DI(1:kl-1);
GFI2H=1-(sum((preI2H-I2H(2:kl)).^2)/(sum((I2H(2:kl)-
mean(I2H(2:kl))).^2))).^(0.5);
% subplot(132)
% plot(preI2H)
% hold on
% plot(I2H(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('I2H, gof=',num2str(GFI2H)))
% grid

%estimate of coefficients of
%PM(k)=a PM(k-1)+b DI(k-1)+c O2(k-1)+d Gas(k-1)+e Carbon(k-1)
%a is fixed at 1
YPM=PM(2:kl)-PM(1:kl-1);
APM=[KWH(1:kl-1)- [0;KWH(1:kl-2)] DO2(1:kl-1) DGas(1:kl-1)
DCarbon(1:kl-1)];
APM1=[DI(1:kl-1) DO2(1:kl-1) DGas(1:kl-1) DCarbon(1:kl-1)];

coePM=inv(APM'*APM)*APM'*YPM;
coePM(1)=coePM(1)*coeKWH;
prePM=PM(1:kl-1)+APM1*coePM;
GFPM=1-(sum((prePM-PM(2:kl)).^2)/(sum((PM(2:kl)-
mean(PM(2:kl))).^2))).^(0.5);
% subplot(133)
% plot(prePM)
% hold on
% plot(PM(2:end),'r-.')
% set(gca,'FontSize',16)
% title(strcat('PM, gof=',num2str(GFPM)))
% grid


%optimal control with end constraint
F =1;

G =coePM'
```

```
SN =0;

R = [0.00000004 0 0 0;0 10 0 0;0 0 25 0;0 0 0 2500];

Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end

xKWH(1)=initKWH;
xI2H(1)=initI2H;
%xKWH(2)=initKWH;
%xI2H(2)=initI2H;
%u(2,1)=initO2;
%u(3,1)=initGas;
%u(4,1)=initCarbon;
%  xKWH(1)=0;
%  xI2H(1)=0;
for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;
    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
        u(1,kk)=0;
    end

    %     u(:,kk) = max(u(:,kk),0);
    %     u(2,kk) = max(0,min(u(2,kk),(ChWt3*1000)/kl));
    %     if kk > 0.6*kl
    %         u(3,kk) = 0;
    %     else
    %         u(3,kk) = max(0,min(u(3,kk),(ChWt3*300)/kl));
    %     end
    %     u(4,kk) = max(0,min(u(4,kk),(ChWt3*230000000)/kl));
    %     u(3,kk) = max(0,min(u(3,kk),100/kl));
    %     u(4,kk) = max(0,min(u(4,kk),2000/kl));
    %     u(2,kk) = max(0,min(u(2,kk),350/kl));
    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end

% figure;
% subplot(2,4,1);plot(x);grid;title('PM');
% hold on; plot(PM,'r')
% subplot(2,4,2);plot(xKWH);grid;title('KWH');
% hold on; plot(KWH,'r')
% subplot(2,4,3);plot(xI2H);grid;title('I2H');
% hold on; plot(I2H,'r')

Itot=u(1,:);
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));

% subplot(2,4,4);plot(Itot);grid;title('Current');
% hold on; plot(DI,'r')
% subplot(2,4,5);plot(accO2);grid;title('accumulated O_2');
% hold on; plot(O2,'r')
% subplot(2,4,6);plot(accGas);grid;title('accumulated Gas');
% hold on; plot(Gas,'r')
% subplot(2,4,7);plot(accCarbon);grid;title('accumulated Carbon');
% hold on; plot(Carbon,'r')
KE
KF
nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;

% subplot(2,4,8);
% text(0.1,0.8,strcat('Jnew= ',num2str(Jnew)))
% text(0.1,0.5,strcat('Jold= ',num2str(Jold)))
% text(0.1,0.2,strcat('Jnew/Jold= ',num2str(Jnew/Jold)))
% axis off
```

B.7: Source Listing: firstchargefunctioncompare.m

```
%this file uses coefficents from a different file and then finds the
optimal solution
%for the first charge of the file given by HeatComparison2.m
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H] =
firstchargefunctioncompare(xx,coePM,coeKWH,coeI2H)


% temp = xlsread('-30025h.csv');
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(1);
ke = KE(1);
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*temp(kf:ke,60);
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*temp(kf:ke,61);
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

PM = temp(kf:ke,91);
PM = PM/max(PM)*100;

%optimal control with end constraint
F =1;
```

```
G =coePM';

SN =0;

R = [0.0003 0 0 0;0 6 0 0;0 0 14 0;0 0 0 8500];
Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end

xKWH(1)=0;
xI2H(1)=0;
for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;

    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
        u(1,kk)=0;
    end

    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end

Itot=u(1,:);
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));
```

```
nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;
```

B.8: Source Listing: secondchargefunctioncompare.m

```
%this file uses coefficents from a different file and then finds the
optimal solution
%for the second charge of the file given by HeatComparison2.m
```

```
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H] =
secondchargefunctioncompare(xx,initKWH,initI2H,initO2,initGas,initCarbo
n,coePM,coeKWH,coeI2H)


% temp = xlsread('-30025h.csv');
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(2);
ke = KE(2);
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*(temp(kf:ke,60));
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*(temp(kf:ke,61));
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

PM = temp(kf:ke,91);
PM = PM/max(PM)*100;
```

```
%optimal control with end constraint
F =1;

G =coePM';

SN =0;

R = [0.0000012 0 0 0;0 1 0 0;0 0 20 0;0 0 0 5000];

Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end
xKWH(1)=initKWH;
xI2H(1)=initI2H;

for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;

    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
        u(1,kk)=0;
    end

    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end

Itot=u(1,:);
```

```
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));

nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;
```

B.9: Source Listing: thirdchargefunctioncompare.m

```
%this file uses coefficents from a different file and then finds the
optimal solution
%for the third charge of the file given by HeatComparison2.m
function
[Jold,Jnew,x,xKWH,xI2H,PM,KWH,I2H,Itot,accO2,accGas,accCarbon,DI,O2,Gas
,Carbon,coePM,coeKWH,coeI2H] =
thirdchargefunctioncompare(xx,initKWH,initI2H,initO2,initGas,initCarbon
,coePM,coeKWH,coeI2H)


% temp = xlsread('-30025h.csv');
temp = xlsread(xx);
%find where it starts and stops
KE=[];
KF=[];
PMtemp = temp(:,91);
FinalValue = find(PMtemp,1,'last');
for kk=2:FinalValue-1
    if PMtemp(kk)-PMtemp(kk-1)<-50
        KE=[KE;kk-5];
    end
    if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
        KF=[KF;kk];
    end
end

kf = KF(3);
ke = FinalValue-20;
kl=ke-kf;

%read in variables
KWH = 1000*(temp(kf:ke,1));

I1 = 10*temp(kf:ke,5);I2 = 10*temp(kf:ke,6);I3 = 10*temp(kf:ke,7);
DI = (I1+I2+I3)/3;

I2H1 = temp(kf:ke,44);
I2H2 = temp(kf:ke,45);
I2H3 = temp(kf:ke,46);
I2H  = (I2H1+I2H2+I2H3)/3;

ChWt1 = temp(kf+75,63);  %first charge weight
ChWt2 = temp(kf+75,64);  %second charge weight
ChWt3 = temp(kf+75,65);  %third charge weight

Gas = 100*(temp(kf:ke,60));
DGas=(Gas(2:kl)-Gas(1:kl-1));

Carbon = temp(kf:ke,62);
DCarbon=(Carbon(2:kl)-Carbon(1:kl-1));

O2 = 100*(temp(kf:ke,61));
```

```
FreeO2 = max(O2 - 0.5*2.3*Gas - 1.5*13*Carbon,0);
DO2=max(FreeO2(2:kl)-FreeO2(1:kl-1),0);

PM = temp(kf:ke,91);
PM = PM/max(PM)*100;

%optimal control with end constraint
F =1;

G =coePM';

SN =0;

R = [0.00000004 0 0 0;0 10 0 0;0 0 25 0;0 0 0 2500];

Q =0;

C = 1;
rr =100;

%%%%
N = kl;
x = zeros(1,N);
u = zeros(4,N);
x(1) =0;

P = zeros(1,N);
SK = zeros(1,N);
SK(N) = SN;
KN = zeros(4,N);
VK = zeros(1,N);
VK(N) = C;

for kk = N-1:-1:1
    KN(:,kk) = inv(G'*SK(kk+1)*G+R)*G'*SK(kk+1)*F;
    SK(kk) = F'*SK(kk+1)*(F-G*KN(:,kk))+Q;
    VK(kk) = (F-G*KN(:,kk))'* VK(kk+1);
    P(kk)  = P(kk+1) - VK(kk+1)'*G*inv(R+G'*SK(kk+1)*G)*G'*VK(kk+1);
end

xKWH(1)=initKWH;
xI2H(1)=initI2H;

for kk = 1:N-1
    Ku = inv(G'*SK(kk+1)*G + R)*G';

    u(:,kk) = -(KN(:,kk)-Ku*VK(kk+1)*inv(P(kk))*VK(kk)')*x(kk)...
        - Ku * VK(kk+1)* inv(P(kk))*rr;

    u(1,kk) = max(30000,min(u(1,kk),50000));
    if x(kk) > 100
        u(1,kk)=0;
    end
    x(kk+1) = F*x(kk) + G * u(:,kk);
    xKWH(kk+1)=xKWH(kk)+coeKWH*u(1,kk);
    xI2H(kk+1)=xI2H(kk)+coeI2H*u(1,kk);
end


Itot=u(1,:);
accO2=cumsum(u(2,:));
accGas=cumsum(u(3,:));
accCarbon=cumsum(u(4,:));

KE
KF
nI=sum(u(1,:));
nO2=sum(u(2,:));
nGas=sum(u(3,:));
nCarbon=sum(u(4,:));

Jold=[redacted]*KWH(end)+[redacted]*I2H(end)+[redacted]*O2(end)+[redacted]*Gas(end)+
[redacted]*Carbon(end);
Jnew=[redacted]*xKWH(end)+[redacted]*xI2H(end)+[redacted]*nO2+[redacted]*nGas+[redacted]*nCarbon;
```

B.10: Source Listing: ModelComparison.m

```
%this file compiles all the coefficents from the TargetFile
%then it uses them on a different file and compares the model to the
real
%It also does the graphing.
clear all
%choose your target file (coefficents will be calculated from this file)
TargetFile='-30096h.csv';
%choose your comparison file
CompFile='-30125h.csv';

%read in plot and cost variables for target file to find the 3 sets of
%coefficents from the 3 different charges
[Jo1,Jn1,x1,xKWH1,xI2H1,PM1,KWH1,I2H1,Itot1,accO21,accGas1,accCarbon1,D
I1,O21,Gas1,Carbon1,coePM1,coeKWH1,coeI2H1]=firstchargefunction(TargetF
ile);
initKWH=xKWH1(find(xKWH1,1,'last'));
initI2H=xI2H1(find(xI2H1,1,'last'));
initO2=accO21(find(accO21,1,'last'));
initGas=accGas1(find(accGas1,1,'last'));
initCarbon=accCarbon1(find(accCarbon1,1,'last'));
[Jo2,Jn2,x2,xKWH2,xI2H2,PM2,KWH2,I2H2,Itot2,accO22,accGas2,accCarbon2,D
I2,O22,Gas2,Carbon2,coePM2,coeKWH2,coeI2H2]=secondchargefunction(Target
File,initKWH,initI2H,initO2,initGas,initCarbon);
initKWH2=xKWH2(find(xKWH2,1,'last'));
initI2H2=xI2H2(find(xI2H2,1,'last'));
initO22=accO22(find(accO22,1,'last'));
initGas2=accGas2(find(accGas2,1,'last'));
initCarbon2=accCarbon2(find(accCarbon2,1,'last'));
[Jo3,Jn3,x3,xKWH3,xI2H3,PM3,KWH3,I2H3,Itot3,accO23,accGas3,accCarbon3,D
I3,O23,Gas3,Carbon3,coePM3,coeKWH3,coeI2H3]=thirdchargefunction(TargetF
ile,initKWH2,initI2H2,initO22,initGas2,initCarbon2);

%read in plot and cost variables
[Jo1,Jn1,x1,xKWH1,xI2H1,PM1,KWH1,I2H1,Itot1,accO21,accGas1,accCarbon1,D
I1,O21,Gas1,Carbon1,coePM1,coeKWH1,coeI2H1,preKWH1,preI2H1,prePM1]=firs
tchargefunctioncompare(CompFile,coePM1,coeKWH1,coeI2H1);
initKWH=xKWH1(find(xKWH1,1,'last'));
initI2H=xI2H1(find(xI2H1,1,'last'));
initO2=accO21(find(accO21,1,'last'));
initGas=accGas1(find(accGas1,1,'last'));
initCarbon=accCarbon1(find(accCarbon1,1,'last'));
[Jo2,Jn2,x2,xKWH2,xI2H2,PM2,KWH2,I2H2,Itot2,accO22,accGas2,accCarbon2,D
I2,O22,Gas2,Carbon2,coePM2,coeKWH2,coeI2H2,preKWH2,preI2H2,prePM2]=seco
ndchargefunctioncompare(CompFile,initKWH,initI2H,initO2,initGas,initCar
bon,coePM2,coeKWH2,coeI2H2);
initKWH2=xKWH2(find(xKWH2,1,'last'));
initI2H2=xI2H2(find(xI2H2,1,'last'));
initO22=accO22(find(accO22,1,'last'));
initGas2=accGas2(find(accGas2,1,'last'));
initCarbon2=accCarbon2(find(accCarbon2,1,'last'));
[Jo3,Jn3,x3,xKWH3,xI2H3,PM3,KWH3,I2H3,Itot3,accO23,accGas3,accCarbon3,D
I3,O23,Gas3,Carbon3,coePM3,coeKWH3,coeI2H3,preKWH3,preI2H3,prePM3]=thir
```

```
dchargefunctioncompare(CompFile,initKWH2,initI2H2,initO22,initGas2,init
Carbon2,coePM3,coeKWH3,coeI2H3);


%Total up quantities
PM = [PM1;PM2;PM3]';
KWH = [KWH1;KWH2;KWH3]';
I2H = [I2H1;I2H2;I2H3]';
preKWH = [preKWH1;preKWH2;preKWH3];
preI2H = [preI2H1;preI2H2;preI2H3];
prePM = [prePM1;prePM2;prePM3];
altPM = [PM1(2:end-1);PM2(2:end-1);PM3(2:end-1)];
altKWH = [KWH1(2:end-1);KWH2(2:end-1);KWH3(2:end-1)];
altI2H = [I2H1(2:end-1);I2H2(2:end-1);I2H3(2:end-1)];
GFKWH =1-(sum((preKWH-altKWH).^2)/(sum((altKWH-
mean(altKWH)).^2))).^(0.5)
GFI2H=1-(sum((preI2H-altI2H).^2)/(sum((altI2H-
mean(altI2H)).^2))).^(0.5)
GFPM=1-(sum((prePM-altPM).^2)/(sum((altPM- mean(altPM)).^2))).^(0.5)

% %close all
% %plot gof figure
% figure
% subplot(131)
% plot(preKWH)
% hold on
% plot(KWH(2:end-1),'r-.')
% set(gca,'FontSize',16)
% title(strcat('KWH, gof=',num2str(GFKWH)))
% grid
% subplot(132)
% plot(preI2H)
% hold on
% plot(I2H(2:end-1),'r-.')
% set(gca,'FontSize',16)
% title(strcat('I2H, gof=',num2str(GFI2H)))
% grid
% subplot(133)
% plot(prePM)
% hold on
% plot(PM(2:end-1),'r-.')
% set(gca,'FontSize',16)
% title(strcat('PM, gof=',num2str(GFPM)))
% grid
```

B.11: Source Listing: similarchargefinder2.m

```
clear all
%this file labels each charge as a group 1-3
%then it finds heats where all three charges
%are similar to the target charge
ct1groupnum = [];
ct2groupnum = [];
ct3groupnum = [];

%read in starting file
%srtfile = '-30000'h.csv'

%in this case, -30001h.csv is the file we're interested in
for i=1:299
    %read in the file
    if i<10
        x=['-3000',num2str(i),'h.csv'];
        temp = xlsread(x);
    elseif i>9 && i<100
        %these ones are outliers
        if i == 14 || i == 15 || i == 30 || i == 51 || i == 52 || i ==
54 || i == 59 || i == 89

        else
            x=['-300',num2str(i),'h.csv'];
            temp = xlsread(x);
        end
    elseif i>99 && i<1000
        if i == 101 || i == 134 || i == 135 || i == 155 || i == 158 ||
i == 179 || i == 180 || i == 191 || i == 192 || i == 194 || i == 196 ||
i == 199 ...
                || i == 200 || i == 203 || i == 204 || i == 212 || i ==
213 || i == 214 || i == 215 || i == 216 || i == 224 || i == 227 || i ==
228 || i == 229 ...
                || i == 252 || i == 253 || i == 254 || i == 256 || i ==
261 || i == 268 || i == 269 || i == 271 || i == 272 || i == 280 || i ==
281 || i == 282 ...
                || i == 283 || i == 284
        else
            x=['-30',num2str(i),'h.csv'];
            temp = xlsread(x);
        end
    elseif i>999 && i<10000
        x=['-3',num2str(i),'h.csv'];
        temp = xlsread(x);
    end

    %track the file number
    i
    %find the start and endpoints and length of each charge
    KE=[];
    KF=[];
    PMtemp = temp(:,91);
    FinalValue = find(PMtemp,1,'last');
    for kk=2:FinalValue-1
        if PMtemp(kk)-PMtemp(kk-1)<-50
            KE=[KE;kk-5];
        end
        if PMtemp(kk+1)>0 && PMtemp(kk)==0 && PMtemp(kk-1)==0
            KF=[KF;kk];
        end
    end
    kl1=KE(1)-KF(1);
    kl2=KE(2)-KF(2);
    kl3=(FinalValue-20)-KF(3);

    %find charge weights in tons
    ChWt1 = temp(KF(1)+75,63);
    ChWt2 = temp(FinalValue-20,64);
    ChWt3 = temp(FinalValue-20,65);

    %put each file charge in a group 1-3 according to weight
    %put each file charge in a group 1-3 according to length of charge
    if ChWt1<33
        ct1groupnum(i)=1;
    elseif ChWt1<35 && ChWt1>33
        ct1groupnum(i)=2;
    else
        ct1groupnum(i)=3;
    end

    if ChWt2<33
        ct2groupnum(i)=1;
    elseif ChWt2<35 && ChWt2>33
        ct2groupnum(i)=2;
    else
        ct2groupnum(i)=3;
    end

    if ChWt3<18
        ct3groupnum(i)=1;
    elseif ChWt3<22 && ChWt3>19
        ct3groupnum(i)=2;
    else
        ct3groupnum(i)=3;
    end

end
%set the outliers to 0
ct1groupnum(14) = 0;
ct2groupnum(14) = 0;
ct3groupnum(14) = 0;
ct1groupnum(15) = 0;
ct2groupnum(15) = 0;
ct3groupnum(15) = 0;
ct1groupnum(30) = 0;
ct2groupnum(30) = 0;
ct3groupnum(30) = 0;
ct1groupnum(51) = 0;
ct2groupnum(51) = 0;
```

```
ct3groupnum(51) = 0;
ct1groupnum(52) = 0;
ct2groupnum(52) = 0;
ct3groupnum(52) = 0;
ct1groupnum(54) = 0;
ct2groupnum(54) = 0;
ct3groupnum(54) = 0;
ct1groupnum(59) = 0;
ct2groupnum(59) = 0;
ct3groupnum(59) = 0;
ct1groupnum(89) = 0;
ct2groupnum(89) = 0;
ct3groupnum(89) = 0;
%second hundred
ct1groupnum(101) = 0;
ct2groupnum(101) = 0;
ct3groupnum(101) = 0;
ct1groupnum(134) = 0;
ct2groupnum(134) = 0;
ct3groupnum(134) = 0;
ct1groupnum(135) = 0;
ct2groupnum(135) = 0;
ct3groupnum(135) = 0;
ct1groupnum(155) = 0;
ct2groupnum(155) = 0;
ct3groupnum(155) = 0;
ct1groupnum(158) = 0;
ct2groupnum(158) = 0;
ct3groupnum(158) = 0;
ct1groupnum(179) = 0;
ct2groupnum(179) = 0;
ct3groupnum(179) = 0;
ct1groupnum(180) = 0;
ct2groupnum(180) = 0;
ct3groupnum(180) = 0;
ct1groupnum(191) = 0;
ct2groupnum(191) = 0;
ct3groupnum(191) = 0;
ct1groupnum(192) = 0;
ct2groupnum(192) = 0;
ct3groupnum(192) = 0;
ct1groupnum(194) = 0;
ct2groupnum(194) = 0;
ct3groupnum(194) = 0;
ct1groupnum(196) = 0;
ct2groupnum(196) = 0;
ct3groupnum(196) = 0;
ct1groupnum(199) = 0;
ct2groupnum(199) = 0;
ct3groupnum(199) = 0;
%third hundred
ct1groupnum(200) = 0;
ct2groupnum(200) = 0;
ct3groupnum(200) = 0;
ct1groupnum(203) = 0;
ct2groupnum(203) = 0;
ct3groupnum(203) = 0;
```

```
ct1groupnum(204) = 0;
ct2groupnum(204) = 0;
ct3groupnum(204) = 0;
ct1groupnum(212) = 0;
ct2groupnum(212) = 0;
ct3groupnum(212) = 0;
ct1groupnum(213) = 0;
ct2groupnum(213) = 0;
ct3groupnum(213) = 0;
ct1groupnum(214) = 0;
ct2groupnum(214) = 0;
ct3groupnum(214) = 0;
ct1groupnum(215) = 0;
ct2groupnum(215) = 0;
ct3groupnum(215) = 0;
ct1groupnum(216) = 0;
ct2groupnum(216) = 0;
ct3groupnum(216) = 0;
ct1groupnum(224) = 0;
ct2groupnum(224) = 0;
ct3groupnum(224) = 0;
ct1groupnum(227) = 0;
ct2groupnum(227) = 0;
ct3groupnum(227) = 0;
ct1groupnum(228) = 0;
ct2groupnum(228) = 0;
ct3groupnum(228) = 0;
ct1groupnum(229) = 0;
ct2groupnum(229) = 0;
ct3groupnum(229) = 0;
ct1groupnum(252) = 0;
ct2groupnum(252) = 0;
ct3groupnum(252) = 0;
ct1groupnum(253) = 0;
ct2groupnum(253) = 0;
ct3groupnum(253) = 0;
ct1groupnum(254) = 0;
ct2groupnum(254) = 0;
ct3groupnum(254) = 0;
ct1groupnum(256) = 0;
ct2groupnum(256) = 0;
ct3groupnum(256) = 0;
ct1groupnum(261) = 0;
ct2groupnum(261) = 0;
ct3groupnum(261) = 0;
ct1groupnum(268) = 0;
ct2groupnum(268) = 0;
ct3groupnum(268) = 0;
ct1groupnum(269) = 0;
ct2groupnum(269) = 0;
ct3groupnum(269) = 0;
ct1groupnum(271) = 0;
ct2groupnum(271) = 0;
ct3groupnum(271) = 0;
ct1groupnum(272) = 0;
ct2groupnum(272) = 0;
ct3groupnum(272) = 0;
```

```
ct1groupnum(280) = 0;
ct2groupnum(280) = 0;
ct3groupnum(280) = 0;
ct1groupnum(281) = 0;
ct2groupnum(281) = 0;
ct3groupnum(281) = 0;
ct1groupnum(282) = 0;
ct2groupnum(282) = 0;
ct3groupnum(282) = 0;
ct1groupnum(283) = 0;
ct2groupnum(283) = 0;
ct3groupnum(283) = 0;
ct1groupnum(284) = 0;
ct2groupnum(284) = 0;
ct3groupnum(284) = 0;

%find file of interests numbers
srt1=ct1groupnum(2)
srt2=ct2groupnum(2)
srt3=ct3groupnum(2)

%now identify the nearest similar charges to the file of interest
i1=find(ct1groupnum==srt1)
i2=find(ct2groupnum==srt2)
i3=find(ct3groupnum==srt3)

%now find entire heats that have all three charge numbers in common
with
%the file of interest
CompFiles = [];
for i=1:299
    if ct1groupnum(i) == srt1 && ct2groupnum(i) == srt2 &&
ct3groupnum(i) == srt3
        CompFiles = [CompFiles; i];
    end
end
CompFiles
```

REFERENCES

1. Bryant Jr, Billy W. 2010. Interview with author. Iowa City, Iowa, 7 May.

2. Bryant Jr, Billy W. 2006. Identification, Simulation, and Optimization of an Electric Arc Furnace, The University of Iowa.

3. Bureau of Labor Statistics, U.S. Department of Labor, Career Guide to Industries, 2010-11 Edition, Steel Manufacturing, on the Internet at http://www.bls.gov/oco/cg/cgs014.htm (accessed June 2, 2010 *)*.

4. Hsia, T.C. 1977. System Identification. Lexington, Massachusetts: Lexington Books.

5. Lewis, Frank. 1986. Optimal Control. New York: John Wiley & Sons, Inc.

6. Leybovich, Ilya. May 20, 2010. World Steel Production Climbs in April. http://news.thomasnet.com/IMT/archives/2010/05/world-steel-production-climbs-in-april.html?t=recent (accessed June 2, 2010).

7. McBroom, N.H. 1983. Small Scale Steelmaking. New York: Applied Science Publishers.

8. Rathaba, Letsane Paul. 2005. Model Fitting For Electric Arc Furnace Refining, University of Pretoria.