Graduate Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2011

The Use of Advanced Statistical Concepts and Analysis to Improve Nonlinear Dynamic Glucose Modeling

Lucas Beverlin
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Statistics and Probability Commons

Recommended Citation
https://lib.dr.iastate.edu/etd/10071

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
The use of advanced statistical concepts and analysis to improve nonlinear dynamic glucose modeling

by

Lucas P. Beverlin

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Derrick Rollins, Major Professor
    Philip Dixon
    Mark Kaiser
    Heike Hoffman
    Warren Franke

Iowa State University
Ames, Iowa
2011
TABLE OF CONTENTS

LIST OF TABLES ............................................. v
LIST OF FIGURES ............................................. vii

CHAPTER 1. INTRODUCTION ................................. 1

CHAPTER 2. BACKGROUND AND LIMITATIONS OF CURRENT
METHODS IN GLUCOSE MODELING ......................... 4

2.1 The History and Role of Glucose Modeling in Improving Glucose Control 4
  2.1.1 Compartment Models ................................. 5
  2.1.2 Empirical Modeling ................................. 6
  2.1.3 Semi-empirical Modeling ............................ 7
  2.1.4 Feedback and Feedforward Control ................. 7
  2.1.5 Model Predictive Control ......................... 9
  2.1.6 Issues ............................................. 9
  2.1.7 Devices used in Monitoring Blood Glucose Concentrations .......... 10

2.2 Basic Approaches in Glucose Modeling ................ 11
  2.2.1 Autoregressive models ......................... 12
  2.2.2 ARMA models .................................. 14
  2.2.3 ARX models .................................. 16
  2.2.4 ARMAX models ................................ 17


**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1</td>
<td>Bounds and $s_i$ used for the first grid search.</td>
<td>45</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>A table of inputs used in the modeling work of Rollins et al. [86] Those in bold will be used in this work, and the number in parentheses denotes the input number, i.e. $v_1$ corresponds to carbohydrates.</td>
<td>47</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Results from fitting four weeks of data from four subjects. Here the first 1.5 weeks of the data was used in training, the next 1.5 weeks was used in validation, and the remainder in the test set.</td>
<td>49</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>A comparison of model fits. Auto denotes the algorithm presented in this paper was used to find the model while Manual denotes that Wiener networks fit manually via the Solver add-in for Microsoft Excel®.</td>
<td>50</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Input variables: Food (1); Armband (2-7)</td>
<td>62</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Results under Equation 4.5 for the 22 test subjects with 2 weeks of training and 2 weeks of validation. AE and AAE values are in mg/dL.</td>
<td>72</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Armband (inputs 2-8 in Table 1) modeling results for the 22 test subjects with 2 weeks of training and 2 weeks of validation. AE and AAE values are in mg/dL. AE is not reported for Training since it is 0 for all subjects.</td>
<td>73</td>
</tr>
</tbody>
</table>
Table 4.4  Modeling results for the 22 test subjects with 1 week of training and 1 week of validation. AE and AAE values are in mg/dL. AE is not reported for Training since it is 0 for all subjects.  

Table 4.5  Modeling results for the 22 test subjects with 3 days of training and no validation. AE and AAE values are in mg/dL.  

Table 5.1  A table of inputs used in the modeling work of Rollins et al. [86] Those in bold will be used in this work, and the number in parentheses denotes the input number, i.e. $v_1$ corresponds to carbohydrates.  

Table 5.2  Results from the calculation of forecast intervals. Terms is the number of error terms used in the KSAP model for predicting BGC $k$ steps ahead for subject $i$, Coverage denotes the proportion of observations $y_{t+k-1}$ contained in the 95% forecast intervals calculated from Equation 5.32.
LIST OF FIGURES

Figure 2.1 A graphical representation of the Wiener model used in Rollins et al. [86]. .......................................................... 21

Figure 2.2 A graphical representation of a neural network with three inputs $(X_i)$, two hidden nodes $(Z_i)$ and three outputs $(Y_j)$. .............. 24

Figure 3.1 A graphical representation of the Wiener model used in Rollins et al. [86]. .......................................................... 37

Figure 4.1 The SenseWear® Pro 3 armband of BodyMedia, Inc ............... 59

Figure 4.2 A graphical representation of the Wiener network. Each of the $p$ inputs, $x_i$, are passed through a separate linear dynamic block of $G_i$ and collected and passed through a static function $f(V)$ to give output $\eta$. ................................................................. 63

Figure 4.3 Modeling results for subject 10 with two weeks training and 2 weeks of validation showing the model fit under Equation 4.5 (left plot) and Equation 4.9 (right plot). ....................... 77

Figure 5.1 A graphical representation of the Wiener model used in Rollins et al. [86]. .......................................................... 84
Figure 5.2  The observed relationship between $k$, the use of inputs and/or outputs in the model and $r_{fit}$. Taken from [86]. 91

Figure 5.3  Histograms of the residuals for 1, 6, and 12 steps ahead for two subjects. 95

Figure 5.4  Plot of observed and predicted BGC as well as 95% forecast intervals for each of the models before and after a meal. 97
CHAPTER 1. INTRODUCTION

Diabetes mellitus is a condition where the body’s ability to control its blood glucose concentration (BGC) has been impaired or destroyed. Unfortunately it is a growing problem throughout the world, as Zhang et al. [114] report that the global health expenditure for diabetes is projected to be at least 375 billion US dollars in 2010 and at least 500 billion US dollars in 2030. There are many reasons for this, including a rising population, a longer lifespan, as well as poor eating and exercise habits. [115] This has caused diabetes to become the seventh leading cause of death in the United States as of 2007. [14]

There are two main types of diabetes mellitus. Both of them affect the body’s production of insulin, a hormone produced by the $\beta$ cells of the pancreas that triggers cells to uptake glucose for energy or fat storage. Type 1 diabetes mellitus is a condition where the $\beta$ cells are destroyed by the body. What triggers the body to destroy the $\beta$ cells is still being investigated, though it is believed to be an autoimmune response. [21] Type 2 diabetes mellitus is characterized by poor control of the body’s BGC. This can be caused by cells developing insulin resistance and/or the $\beta$ cells’ inability to produce a sufficient amount of insulin for glucose uptake. Those with type 2 diabetes are at higher risk for several other health problems, such as obesity, high blood pressure, nephropathy, neuropathy and blindness. [21] While the onset of type 1 diabetes generally occurs before the age of 30 and type 2 diabetes after the age of 30, an increasing number of people, particularly of Pacific Island or south Asian descent are experiencing the onset of type 2 diabetes in their
To decrease the risk for other health problems, diabetics need to tightly control their BGC. Typically this means to keep one’s BGC between 70 mg/dL and 180 mg/dL. One is said to be hyperglycemic when their BGC is above 180 mg/dL and hypoglycemic when under 70 mg/dL. When hyperglycemic, organs that do not require insulin to uptake glucose, such as the kidneys and eyes, are subjected to an excess of glucose, which causes damage. When hypoglycemic, the brain begins to shut down organs because the body does not have enough glucose to keep them running. Thus if one is hypoglycemic for too long, one can experience diabetic shock, a coma, or even death.

From this one can see that the development of an accurate glucose model could have a significant impact on glucose control, and thus improve the well-being of a type 2 diabetic. Thus the main objective of this work is to enhance glucose modeling via the advancement of statistical methods and theory to achieve the ultimate goal of the development of a noninvasive continuous glucose monitoring system (CGMS). To this end, since one’s BGC is dynamic and the relationship it has with other factors is complex, the use of a nonlinear dynamic model is a necessity. Thus the following conditions and model assumptions define the scope of the proposed research:

1. Non-insulin dependent type 2 diabetes mellitus
2. Free-living conditions
3. 2nd order nonlinear dynamic behavior
4. Serially correlated noise
5. Multiple inputs
6. Time and space variant parameters
The development of an accurate glucose model is very important to the development of a noninvasive CGMS. This noninvasive CGMS must be able to provide accurate predictions of one's BGC under free-living conditions, i.e. conditions that are typical for a human being during the course of their day. This will not be an easy task, as a type 2 diabetic's ability to regulate their BGC will degrade over time. Thus regardless of the model chosen, the parameters will be time variant. The sensors needed to measure multiple inputs must also be highly accurate. This glucose model must also not depend on insulin, as this device is intended for those who do not take insulin.

In the next chapter several approaches that have been proposed for modeling BGC will be discussed. After this, a discussion on the devices that have been used to monitor BGC noninvasively will be given. Then an overview of several current models for predicting BGC and current methods for calculating \((1 - \alpha)100\%\) prediction intervals for forecasting purposes will be given. In subsequent chapters, papers to be submitted for publication that address the issues of building a model that accurately predicts BGC and the construction of predictions intervals for BGC are given.

My contributions to each chapter are as follows. For Chapter 3, I created the algorithm and wrote the paper. For Chapter 4, I helped in collecting the data from the subjects and contributed the result given in the appendix for that chapter. For Chapter 5, I generated the results, performed the derivations and wrote the paper.
CHAPTER 2. BACKGROUND AND LIMITATIONS OF CURRENT METHODS IN GLUCOSE MODELING

In the literature there are several methods that have been suggested for use in glucose modeling. These methods vary widely in their structure and have had varied success. While each of these has its strengths, each also has its issues. In this chapter several models that have been used for glucose modeling will be discussed. After briefly discussing their history, a short history on the use of noninvasive devices for measuring BGC will be given. After this several models that are currently being used will be introduced. Finally current methods for calculating $(1 - \alpha)100\%$ prediction intervals for future values of a response variable will be given.

2.1 The History and Role of Glucose Modeling in Improving Glucose Control

Researchers have been modeling one’s BGC for over 50 years in hopes of finding an accurate model for one’s BGC. This has occurred despite the fact that the technology needed to continuously monitor one’s BGC under free-living conditions has only become commercially available in the last ten years. In this section several approaches that have been used are introduced, and the section concludes with a discussion of the issues found in each of the studies mentioned in this work.
2.1.1 Compartment Models

Compartment models were the first models used in the modeling of blood glucose and insulin. Compartment models are models described by a set of linear differential equations. A major advantage of compartment models is that one can have multiple inputs and multiple outputs of interest. For example, a simplified compartment model for tracking changes in blood glucose and insulin concentrations when glucose is administered intravenously is suggested by Bolie [10]:

\[
\frac{dx}{dt} = p - \alpha x + \beta y \tag{2.1}
\]

\[
\frac{dy}{dt} = q - \gamma x - \delta y. \tag{2.2}
\]

Here \( x \) represents the measured extracellular insulin concentration and \( y \) the measured extracellular glucose concentration, both centered with respect to the mean insulin and glucose concentrations, respectively. Also \( p \) denotes the rate of insulin injection divided by the total extracellular fluid volume and \( q \) the rate of glucose injection divided by the total extracellular fluid volume. Finally the coefficients of \( x \) and \( y \) in these differential equations are parameters to be estimated. Segre et al. [92] modified these equations to allow for the use of insulin infusion and Ackerman et al. [1] modified these equations for an oral glucose tolerance test. More recent work in modeling BGC with compartment models include Topp et al. [103], Ribbing et al. [84], and Karimipour and Shandiz [52].

Cobelli et al. [21] note two major drawbacks to Bolie’s model. One is the simplistic assumption of a linear relationship between insulin secretion and glucose. The other is that this model does not take into account the complex interaction between production of glucose by the liver and uptake of glucose and insulin. For modeling BGC in type 2 diabetics, another drawback is that the true rate of glucose injection from the digestive system and the amount of insulin in the blood cannot be measured noninvasively. Thus
determining the optimal rate of insulin infusion at a given time would be very difficult.

### 2.1.2 Empirical Modeling

Empirical models are models that do not take into account any information about the true model. In this situation, one chooses a model and determines parameter estimates from collected data. An advantage to this is that parameterization may be kept to a minimum, but an obvious problem is that if the model structure chosen is grossly inaccurate, then determining an accurate model from the chosen model structure may be impossible.

Several empirical models have been used in the literature. The simplest such model is the autoregressive model. Sparacino et al. [96] fit first-order autoregressive model with time-varying parameters to predict blood glucose concentrations 30 minutes into the future with a fairly high degree of accuracy. Reifman et al. [83] achieve similar results with a first-order autoregressive model fit using the regularized least squares method, which introduces a small amount of bias in return for a sizeable reduction in the variance of the parameters.

The next empirical model to consider is the autoregressive moving average (ARMA) model. Eren-Oruklu et al. [25] use an ARMA model to predict future BGC. Ståhl and Johansson [99] consider several models, including compartment models, AR models and ARMA models for predicting current and future BGC.

Autoregressive exogenous input (ARX) models have also been used. Pedersen and Hansen [79] consider ARX models in the prediction of a type 1 diabetic’s BGC. Finan et al. [28] consider them for predicting BGC in simulated data. Finan et al. [27] use them while investigating the effect of smoothing input data.

Another empirical model used for modeling BGC is the autoregressive moving average model with exogenous variables (ARMAX model). Ståhl and Johansson [99] consider them in their comparison of models. Eren-Oruklu et al. [25] use an ARMAX model in order to
predict the onset of hypoglycemia. Finan et al. [27] consider these models as well while investigating the effect of smoothing the input data on the ability of an ARMAX model to predict BGC.

The last empirical model to be discussed is the artificial neural network. Mougiakakou and Nikita [71] use them to determine the amount of insulin to administer for a type 1 diabetic in order to control their BGC. Paddada et al. [77] and Tresp et al. [104] use neural networks to predict BGC in type 1 diabetics.

2.1.3 Semi-empirical Modeling

Semi-empirical models are models that use first-principle, i.e. biological, chemical, etc., laws in conjunction with an empirical model. This allows for models where some information about the true model is known, but there is still some uncertainty in the true model structure. The only semi-empirical model to be considered in this work is the Wiener network. While it has not been used extensively for modeling in the diabetes field, it has been used extensively in the chemical engineering field to model gas concentrations [101] and pH levels [76], among other things. Rollins et al. [86] use this model to monitor the BGC of a type 2 diabetic. Ståhl and Johansson [99, 98] use such a model to predict BGC 2 hours into the future in type 1 diabetics.

2.1.4 Feedback and Feedforward Control

Feedback control is a technique that has been used in chemical engineering for decades. In feedback control, one takes measurements of some output of interest and determines the adjustment of the process controlling that output from these measurements in order to optimize the output in some fashion. As an example, type 1 diabetics have used this to control their BGC for decades. If they measure their BGC and find that it is low, then
they will ingest some carbohydrates which will increase their BGC. If their BGC is too high, then they administer insulin to themselves. Of course, if their BGC is within their target range, then their BGC is already optimal, and typically they will do nothing.

As for feedforward control, one adjusts the process controlling the outputs of interest based on what he believes will happen in the near future in order to negate any undesirable effects on the process before they can happen. Type 1 diabetics have some experience in this as well, as they will alter their insulin prescription before they go to bed in order to counteract the dawn phenomenon [2], which is an increase in BGC during the early morning hours due to a rise in hormonal levels. They may also do so if they know they are about to ingest a large quantity of carbohydrates.

Models have been created that utilize feedback and/or feedforward control. Fisher [30] uses it to model BGC in type 1 diabetics while Salzsieger et al. [89] use it to try to determine the optimal insulin delivery for a type 1 diabetic. Ruiz-Velázquez et al. [88] use it for the same purpose in a simulation. Fisher and Teo [31] investigate three insulin infusion programs that allows one to recover from a hyperglycemic state, one of which uses feedback control. Tolić et al. [102] note that there are two sets of oscillations in the delivery of insulin, one of which is observed over 80-150 minutes when insulin is released by the pancreas and another over 8-15 minutes, and that delivering insulin in this manner via feedback control could improve control over BGC. Others have used both feedback and feedforward control in order to try to tightly control BGC. Marchetti et al. [67] use a compartment model first derived by Hovorka et al. [49] and then revised by Wilinska et al. [110] to implement such control.
2.1.5 Model Predictive Control

Model predictive control (MPC) is a technique used in process control to make changes to inputs in order to minimize the deviation of a response variable from a set point. This has become one of the most popular techniques for modeling BGC. Gillis et al. [40] use MPC in order to determine a type 1 diabetic’s BGC in response to a meal. Dua et al. [23] use MPC to determine online changes to the optimal insulin delivery rate, which in their approach was calculated offline. Nonlinear MPC has been used for glucose control during fasting for type 1 diabetics [93], and for controlling BGC during the overnight hours. [49]

MPC and nonlinear MPC have also been used in silico, i.e. with a simulator of a type 1 diabetic’s BGC that has been approved by the FDA. [22, 56] This allows researchers to simulate glucose data for hundreds of sample people without the need to physically collect data. Lee et al. [59] used MPC to reduce the average BGC of an in silico type 1 diabetic by 15-20 mg/dL. Magni et al. [66] consider MPC and nonlinear MPC in their BGC control strategy and conclude that nonlinear MPC controls BGC better.

2.1.6 Issues

A major issue with each of these approaches is the number of inputs used in modeling. Every study that was found in the literature only used glucose, insulin, carbohydrates and/or exercise. While insulin will not be available for use in a noninvasive CGMS, there are other factors that affect glucose. Stress, both physical and emotional, can affect BGC. [46] It has also been shown that blood glucose and insulin concentrations exhibit a circadian rhythm. [44, 106] In particular, one’s BGC tends to increase during the morning hours before waking up due to an increase in hormones, which is known as the dawn phenomenon. [2]
Another major issue with many of these approaches is that they are tested on simulated data. If an accurate model is to be found for use in a noninvasive CGMS, it must be tested under free-living conditions. This is because the noninvasive CGMS will be used while one is performing their daily activities, and so there will be variations observed under free-living conditions that may not be seen in simulation.

2.1.7 Devices used in Monitoring Blood Glucose Concentrations

To monitor one’s BGC, one has typically used a CGMS that requires a sensor to be inserted into one’s subcutaneous layer. This sensor can be uncomfortable, and there appears to be a lag between a rise in BGC and a rise in glucose levels in the subcutaneous layer. \cite{38} This can be very dangerous in the presence of hypoglycemia, as one’s BGC could be dangerously low before the CGMS predicts the onset of hypoglycemia. Finally while the accuracy of a CGMS has increased over the past 10 years, they are still not as accurate as glucose meters. \cite{20, 37, 57} Several ideas have been suggested to measure one’s BGC noninvasively: reverse iontophoresis \cite{82}, sonophoresis \cite{55}, passive diffusion \cite{100} and microporation \cite{54}, but only reverse iontophoresis appears to have any real chance at commercial success.

Reverse iontophoresis is the use of electric current to pull a sample of interstitial fluid from the skin. One such product, which was produced by Cygnus, Inc., now a part of Johnson and Johnson (New Brunswick, NJ), was approved by the FDA in 2001 and worn on the wrist like a watch. \cite{80, 108} This device is calibrated by a measurement from a glucose meter, and it can give warnings if it determines BGC to be too high or too low, as specified by the user. However, around 10% of subjects could not use the device due to skin irritation, and roughly 25% of the measurements from the device were found to be inaccurate. \cite{100} Thus the device was discontinued in 2007. Another device that
uses reverse iontophoresis is the Symphony™ transdermal continuous glucose monitoring system, produced by Echo Therapeutics (Franklin, MA). Chuang et al. [19] report that its accuracy is on par with other FDA approved glucose meters.

There are several obstacles that must be overcome in order to successfully develop a noninvasive CGMS. The first is that a set of noninvasive inputs must be identified that can be accurately mapped to BGC. Not only does this include determining which inputs, but also determining a model structure. Once this is done, steps must be taken to ensure that these inputs are measured accurately. Given the complexity of the relationship between the inputs and BGC, another obstacle will be to devise a method for quickly finding parameter estimates that yield an accurate model of one's BGC without overfitting. While parameter estimates can be found to yield a highly accurate model for training data quickly, these models tend to be extremely inaccurate on validation data in this work. The final obstacle is the development of sound statistical inference methods for this model. There has been very little research done on inference methods for the choice of model in this work, the Wiener network. This work addresses the latter two obstacles.

In the next section, some background on current methods for modeling BGC are given as well as their strengths and weaknesses. This includes an introduction to autoregressive models, neural networks and Wiener networks.

### 2.2 Basic Approaches in Glucose Modeling

In this section many of the models used in glucose modeling are introduced. It will be shown that many of these models can be represented by a differential equation. Methods on fitting these models will be discussed as well as the advantages and disadvantages of using these models.
2.2.1 Autoregressive models

The basic autoregressive AR(p) model is written as

\[ y_m = \phi_0 + \sum_{i=1}^{p} \phi_i y_{m-i} + e_m, \]  

(2.3)

where \( y_m \) is the value of the response variable \( Y \) at timepoint \( m \), \( p \) is the order of the model, \( \phi_i, i = 0, \ldots, p \) are parameters to be estimated and \( e_m \) is a white noise disturbance term with mean 0 and variance \( \sigma^2 \). Here assume that the timepoints are equally spaced, i.e. the amount of time that passes between timepoint \( m \) and timepoint \( m + 1 \) is \( \Delta t \). This model is frequently used when the current measurement of \( Y \) is highly correlated with previous measurements of \( Y \).

To see that this model can be derived from a differential equation, first consider an AR(2) model:

\[ y_m = \phi_0 + \phi_1 y_{m-1} + \phi_2 y_{m-2} + e_m. \]  

(2.4)

It will be shown that Equation 2.4 can be represented by the second order differential equation

\[ c_1 \frac{d^2y}{dt^2}(t) + c_2 \frac{dy}{dt}(t) + y(t) = c_4 + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) e(t), \]  

(2.5)

where \( t \) denotes the time corresponding to timepoint \( m \) and each \( c_i \) is a constant. First one must approximate the derivatives \( \frac{dy}{dt}(t) \) and \( \frac{d^2y}{dt^2}(t) \). One can approximate \( \frac{dy}{dt}(t) \) with a backward difference approximation:

\[ \frac{dy}{dt}(t) \approx \frac{y_m - y_{m-1}}{\Delta t}. \]  

(2.6)

Now \( \frac{d^2y}{dt^2}(t) \) can be approximated in a similar manner:

\[ \frac{d^2y}{dt^2}(t) \approx \frac{\frac{dy}{dt}(t) - \frac{dy}{dt}(t-\Delta t)}{\Delta t} \approx \frac{y_m - 2y_{m-1} + y_{m-2}}{\Delta t^2}. \]  

(2.7)
From these approximations, Equation 2.5 can be discretized and rewritten as

\[ c_1 \frac{y_m - 2y_{m-1} + y_{m-2}}{\Delta t^2} + c_2 \frac{y_m - y_{m-1}}{\Delta t} + y_m = c_4 + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) \epsilon_m \]  

(2.8)

\[ \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) y_m - \left( \frac{2c_1}{\Delta t^2} + \frac{c_2}{\Delta t} \right) y_{m-1} + \frac{c_1}{\Delta t^2} y_{m-2} = c_4 + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) \epsilon_m \]  

(2.9)

\[ (c_1 + c_2 \Delta t + \Delta t^2) y_m - (2c_1 + c_2 \Delta t) y_{m-1} + c_1 y_{m-2} = c_4 \Delta t^2 + (c_1 + c_2 \Delta t + \Delta t^2) \epsilon_m \]  

(2.10)

\[ y_m = \frac{c_4 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} + \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-1} - \frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-2} + \epsilon_m \]  

(2.11)

Setting \( \phi_0 = \frac{c_4 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \phi_1 = \frac{2c_1 - c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} \) and \( \phi_2 = -\frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} \) yields the AR(2) model. To generalize, an AR(p) model can be represented with a \( p^{th} \) order differential equation.

There are two steps to fitting an AR(p) model. First, one must determine the best value of \( p \). This can be done by calculating the partial autocorrelation function of lag \( p \) [15]. A value of \( p \) is chosen such that the partial autocorrelation function value is near 0 for every value greater than or equal to \( p \). Once \( p \) has been determined, the Yule-Walker equations [15] are set up and solved to estimate the parameters as well as \( \sigma^2 \). One can also minimize the sum of squared residuals, which is often called the least squares objective function, with respect to the parameters to be estimated to achieve this, as an explicit solution can be calculated for this model.

The main advantage of using an AR model is that there exists a closed form solution for estimating the parameters. Thus work done by Skrøvseth et al. [95] and Sparacino et al. [96] have used AR models to predict BGC. However, determining \( p \) can be difficult at times, since a judgment call based on a plot of the autocorrelation function is made to determine the value of \( p \) such that the autocorrelation function of some lag \( q > p \) is not significantly different than 0. The residuals may not exhibit white noise either. One way to overcome this is to use an autoregressive moving average (ARMA) model.
2.2.2 ARMA models

An autoregressive moving average ARMA(p,q) model can be written as

\[ y_m = \beta_0 + \sum_{i=1}^{p} \beta_i y_{m-i} + \sum_{j=1}^{q} \phi_j e_{m-j} + e_m \]  

(2.12)

where \( y_m \) is the value of the response variable \( Y \) at timepoint \( m \) and \( e_m \) is a white noise disturbance term with mean 0 and variance \( \sigma^2 \). Also \( \beta_i, i = 1, \ldots, p \) and \( \gamma_j, j = 1, \ldots, q \) are parameters to be estimated.

This model can also be represented by a differential equation. For example, consider the ARMA(2,1) model

\[ y_m = \beta_0 + \beta_1 y_{m-1} + \beta_2 y_{m-2} + \phi_1 e_{m-1} + e_m, \]

(2.13)

This equation can be represented by the following differential equation:

\[ c_1 \frac{d^2 y}{dt^2}(t) + c_2 \frac{dy}{dt}(t) + y(t) = d_1 + d_2 \frac{de}{dt}(t) + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 - \frac{d_2}{\Delta t} \right) e(t), \]  

(2.14)

where \( t \) denotes the time corresponding to timepoint \( m \) and each \( c_i \) and \( d_i \) is a constant. One can approximate \( \frac{de}{dt}(t) \) with a backward difference approximation as was done with \( \frac{dy}{dt}(t) \) in Equation 2.6. From these and Equation 2.7, one finds that this differential equation can be approximated by

\[ c_1 \frac{y_m - 2y_{m-1} + y_{m-2}}{\Delta t^2} + c_2 \frac{y_m - y_{m-1}}{\Delta t} + y_m = d_1 + d_2 \frac{e_m - e_{m-1}}{\Delta t} \]

\[ + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 - \frac{d_2}{\Delta t} \right) e_m \]  

(2.15)

\[ \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) y_m - \left( \frac{2c_1}{\Delta t^2} + \frac{c_2}{\Delta t} \right) y_{m-1} + \frac{c_1}{\Delta t^2} y_{m-2} = d_1 - \frac{d_2}{\Delta t} e_{m-1} + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) e_m \]  

(2.16)

\[(c_1 + c_2 \Delta t + \Delta t^2) y_m - (2c_1 + c_2 \Delta t) y_{m-1} + c_1 y_{m-2} = d_1 - d_2 \Delta t e_{m-1} + (c_1 + c_2 \Delta t + \Delta t^2) e_m. \]  

(2.17)
Solving this equation for \( y_m \) yields
\[
y_m = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} + \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-1} - \frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-2} - \frac{d_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} e_{m-1} + e_m. \tag{2.18}
\]
Setting \( \beta_0 = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \beta_1 = \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \beta_2 = -\frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} \) and \( \phi_1 = -\frac{d_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} \) shows equivalence to the model given in Equation 2.13.

Fitting an ARMA\((p,q)\) model is a slightly longer process than fitting an AR model. One can determine \( p \) using the partial autocorrelation function of lag \( p \), as done in fitting an AR\((p)\) model. As for \( q \) one uses the same process as for choosing \( p \), except that one considers the correlation \( r_l^* \) between \( e_m \) and \( e_{m-l} \) instead, which is called the autocorrelation function of lag \( l \). Once \( p \) and \( q \) are chosen, there are several options for fitting a model, such as minimizing a least squares objective function \([18]\), using the Hannan-Rissanen algorithm \([43, 73]\), and maximum likelihood estimation \([12, 18]\). Typically maximum likelihood estimation is used, as one of the other algorithms is used to determine starting values for the parameters in a maximization of the likelihood. Using maximum likelihood estimation requires iterative techniques, such as the Levenberg-Marquardt algorithm \([73, 61, 68]\).

Another issue with AR and ARMA models is that there may be other factors influencing the value of the response variable. To overcome this, one may add explanatory variables to the model. If one adds an explanatory variable to an AR model, then one is said to be using an autoregressive exogenous variable (ARX) model. If added to an ARMA model, then one is said to be using an autoregressive moving average model with exogenous variables (ARMAX models).
2.2.3 ARX models

An ARX model with a single input has the form

\[ y_m = \beta_0 + \sum_{i=1}^{p} \beta_i y_{m-i} + \sum_{j=1}^{q} \gamma_j x_{m-j} + \epsilon_m \]  \hspace{1cm} (2.19)

where \( y_m \) is the value of the response variable \( Y \) at timepoint \( m \), \( x_m \) is the value of the independent variable \( X \) at timepoint \( m \) and \( \epsilon_m \) is a white noise disturbance term with mean 0 and variance \( \sigma^2 \). Also \( \beta_i, i = 1, \ldots, p \) and \( \gamma_j, j = 1, \ldots, q \) are parameters to be estimated.

Once again it can be shown that this model has a differential equation representation. Consider an ARX model with lag 2 and a single input of lag 2:

\[ y_m = \beta_0 + \beta_1 y_{m-1} + \beta_2 y_{m-2} + \gamma_1 x_{m-1} + \gamma_2 x_{m-2} + \epsilon_m \]  \hspace{1cm} (2.20)

Here the corresponding differential equation is

\[ c_1 \frac{d^2 y}{dt^2}(t) + c_2 \frac{dy}{dt}(t) + y(t) = d_1 + d_2 \frac{dx}{dt}(t) + d_3 x(t) + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) \epsilon(t), \]  \hspace{1cm} (2.21)

where \( t \) denotes the time corresponding to timepoint \( m \) and each \( c_i \) and \( d_i \) is a constant.

Here one can use the backward difference approximations for the derivatives of \( y(t) \) given in Equations 2.6 and 2.7, but this cannot be done for \( \frac{dx}{dt}(t) \), as a change in the input \( x_m \) does not result in an immediate change in the output \( y_m \). Thus \( x(t) \) is replaced with \( x_{m-1} \) rather than \( x_m \) and the backward difference approximation for \( \frac{dx}{dt}(t) \) is

\[ \frac{dx}{dt}(t) \approx \frac{x_{m-1} - x_{m-2}}{\Delta t}. \]  \hspace{1cm} (2.22)

Thus one can approximate Equation 2.21 with

\[ \frac{c_1 y_m - 2y_{m-1} + y_{m-2}}{\Delta t^2} + \frac{c_2 y_m - y_{m-1}}{\Delta t} + y_m = d_1 + d_2 \frac{x_{m-1} - x_{m-2}}{\Delta t} + \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) \epsilon_m \]  \hspace{1cm} (2.23)
\[
\left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) y_m - \left( \frac{2c_1}{\Delta t^2} + \frac{c_2}{\Delta t} \right) y_{m-1} + \frac{c_1}{\Delta t^2} y_{m-2} = d_1 + \left( \frac{d_2}{\Delta t} + d_3 \right) x_{m-1} - \frac{d_2}{\Delta t} x_{m-2} + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 \right) e_m \quad (2.24)
\]

\[
(c_1 + c_2 \Delta t + \Delta t^2) y_m - (2c_1 + c_2 \Delta t) y_{m-1} + c_1 y_{m-2} = d_1 \Delta t^2 + (d_2 \Delta t + d_3 \Delta t^2) x_{m-1} - d_2 \Delta t x_{m-2} + (c_1 + c_2 \Delta t + \Delta t^2) e_m \quad (2.25)
\]

\[
y_m = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} + \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-1} - \frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-2} + \frac{d_2 \Delta t + d_3 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} x_{m-1} - \frac{d_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} x_{m-2} + e_m \quad (2.26)
\]

Setting \( \beta_0 = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \beta_1 = \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \beta_2 = -\frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} \), \( \gamma_1 = \frac{d_2 \Delta t + d_3 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} \), and \( \gamma_2 = -\frac{d_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} \) will show equivalence.

Fitting an ARX model is similar to fitting an AR model, as one can use either the Yule-Walker equations or least squares once \( p \) has been determined. While this model can take into account external factors that influence the response variable, Nelles [73] reports that if the residuals from fitting an ARX model do not appear to be white noise, then the parameter estimates may be biased. While there exist techniques to fit an ARX model that deal with the issue of bias [73], one may choose to fit an ARMAX model to overcome this problem instead.

### 2.2.4 ARMAX models

An ARMAX model can be written as

\[
y_m = \beta_0 + \sum_{i=1}^{p} \beta_i y_{m-i} + \sum_{j=1}^{q} \gamma_j x_{m-j} + \sum_{k=1}^{r} \phi_k e_{m-k} + e_m \quad (2.27)
\]

where \( y_m \) is the value of the response variable \( Y \) at timepoint \( m \), \( x_m \) is the value of the independent variable \( X \) at timepoint \( m \), and \( e_m \) is a white noise disturbance term with
mean 0 and variance $\sigma^2$. Also $\beta_i, i = 1, \ldots, p, \gamma_j, j = 1, \ldots, q,$ and $\phi_k, k = 1, \ldots, r$ are parameters to be estimated.

To show that an ARMAX model can be represented with a differential equation, consider an ARMAX model with lag 2, a single input of lag 2 and an error term with lag 1:

$$y_m = \beta_0 + \beta_1 y_{m-1} + \beta_2 y_{m-2} + \gamma_1 x_{m-1} + \gamma_2 x_{m-2} + \phi_1 e_{m-1} + e_m$$  \hspace{1cm} (2.28)

This model is equivalent to the following second order plus lead time differential equation

$$c_1 \frac{d^2 y(t)}{dt^2} + c_2 \frac{dy(t)}{dt} + y(t) = d_1 + d_2 \frac{dx(t)}{dt} + d_3 x(t) + d_4 \frac{de(t)}{dt} + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 - \frac{d_4}{\Delta t} \right) e(t),$$  \hspace{1cm} (2.29)

where $t$ denotes the time corresponding to timepoint $m$ and each $c_i$ and $d_i$ is a constant. One can approximate the derivatives in this differential equation as before and solve for $y_m$. Setting $\beta_0 = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \beta_1 = \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2}, \beta_2 = -\frac{c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_1 = \frac{d_2 \Delta t + d_3 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_2 = -\frac{d_3 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2},$ and $\phi_1 = -\frac{d_4 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2}$ will show equivalence.

Determining parameter estimates for an ARMAX model is similar to that of an ARMA model. Nelles [73] outlines two other approaches for finding optimal parameters. For both approaches, an ARX model is first fit to determine starting parameters. In the first approach, a nonlinear optimization technique, such as the Levenberg-Marquardt algorithm, utilizes these starting parameters to determine the parameter estimates. In the other approach, which is also called extended least squares [62], the residuals from the ARX model are calculated and then the ARMAX model is fit using the ARX model residuals as estimates of $e_m$. Afterwards an ARX model is fit with the fitted ARMAX model residuals. These two models are refit using the other’s residuals until the parameter estimates converge. While this model deals with the possible problems of the residuals exhibiting nonconstant variance and outside factors that are influencing the response variable, it does
not allow for possible interactions between inputs. This can be overcome by using a NARMAX model, which is a nonlinear autoregressive moving average model with exogenous inputs.

### 2.2.5 NARMAX models

NARMAX models have rarely been used for glucose modeling. Ståhl and Johannsen\cite{99} use it in their comparison of models for glucose modeling. While a NARMAX model can have a very general form\cite{78}, here a NARMAX model with \( w \) inputs will be written as

\[
y_m = f(y_{m-1}, \ldots, y_{m-p}, x_{1,m-1}, \ldots, x_{1,m-q}, x_{2,m-1}, \ldots, x_{w,m-q}, e_{m-1}, \ldots, e_{m-r}) + e_m,
\]

(2.30)

where \( f \) is a function that is nonlinear in the inputs, but linear in the previous values of \( y \) and \( e_m \) is a white noise disturbance term with mean 0 and variance \( \sigma^2 \). While NARMAX models are nonlinear in the inputs, they are linear in the parameters. An example with 2 inputs and their interaction is

\[
y_m = \beta_0 + \beta_1 y_{m-1} + \beta_2 y_{m-2} + \gamma_{11} x_{1,m-1} + \gamma_{12} x_{1,m-2} + \gamma_{21} x_{2,m-1} + \gamma_{22} x_{2,m-2} + \delta_1 x_{1,m-1} x_{2,m-1} + \delta_2 x_{1,m-2} x_{2,m-2} + \phi_1 e_{m-1} + e_m
\]

(2.31)

A NARMAX model defined in this way can also be represented by a differential equation:

\[
c_1 \frac{d^2 y}{dt^2}(t) + c_2 \frac{dy}{dt}(t) + y(t) = d_1 + d_2 \frac{dx_1}{dt}(t) + d_3 x_1(t) + d_4 \frac{dx_2}{dt}(t) + d_5 x_2(t) + d_6 \frac{dx_1 x_2}{dt}(t) + d_7 x_1(t) x_2(t) + d_8 \frac{de}{dt}(t) + \left( \frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1 - \frac{d_8}{\Delta t} \right) e(t).
\]

(2.32)

The derivatives for the outputs and residuals may be approximated as done in previous derivations. While there are multiple inputs, \( x_i(t) \) is replaced in the differential equation with \( x_{i,m-1} \) since a change in the input cannot immediately impact the output, and \( \frac{dx}{dt}(t) \)
for $i = 1, 2$ are approximated as in Equation 2.22, i.e.

$$\frac{dx_i}{dt}(t) \approx \frac{x_{i,m-1} - x_{i,m-2}}{\Delta t}.$$  \hspace{1cm} (2.33)

With these approximations for the derivatives, one can solve for $y_t$. By setting $\beta_0 = \frac{d_1 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \beta_1 = \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2}, \beta_2 = -\frac{c_1 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_{11} = \frac{d_2 \Delta t + d_3 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_{12} = -\frac{d_2 \Delta t + d_3 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_{21} = \frac{d_4 \Delta t + d_5 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \gamma_{22} = -\frac{d_4 \Delta t + d_5 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \delta_1 = \frac{d_6 \Delta t + d_7 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2}, \delta_2 = -\frac{d_6 \Delta t + d_7 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2},$ and $\phi_1 = -\frac{d_8 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2},$ equivalence between the resulting equation and the NARMAX model is shown.

Like an ARMAX model, a NARMAX model must be fit via iterative methods due to the recursive definition of the residuals. While the NARMAX model compensates for many of the shortcomings of the other autoregressive models, it has weaknesses. Rollins et al. [86] note that if inputs are highly correlated, then the model matrix is ill-conditioned, which can cause large variance estimates for the parameters, and thus for predictions. Another issue is the possibility of overfitting.

### 2.2.6 Wiener networks

The Wiener network has a powerful structure for modeling nonlinear dynamic systems. A block diagram with $p$ inputs and one output is given in Figure 5.1. Each input $x_i$ is first passed through a dynamic linear block, denoted $g(x_i)$ and converted into its corresponding dynamic variable $v_i$. For this work the following second-order-plus-lead with dead time differential equation will be used:

$$\tau_i^2(t, x_i) \frac{d^2 v_i}{dt^2}(t) + 2\tau_i(t, x_i)\zeta_i(t, x_i) \frac{dv_i}{dt}(t) + v_i(t, x_i) = \tau_{ai}(t, x_i) \frac{dx_i}{dt}(t)x_i(t - \theta_i),$$  \hspace{1cm} (2.34)

where $\tau_i$ is a time constant, $\tau_{ai}$ is a lead parameter, $\zeta_i$ is a damping coefficient, and $\theta_i$ denotes dead time. For simplicity, assume that the dynamic parameters are time and
space invariant, i.e. $\tau_i(t, x_i) = \tau_i$, $\tau_{ai}(t, x_i) = \tau_{ai}$ and $\zeta_i(t, x_i) = \zeta_i$. Also $v_i(t, x_i)$ will be written as $v_i(t)$ henceforth. The vectors $\tau$, $\tau_a$, and $\zeta$ will denote all time constants, lead parameters and damping coefficients, respectively.

Since there does not exist a solution to this differential equation, a recursive definition for $v_i(t)$ must be found. To this end, a backward difference approximation to $\frac{dv_i}{dt}(t)$ will be used. Then, as done for Equations 2.6 and 2.7,

$$\frac{dv_i}{dt}(t) \approx \frac{v_i(t) - v_i(t - \Delta t)}{\Delta t}. \quad (2.35)$$

and

$$\frac{d^2v_i}{dt^2}(t) \approx \frac{v_i(t) - 2v_i(t - \Delta t) + v_i(t - 2\Delta t)}{\Delta t^2}. \quad (2.36)$$

Furthermore, as for ARMAX and NARMAX models, a step change in the input does not immediately affect the dynamic variable or the output. Thus, in determining $v_i(t)$,
$x_i(t)$ is replaced by $x_i(t - \Delta t)$. This substitution is also used in the backward difference approximation of $\frac{dx_i}{dt}(t)$. By substituting Equations 5.2 and 5.3 into Equation 5.1:

$$v_i(t) = \frac{2\tau^2 + 2\tau \zeta \Delta t}{\tau^2 + 2\tau \zeta \Delta t + \Delta t^2} v_i(t - \Delta t) - \frac{\tau^2}{\tau^2 + 2\tau \zeta \Delta t + \Delta t^2} v_i(t - 2\Delta t)$$

$$+ \frac{\Delta t(\tau_a + \Delta t)}{\tau^2 + 2\tau \zeta \Delta t + \Delta t^2} x_i(t - \theta_i - \Delta t) - \frac{\tau_a \Delta t}{\tau^2 + 2\tau \zeta \Delta t + \Delta t^2} x_i(t - \theta_i - 2\Delta t) \quad (2.37)$$

The resulting $v_i$'s are then passed through a static nonlinear block, denoted $f(v)$ in Figure 5.1. The function $f$ is typically a nonlinear function with respect to the inputs. For example, denoting $x = (x_1, \ldots, x_{11})$, Rollins et al. [86] use

$$f(x) = a_0 + \sum_{i=1}^{11} a_i x_i + \sum_{j=1}^{11} b_j x_j^2 + \sum_{k=1}^{10} \sum_{l=k+1}^{11} c_{k,l} x_k x_l. \quad (2.38)$$

Let $a$, $b$, and $c$ denote the vectors corresponding to all linear, quadratic and interaction parameters. They will be collectively referred to as the static parameters. Replacing the $x_i$ with $v_{ij}$ and denoting the fact that the $v_i$'s depend on $x_1$, the final resulting model is

$$f(t, X) = a_0 + \sum_{i=1}^{11} a_i v_i(t) + \sum_{j=1}^{11} b_j v_j^2(t) + \sum_{k=1}^{10} \sum_{l=k+1}^{11} c_{k,l} v_k(t) v_l(t) + \epsilon(t), \quad (2.39)$$

where $X = [x_1 \ x_2 \ \cdots \ x_{11}]$ and $\epsilon(t)$ is a normally distributed error term with mean 0 and variance $\sigma^2$. The residuals $\epsilon(t_1)$ and $\epsilon(t_2)$, $t_1 \neq t_2$ are also assumed to be independent of one another.

Since the errors are assumed to be independent and identically normally distributed with mean 0 and constant variance, typically a least squares objective function is minimized to find parameter estimates. This must be done with iterative methods due to the dynamic parameters. There are several advantages to using a Wiener network. One is that the dynamic parameters have phenomenological meaning. For example, the residence time of input $i$ can be estimated as $2\tau_i \zeta_i$. Second, its flexible structure allows it to model many different types of phenomena, including models where multiple outputs are needed. The
final advantage is that inputs are allowed to have different dynamics. If one chooses to use carbohydrates and protein consumption in modeling BGC, then given that the pathway for protein to be converted to glucose is much longer than that of carbohydrates, it makes sense for the residence time of protein to be greater than that of carbohydrates. One disadvantage is that the number of parameters can become very large depending on the number of inputs and the choice of function for the static nonlinear block.

### 2.2.7 Neural Networks

The last model to be introduced here, the neural network, has a different structure than the previous models. Like compartment models, they can handle any number of inputs or outputs. A neural network can be used for either regression or classification, but the focus here will be on regression. A figure of a neural network is given in Figure 2.2. Here we will follow the notation from Hastie et al. [47]. Given a set of inputs $X$, for $i = 1, \ldots, n$, one can define a neural network with $p$ inputs, $m$ hidden nodes and $q$ outputs. Let $Z_i$ denote the $i^{th}$ hidden node and $Y_j$ denote the $j^{th}$ output. Then

$$Z_1 = \sigma(\alpha_{10} + \alpha_{11}X_1 + \cdots + \alpha_{1p}X_p)$$
$$\vdots$$
$$Z_m = \sigma(\alpha_{m0} + \alpha_{m1}X_1 + \cdots + \alpha_{mp}X_p)$$

and

$$Y_1 = g_1(\beta_{10} + \beta_{11}Z_1 + \cdots + \beta_{1m}Z_m)$$
$$\vdots$$
$$Y_q = g_q(\beta_{q0} + \beta_{q1}Z_1 + \cdots + \beta_{qm}Z_m),$$

(2.40)
Figure 2.2 A graphical representation of a neural network with three inputs ($X_i$), two hidden nodes ($Z_i$) and three outputs ($Y_j$).

where $\sigma$ and $g_j$ for $j = 1, \ldots, q$ are functions. For $\sigma$, a popular choice is

$$\sigma(v) = (1 + e^{-v})^{-1}, \quad (2.42)$$

though any real-valued function can be used. As for $g_j$, the identity function is typically used if the neural network is used for regression. Here the $\alpha_i$’s and $\beta_j$’s are parameters to be estimated.

A neural network is typically fit using a least squares objective function and the back-propagation algorithm. This algorithm uses the chain rule and a steepest descent step to update the parameters. Typically this algorithm is not fit to convergence due to the possibility of overfitting.

Due to the flexibility in the choice of $\sigma$ and $g_j$, the neural network can be used as a model for a variety of situations. However, Rollins et al. [87] argue that an artificial neural
network can be used to find a model with a low error sum of squares on one subset of the data, but its performance on another subset of data may be poor due to extrapolation or changing dynamics.

Due to its advantages mentioned above and the disadvantages of other models, the Wiener network will be used for modeling BGC in this work. However, there are some difficulties that must be addressed. One is finding a methodology for fitting a Wiener network that avoids overfitting. While iterative methods can be used to find parameter estimates that yield a highly accurate model to the training data, typically this model will fit validation data poorly. Given this choice of function for the static nonlinear block, there are some parameters that are linear in the model. This is said to be a separable model, which allows for specialized algorithms to be used, such as those by Barham and Drane [4] and Golub and Pereyra [42]. However, these algorithms have also tended to badly overfit training data.

2.3 Forecasting Intervals

Another goal of this research is to be able to predict BGC at a given point in time in the future. Typically every model discussed thus far is used to predict BGC at the current time. However, several studies have been done to predict future BGC as well, and a wide variety of models have been used for this purpose. Sparacino et al. [96], Reifman et al. [83], and Gani et al. [36] use autoregressive models to predict BGC up to 90 minutes in the future. Eren-Oruklu et al. [25] utilize ARMA and ARMAX models to detect hypoglycemia 30 minutes into the future. Pappada et al. [77] use neural networks to predict BGC in type 1 diabetics up to 3 hours into the future. Rollins et al. [86] use a Wiener network in order to develop a model that predicts BGC in type 2 diabetics up to 90 minutes ahead of time. Such models could be referred to as $k$-steps ahead prediction (KSAP) models.
In econometrics, predicting the level of some variable in the future is called forecasting. Chatfield [16, 17] gives a fairly comprehensive review of current methods for calculating \((1 - \alpha)100\%\) prediction intervals when forecasting the value of the response variable at some time in the future. For this work, such an interval will be denoted a \((1 - \alpha)100\%\) forecast interval. The rest of this section is devoted to the various techniques used to calculate forecast intervals.

2.3.1 Intervals Based on Theory

In some cases, one can derive a formula for the variance of the response variable \(Y_t\), denoted \(\sigma^2\) for use in a forecast interval. This is the case for linear regression, and as will be shown, many of the autoregressive models described in Section 2.2. In this approach, a \((1 - \alpha)100\%\) forecast interval for the value of the response variable at \(k\) timesteps into the future can be calculated by

\[
\hat{y}_{t+k} \pm z_{\alpha/2} \hat{\sigma}_k \sqrt{1 + c_k} ,
\]

where \(\hat{\sigma}_k\) is the estimate of the square root of the variance of \(Y_{t+k}\) found from fitting the model and \(c_k\) is some constant such that \(\text{Var}(Y_{t+k} - \hat{y}_{t+k}) = \sigma_k^2(1 + c_k)\).

One model where \(\sigma_k^2\) can be determined is an AR(1) model. As given by Madsen et al. [64], the AR(1) model, written without an intercept, is \(y_t = \phi y_{t-1} + \epsilon_t\), where the residuals are assumed to be independent and identically distributed NOR(0, \(\sigma^2\)). For this model \(k = 1\), or a 1-step ahead prediction model, since \(y_t\) is one timestep ahead of \(y_{t-1}\). One can determine \(\text{Var}(Y_t)\) by first noting that, according to the AR(1) model, \(y_{t-1} = \phi y_{t-2} + \epsilon_{t-1}\),
\[ y_{t-2} = \phi y_{t-3} + \epsilon_{t-2}, \text{ etc.}, \] which implies that
\[
y_t = \phi(\phi y_{t-2} + \epsilon_{t-1}) + \epsilon_t \\
= \phi^2 y_{t-2} + \phi \epsilon_{t-1} + \epsilon_t \\
= \phi^2(\phi y_{t-3} + \epsilon_{t-2}) + \phi \epsilon_{t-1} + \epsilon_t \\
= \phi^3 y_{t-3} + \phi^2 \epsilon_{t-2} + \phi \epsilon_{t-1} + \epsilon_t \\
\vdots \\
= \sum_{i=0}^{\infty} \phi^i \epsilon_{t-i}. \quad (2.44)
\]

From this, one finds that
\[
\text{Var}(Y_t) = \sum_{i=0}^{\infty} \phi^2 \sigma^2 \\
= \frac{\sigma^2}{1 - \phi^2}. \quad (2.45)
\]

In this situation, \(\text{Var}(Y_t)\) can be estimated by replacing \(\phi\) and \(\sigma^2\) with \(\hat{\phi}\) and \(\hat{\sigma}^2\) found from fitting the model. One can show that for an ARX(1,\(z\)) model, where \(z\) is any integer and the exogenous variable is assumed to be measured without error, \(\text{Var}(Y_t)\) is the same as that for an AR(1) model.

An ARMA(1,1) model is another model where a formula for the variance can be derived. The derivation is similar as before, as the goal is to reduce the model to an infinite sum of
residuals. Again $k = 1$ and for convenience the intercept will be set to zero. This yields

\[
y_t = \alpha y_{t-1} + \phi \epsilon_{t-1} + \epsilon_t
\]

\[
= \alpha (\alpha y_{t-2} + \phi \epsilon_{t-2} + \epsilon_{t-1}) + \phi \epsilon_{t-1} + \epsilon_t
\]

\[
= \alpha^2 y_{t-2} + \alpha \phi \epsilon_{t-2} + (\alpha + \phi) \epsilon_{t-1} + \epsilon_t
\]

\[
= \alpha^2 (\alpha y_{t-3} + \phi \epsilon_{t-3} + \epsilon_{t-2}) + \alpha \phi \epsilon_{t-2} + (\alpha + \phi) \epsilon_{t-1} + \epsilon_t
\]

\[
= \alpha^3 y_{t-3} + \alpha^2 \phi \epsilon_{t-3} + \alpha (\alpha + \phi) \epsilon_{t-2} + (\alpha + \phi) \epsilon_{t-1} + \epsilon_t
\]

\[
\vdots
\]

\[
= \epsilon_t + \sum_{i=1}^{\infty} \alpha^{i-1}(\alpha + \phi) \epsilon_{t-i}
\]  

(2.46)

Now $\text{Var}(Y_t)$ is easy to calculate:

\[
\text{Var}(Y_t) = \sigma^2 + \sum_{i=1}^{\infty} \alpha^{2(i-1)}(\alpha + \phi)^2 \sigma^2
\]

\[
= \sigma^2 \left( 1 + \frac{(\alpha + \phi)^2}{1 - \alpha^2} \right).
\]  

(2.47)

Again parameter estimates $\hat{\sigma}^2, \hat{\alpha}$, and $\hat{\phi}$ are substituted into the equation for $\text{Var}(Y_t)$ to estimate it. As with the AR(1) and ARX(1,1) models, adding an exogenous input that is measured without error does not affect the variance. Hence $\text{Var}(Y_t)$ for an ARMAX model with one input, regardless of the lag on that input, is the same as $\text{Var}(Y_t)$ for an ARMA(1,1) model. Unfortunately this approach only works if the model assumptions appear to be valid, which does not always happen in practice. Furthermore, deriving the variance can be extremely difficult for other models.

2.3.2 Empirical Methods

If it appears that the residuals do not exhibit symmetry, then the forecast interval given in Equation 2.43 will exhibit less than nominal coverage. One method for dealing with this
issue is the use of empirical methods. The typical empirical method for the values $\pm z_{\alpha/2} \hat{\sigma}_e$ are replaced with the $100\alpha^{th}$ and the $100(1-\alpha)^{th}$ percentiles of the empirical distribution of the residuals found from fitting a model to a training set. This method requires very little computational power, but it may miss trends in the data, particularly if the data used in fitting the model is not a good representation of the true population. Chatfield [16] notes that two other approaches have been mentioned in the literature.

In the first approach, which is given in Gilchrist [39], the standard deviation of a KSAP model is estimated with the standard deviation of the $k$-steps ahead errors. Let the standard deviation of the errors from the $k$-steps ahead prediction model be denoted $s_k$. Thus the $(1-\alpha)100\%$ forecast interval in this approach is $\hat{y}_{t+k} \pm z_{\alpha/2}s_k$, where $z_{\alpha/2}$ is the $1-\alpha/2$ percentile of a standard normal distribution. If the number of samples used to fit the KSAP model is small, the $(1-\alpha/2) \cdot 100^{th}$ percentile of a $t$ distribution with $n-2$ degrees of freedom could be used in place of $z_{\alpha/2}$. Chatfield [16] reports that this method is unreliable for small $n$ and large $k$. This method could also perform poorly if the residuals are not a random sample from a symmetric distribution.

The second approach is given in Williams and Goodman [111]. In their approach, they use the first 24 observations to fit a model that predicts the number of business main telephones in service on the last day of a month in a city in Michigan, and then use the model to predict the number of business main telephones in service each month for 18 months into the future. After collecting the errors from this model, the first observation from the second set is added to the first set, and then a model is fit to the first 25 observations. An 18 month forecast is again calculated, and the errors are collected. This is repeated for the remainder of the dataset. From their work, the absolute values of the observed errors appeared to follow a gamma distribution as opposed to a standard normal distribution. The resulting prediction intervals from the gamma distribution appeared to have coverage
much closer to the \((1 - \alpha)100\%\) rate than those calculated from the residuals and the standard normal distribution. This approach may not work well if a closed form solution for the parameter estimates cannot easily be obtained, as iterative methods may cause this approach to require a large amount of computation. This is particularly true if the model of interest has many parameters, which requires a larger dataset to fit the initial model. For Williams and Goodman, their model only contained 14 parameters, and thus they could begin forecasting with a dataset with only 24 observations.

### 2.3.3 Bootstrap Methods

The bootstrap method is another option for constructing forecast intervals when the residuals appear to be asymmetric. One of the most widely used methods for calculating forecast intervals, this approach is used when the true distribution of the residuals is unknown or when there is only a small amount of data available for fitting a model. For this approach, one fits a model to obtain residuals \(\{e_1, e_2, \ldots e_n\}\), where \(n\) is the number of observations used to fit the data. A random sample \(\{e^*_1, e^*_2, \ldots e^*_n\}\) of these residuals is drawn with replacement where each residual is equally likely to be drawn. With many such samples, one can estimate the standard error of the residuals in order to perform inference. Two books that give a solid introduction to this topic are Efron and Tibshirani [24] and Hjorth [48].

Several studies on the bootstrap have been done in the literature pertaining to KSAP modeling. Veall [107] gives a fairly comprehensive review of computation methods for econometric models, including the bootstrap. Freedman and Peters [34] use the bootstrap to compute forecast intervals for the prices for capital, labor and energy in 1995 based on historical information from 1948-1971. Several papers that deal with the bootstrap in the presence of an autoregressive model include Findley [29] and Stine [97].
2.3.4 Forecasting Abnormal Glucose Levels

Such a forecast interval could be utilized to assess the risk of hyperglycemia or hypoglycemia before it occurs. This would allow one to take action that could completely prevent abnormal BGC. Some devices that currently offer to alarm the user of impending hyperglycemia or hypoglycemia include the Minimed Guardian® REAL-Time System (Medtronic Minimed, Northridge, CA) [9], the SEVEN® PLUS System (DexCom, Inc. San Diego, CA) [3] and the 5-day FreeStyle Navigator (Abbott Laboratories, Abbott Park, IL) [109]. Both the Minimed Guardian® REAL-Time System and the 5-day FreeStyle Navigator can alarm the user up to 30 minutes before a possible episode of abnormal BGC. Wolpert [112] and Mastrototaro et al. [69] warn that a current limitation of this technology is the possibility of too many false alarms. While those whose main concern is hypoglycemia may be tolerant of these false alarms, those who are not as concerned may turn off this feature, which could have dire consequences for the user should they become hypoglycemic.

Several studies have considered implementing an alarm that could warn the user of impending abnormal BGC. Bequette [7] reviews several techniques for implementing such an alarm. He reports that the most common method for predicting future episodes of hyperglycemia or hypoglycemia is what he calls a “linear projection.” In this linear projection, the simple linear regression model

\[
y(t) = \beta_0 + \beta_1 t + \epsilon,
\]

where \( y(t) \) is the observed BGC at time \( t \), the residuals \( \epsilon \) are independent and identically normally distributed with mean 0 and variance \( \sigma^2 \), and \( \beta_0 \) and \( \beta_1 \) are parameters to be estimated, is fit to a set of previous BGC measurements. From this model, one can use least squares theory to construct \( (1 - \alpha)100\% \) forecast intervals for BGC at an amount
of time into the future. If the predictions and/or the forecast intervals from the linear projection suggest that there is a high probability of hypoglycemia or hyperglycemia in the near future, then an alarm will warn the user of such a possible occurrence. Brauker [11] suggest a similar approach to forecasting BGC, while Noujaim et al. [75] investigate how accurate a CGMS must be in order to accurately institute such a method for forecasting BGC. One disadvantage of this approach is that an alarm may be sounded just before one’s BGC has stabilized on its own. This could occur when previous measurements of BGC indicate a quadratic relationship with time rather than a linear relationship. Another disadvantage of this approach is that it may not work well when one’s BGC has a large variance. This could cause an inaccurate forecast of BGC, which could either cause a false alarm to sound or an episode of future hypoglycemia or hyperglycemia to occur undetected. For example, if one attempts to forecast BGC soon after a meal, then the shape of the BGC data may be curved, as BGC typically rises soon after a meal and then falls. Fitting a regression to this will cause the slope of the fitted regression line to be grossly inaccurate. This causes the forecast to be inaccurate, and so the likelihood of a false alarm or the failure of an alarm to sound to warn the user of abnormal BGC increases.

There are other approaches being taken to forecast BGC. These methods have focused on hypoglycemia, as the risk of death is much greater for one who is hypoglycemic as opposed to one who is hyperglycemic. Eren-Oruklu et al. [25] use ARMA and ARMAX models to predict hypoglycemic episodes 30 minutes prior to their occurrence. Hughes et al. [50] develop two algorithms that alter the infusion rate of insulin when the risk of hypoglycemia is high. These algorithms also try to prevent rebound hyperglycemia, which can happen if the amount of blood insulin concentration becomes too low when one decreases it in order to treat hypoglycemia. It also warns the user so that they may take further action, such as ingesting some carbohydrates to prevent the hypoglycemic episode.
This would be of great use to type 1 diabetics, who could receive an alarm while sleeping to warn them of future hypoglycemia.

Another challenge in using the Wiener network that will be used in this work to predict BGC is the lack of statistical inference methods. Since a Wiener network is a dynamic non-linear model, even calculating a $(1 - \alpha)100\%$ prediction interval with the desired coverage is difficult. An added difficulty is that we would like to use this model to assist in predicting future BGC. The final goal of this work is to construct approximate $(1 - \alpha)100\%$ forecast intervals to be used to forecast future BGC. In order to do this, the Wiener network will be modified so that it can accurately predict BGC at a time point in the future. The model structure of this modified Wiener network will be exploited in order to construct the approximate forecast intervals. It is hoped that such a forecast interval could be used to predict future episodes of hypoglycemia or hyperglycemia.
CHAPTER 3. AN ALGORITHM FOR OPTIMALLY FITTING A WIENER MODEL

A paper to be submitted to Computers and Chemical Engineering

Lucas P. Beverlin, Derrick K. Rollins, Kaylee Kotz, Nisarg Vyas, David Andre, Greg Welk, and Warren Franke

Abstract

The purpose of this work is to present a new methodology for fitting Wiener networks to datasets with a large number of variables for use in a device that noninvasively monitors one’s blood glucose concentration. Wiener networks have the ability to model a wide range of data types and their structures can yield parameters with phenomenological meaning. There are several challenges to fitting such a model: model stiffness, the nonlinear nature of a Wiener network, possible overfitting and the large number of parameters inherent to large input sets. To overcome these challenges, a methodology for fitting Wiener networks is presented that utilizes its structure. In this methodology, dynamic parameters are first identified by utilizing an intelligent grid search, and the resulting dynamic variable is used in the estimation of static parameters. This methodology is used to fit Wiener networks to predict blood glucose concentration in 24 subjects. Models for five of these subjects compare favorably to Wiener networks fit manually to the same data using the Solver...
add-in for Microsoft Excel®.

3.1 Introduction

Wiener networks are widely used in modeling complex nonlinear systems. These networks have been used to model a wide range of data, such as gas concentrations [101], blood glucose concentrations [86], and pH levels [76], and their structure can yield parameters with phenomenological meaning [86]. In this work, Wiener networks are used to first convert inputs into their corresponding dynamic responses and then to pass these dynamic responses through a linear regression function to obtain the fitted output response. The parameters needed to convert the inputs into dynamic responses are referred to as dynamic parameters and the parameters of the linear regression function as static parameters.

The estimation of these parameters can be challenging, as the behavior of these networks can be highly nonlinear in the dynamic parameters. Since the Wiener network is a nonlinear model, there is typically no closed form solution for finding optimal parameter estimates. Hence iterative methods must be used to find parameter estimates. Several such iterative methods are given in Bates and Watts [6], Seber and Wild [90], Nelles [73] and Nocedal and Wright [74]. The number of parameters, which can be large, also increases rapidly as inputs are added. Given that Wiener networks utilize differential equations to convert an input into a dynamic response, stiffness, which is a situation where the derivative in a differential equation increases or decreases rapidly while the solution to the differential equation does not [26, 58], is another concern. This phenomenon causes an algorithm to take very small steps (i.e. progress slowly) in order to reach an optimal solution. Overfitting, a phenomenon where the model fits the data used to find the model well, but fits an independent dataset poorly, can also be a major issue.

The basic purpose of this article is to present a new methodology for fitting Wiener
networks to large input datasets in order to build subject-specific models in a device that noninvasively predicts blood glucose concentration (BGC) in people who are not insulin dependent that overcomes the challenges described in the previous paragraph. By fitting subsets of the parameters iteratively, we can deal with a large number of parameters and their nonlinearity, as numerical instability in the next iteration is less likely when fitting a subset of the parameters. During optimization, intelligent grid searches will be used to overcome stiffness. To avoid overfitting, we will utilize what is called “supervised learning” in the statistics literature[47, 51]. In supervised learning, the dataset is broken up into a training set, a validation set, and a test set. The model is fit to the training set with the validation set used to evaluate the fitted model against an independent dataset to guard against overfitting. The test set is scrutinized at the end of the optimization process to further evaluate if overfitting has occurred.

The proposed methodology is presented with the following outline. First a detailed description of the Wiener network with multiple inputs and a single output is given to establish the problem context. In Section 3 the details of the methodology are presented and an example is given to illustrate the algorithm in Section 4. Finally concluding remarks and some ideas for future work are given in the last section.

3.2 The Wiener Network

The Wiener network has a powerful structure for modeling nonlinear dynamic systems. A block diagram with \( p \) inputs and one output is given in Figure 5.1. Each input \( x \) is first passed through a dynamic linear block, denoted \( g(x_i) \) and converted into its corresponding dynamic variable \( v_i \). In Rollins et al. [86], the following second-order-plus-lead with dead
Figure 3.1 A graphical representation of the Wiener model used in Rollins et al. [86].

\[
\begin{align*}
    \tau_i^2(t, x_i) \frac{d^2 v_i}{dt^2}(t, x_i) + 2\tau_i(t, x_i)\zeta_i(t, x_i) \frac{dv_i}{dt}(t, x_i) + v_i(t, x_i) &= \tau_{ai}(t, x_i) \frac{dx_i}{dt}(t - \theta_i) + x_i(t - \theta_i), \\
    \end{align*}
\]

(3.1)

where \( \tau_i \) is a time constant, \( \zeta_i \) is a damping coefficient, \( \tau_{ai} \) is a lead coefficient, \( \theta_i \) denotes dead time and \( t \) is time. For simplicity, assume that the dynamic parameters are time and space invariant, i.e. \( \tau_i(t, x_i) = \tau_i \) and \( \zeta_i(t, x_i) = \zeta_i \). Also \( v_i(t, x_i) \) will be written as \( v_i(t) \) henceforth. The vectors \( \tau_o, \tau \) and \( \zeta \) will denote all lead coefficients, time constants and damping coefficients, respectively.

In order to calculate \( v_i(t) \), a recursive definition for \( v_i(t) \) must be found since there is no solution to this differential equation. This can be found by using a backward difference
approximation for the derivatives in Equation 5.1:

\[
\frac{dv_i}{dt}(t) \approx \frac{v_i(t) - v_i(t - \Delta t)}{\Delta t} \tag{3.2}
\]

and

\[
\frac{d^2v_i}{dt^2}(t) \approx \frac{v_i(t) - 2v_i(t - \Delta t) + v_i(t - 2\Delta t)}{\Delta t^2} \tag{3.3}
\]

It should be noted that since an instantaneous change in \(x_i\) at time \(t\) cannot result in an instantaneous change in \(v_i\) at time \(t\) under this structure, one must approximate \(x_i(t - \theta_i)\) and \(\frac{dx_i}{dt}(t)\) with \(x_i(t - \theta_i - \Delta t)\) and \(x_i(t - \theta_i - 2\Delta t)\):

\[
\frac{dx_i}{dt}(t) \approx \frac{x_i(t - \theta_i - \Delta t) - x_i(t - \theta_i - 2\Delta t)}{\Delta t}. \tag{3.4}
\]

By substituting Equations 5.2 - 5.4 into Equation 5.1:

\[
v_i(t) = \frac{2\tau_i^2}{\tau_i^2 + 2\tau_i\xi_i\Delta t + \Delta t^2} v_i(t - \Delta t) - \frac{\tau_i^2}{\tau_i^2 + 2\tau_i\xi_i\Delta t + \Delta t^2} v_i(t - 2\Delta t)
+ \frac{\tau_{ai}\Delta t + \Delta t^2}{\tau_i^2 + 2\tau_i\xi_i\Delta t + \Delta t^2} x_i(t - \Delta t - \theta_i) - \frac{\tau_{ai}\Delta t}{\tau_i^2 + 2\tau_i\xi_i\Delta t + \Delta t^2} x_i(t - 2\Delta t - \theta_i).
\]

The resulting \(v_i\)'s are then passed through a static nonlinear block, denoted \(f(v)\) in Figure 5.1. The function \(f\) is typically a nonlinear function with respect to the \(v_i\)'s. For example, Rollins et al. [86] uses 11 inputs and the following function for the static nonlinear block:

\[
f(v_1(t), \ldots, v_{11}(t)) = a_0 + \sum_{i=1}^{11} a_i v_i(t) + \sum_{j=1}^{11} b_j v_j^2(t) + \sum_{k=1}^{10} \sum_{l=k+1}^{11} c_{k,l} v_k(t) v_l(t). \tag{3.6}
\]

In this work, the quadratic and second-order interaction terms will be omitted, in order to reduce the parameterization of the model. Three of the inputs used by Rollins et al. were also removed to further reduce parameterization. Thus the static block is represented by
a linear regression model. Hence the final resulting model is
\[
y(t) = a_0 + \sum_{i=1}^{8} a_i v_i(t) + \epsilon(t).
\] (3.7)

where \( \epsilon(t) \) is a normally distributed error term with mean 0 and variance \( \sigma^2 \). These error terms are assumed to be independent of one another. Finally let \( \mathbf{a} \) denote the vector corresponding to all linear static parameters. They will be collectively referred to as the static parameters.

### 3.3 The Parameter Estimation Algorithm

In this section the featured algorithm to fit the Wiener network given in the previous section will be described. Following Rollins et al. [86], the objective of this modeling problem is to maximize the true but unknown correlation coefficient \( \rho_{y,\hat{y}} \) between the measured and fitted BGC, which is estimated by
\[
r_{\text{fit}} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}},
\] (3.8)

where \( n \) is the number of observations used in fitting the model. More specifically, under this objective a model is declared **useful** if and only if
\[
\rho_{y,\hat{y}} > 0.
\] (3.9)

The meaning of this criterion is that predictions of BGC from the model decrease and increase with measured BGC beyond some degree of mere chance, i.e. there is true positive correlation. Notwithstanding, the closer this value is to the upper limit of 1, the more useful the model. Therefore, to achieve this objective, one seeks to identify a model with a sufficiently large value of \( r_{\text{fit}} \). To this end, the data are separated into three sets: a training set, a validation set, and a test set. The training set is used to build the model and the
validation set is used to evaluate the model against data that are not directly used by the optimization process to estimate the model parameters. The test set is used to ensure that the model accurately predicts BGC on an independent data set.

While $r_{\text{fit}}$ could be used as a function of the parameters to be maximized, its highly complex mapping of the parameters into the response space of $r_{\text{fit}}$ makes this impractical. Thus in order to identify a model with as large a value of $r_{\text{fit}}$ as possible, one might minimize the error sum of squares (SSE) with respect to the parameters instead, i.e.

$$\text{Maximize } r_{\text{fit}} \text{ by minimizing } \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Subject to: $\zeta_i > 0, \tau_i > 0, \theta_i \geq 0 \quad \forall i,$

where $\Theta$ is a vector representing the estimated dynamic and static parameters $\tau, \zeta, \tau_a, \theta, a$ and $n$ is the number of observations in the training set. For simple linear regression, minimizing SSE will yield the same parameter estimates as maximizing $r_{\text{fit}}$. While this is not always guaranteed in nonlinear regression, experimental evidence supports the notion that as SSE decreases, $r_{\text{fit}}$ increases. [86] A major advantage of minimizing SSE is that there exist many algorithms for minimizing it. Specialized algorithms for the minimization of SSE include the Gauss-Newton algorithm [45, 65] and the Levenberg-Marquardt algorithm [61, 68, 65]. Other algorithms used for this purpose include the BFGS algorithm [13, 32, 41, 94] and the conjugate gradient algorithm [74]. While these algorithms will succeed in finding a model with a large $r_{\text{fit}}$ on the training set if allowed to run to convergence, this typically results in a badly overfit model, in that the value of $r_{\text{fit}}$ found for this model on a test set is very low. To combat overfitting, supervised learning can be used in conjunction with these methods. However, this was found to result in a poor fit for training, validation and test sets if all parameters are used simultaneously. Model fitting was found to be slow if subsets of the parameters were fit during supervised learning. If this algorithm is to be
used in a device that will build a model of one’s BGC to be used on data not used to build the model, a different approach must be sought.

While determining a model whose $r_{\text{fit}}$ is as large as possible should result in finding a useful model, the model could still exhibit significant bias. Significant bias could cause the predictions of BGC to indicate normal BGC levels when in fact they may be dangerously high or low. Thus a secondary criterion on the absolute difference between $y_i$ and $\hat{y}_i$ is used for model assessment. This measure of accuracy, denoted the average absolute error (AAE), is

$$\text{AAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n},$$

where $n$ is the number of measurements used in the calculation of this statistic. As AAE decreases, accuracy is judged to increase. In the presence of model bias, $r_{\text{fit}}$ can be large despite a large AAE. Thus eliminating bias from the model improves the accuracy of the model.

Thus, in addition to a sufficiently large $r_{\text{fit}}$ value for the training set and a sufficiently large correlation on the validation set, denoted $r_{\text{val}}$, an acceptable model must also have a relatively small value of AAE in training. Achieving a relatively small value of AAE in validation is not necessary if a large $r_{\text{fit}}$ value is observed from the model on both sets. If a model achieves a high $r_{\text{fit}}$ as demonstrated in training then high accuracy in the validation set can be obtained with effective feedback correction or feedback control to reduce or eliminate bias.

For simplicity, we will assume that $\theta_i$ is fixed for every $i$ and that $\tau_{ai} = 0$. From experience, the improvement in model performance that can be achieved by including $\theta_i$ and $\tau_{ai}$ as estimable parameters appears to be small in comparison to the increase of the algorithm’s speed with 22 fewer parameters to estimate. Finally, in order to maintain stability in the differential equation, $\tau_i$ and $\zeta_i$ must be greater than 0. For more details on
these dynamic parameters, see Seborg et al. [91]

Now we present the algorithm, which will fit a Wiener network model to each subject. The main steps of the algorithm are given, with a detailed explanation of what happens in the algorithm given below it. Pseudocode for this algorithm is given in the Appendix. Let \( y \) denote the observed BGC in the dataset, \( y^{Tr} \) the observed BGC in the training set and \( y^{Va} \) the observed BGC in the validation set. These superscripts will be used to denote training and validation sets throughout this section.

1. Split the data into a training set, a validation set and the test set, which will consist of the rest of the data, to avoid overfitting.

The data is then split into a training set, which will consist of the first 37.5% of the data, a validation set, which will consist of the next 37.5% of the data, and the test set, which will consist of the rest of the data, to avoid overfitting.

2. Calculate \( \bar{y}^{Tr} = \frac{1}{n} \sum_{i=1}^{n} y_i^{Tr} \), where \( n \) is the number of observations in the training set. The model where \( \hat{y}_i = \bar{y}^{Tr} \) for every \( i \) in the dataset will be used as the baseline model.

Here one is fitting the model \( y = a_0 + \epsilon \) to the training data. This is done to create a baseline model with a low AAE when no dynamic variables are used to predict BGC. Thus let \( \bar{y}^{Tr} = y^{Va} = \bar{y}^{Tr} \).

3. Determine the best dynamic parameters for the first input, which is carbohydrates.

This is done by choosing dynamic parameters \( \hat{\tau}_1, \hat{\zeta}_1 \) such that the predictions \( \hat{y} = \bar{y}^{Tr} + v_1(\hat{\tau}_1, \hat{\zeta}_1) \) are as highly correlated with the observed BGC as possible.

Two grid searches are done to determine the dynamic parameter estimates \( \hat{\tau}_1 \) and \( \hat{\zeta}_1 \). For the first grid search, the bounds were chosen from experience in fitting Wiener networks.
to other subjects’ BGC. See Algorithm 3 for the bounds used for model fitting in this work. The first grid search finds $\tau^*_1$ and $\zeta^*_1$ that satisfy three conditions. First, the correlation $r_{Tr}$ between $y^{Tr}$ and $\tilde{y}^{Tr} + v_{1}(\tau^*_1, \zeta^*_1)^{Tr}$, where $v_{1}(\tau^*_1, \zeta^*_1)^{Tr}$ is the vector of observations of the first dynamic variable in the training set, must be greater than 0. Second, the correlation $r_{Va}$ between $y^{Va}$ and $\tilde{y}^{Va} + v_{1}(\tau^*_1, \zeta^*_1)^{Va}$, where $v_{1}(\tau^*_1, \zeta^*_1)^{Va}$ is the vector of observations of the first dynamic variable in the validation set, must also be greater than 0. Finally, $r_{Tr}$ and $r_{Va}$ are maximized such that the absolute difference between $r_{Va}$ and $r_{Tr}$ is minimized. This is desired because one hopes for uniform model performance on the training, validation and test sets. While model performance on the test set cannot be assessed until the model fitting is complete, model performance on the training and validation sets can be assessed. To satisfy the three criteria, the algorithm determines the 25 pairs of dynamic parameter estimates that maximize $r_{Va}$. From these, the dynamic parameter estimates that yield the smallest value of $|r_{Tr} - r_{Va}|$ are chosen. Once this first grid search is completed, a second finer grid search is performed whose center is $(\tilde{\tau}_i, \tilde{\zeta}_i)$, where the goals for $r_{Tr}$ and $r_{Va}$ are the same as the first grid search. We denote the dynamic parameters found from the second grid search $\tilde{\tau}_1$ and $\tilde{\zeta}_1$, the correlation between observed and predicted BGC in the training set $r_{Tr}^*$ and the same correlation in the validation set $r_{Va}^*$.

4. Set $\hat{y}^1 = \hat{a}_0 + \hat{a}_1 v_{1}(\tilde{\tau}_1, \tilde{\zeta}_1)$, where $\hat{a}_0$ and $\hat{a}_1$ are the parameter estimates of $a_0$ and $a_1$ found from least squares.

The simple linear regression model $y = a_0 + a_1 v_{1}(\tilde{\tau}_1, \tilde{\zeta}_1) + \epsilon$ is fit in order to minimize AAE. Since $v_{1}(\tilde{\tau}_1, \tilde{\zeta}_1)$ is the only regressor, the correlations $r_{Tr}^*$ and $r_{Va}^*$ will remain the same regardless of the value of $\hat{a}_0$ and $\hat{a}_1$ found from this fit. Set $\hat{y}^1 = \hat{a}_0 + \hat{a}_1 v_{1}(\tilde{\tau}_1, \tilde{\zeta}_1)$, where the superscript 1 on $\hat{y}$ denotes that one input has been fit.

5. Determine the best dynamic parameters for the second input, which is fat. This is
done by choosing dynamic parameters $\hat{\tau}_2, \hat{\zeta}_2$ and $s_2$ such that the predictions $\hat{y} = \hat{y}^1 + s_2v_2(\hat{\tau}_2, \hat{\zeta}_2)$ are as highly correlated with the observed BGC as possible.

The biggest difference between steps 3 and 5 is the introduction of the a constant $s_2$. This constant $s_2$ is chosen to be one of $\pm c_2$, whichever yields the greater correlation $r_{Va}$ between $y^{Va}$ and $\hat{y}^{1, Va} + s_2v_2(\tau^*_2, \zeta^*_2)^{Va}$, where $c_2$ is based on the what the perceived maximum value of the resulting dynamic variable is. For this work, see Algorithm 3 for the value of $c_i$ used for each $i$. Otherwise a similar process is used to identify dynamic and static parameters for each of the other inputs. For input 2, a grid search is done first in order to identify dynamic parameter estimates and the corresponding static parameter, denoted $\tau^*_2, \zeta^*_2$ and $s_2$, respectively. Here the parameters must satisfy three criteria. First is that the correlation $r_{Tr}$ between $y^{1, Tr}$ and $\hat{y}^{Tr} + s_2v_2(\tau^*_2, \zeta^*_2)^{Tr}$ is greater than $r^*_{Tr}$. Second it is desired that $r_{Va} \geq r^*_{Va}$. Finally the correlation $r_{Va}$ is maximized such that $|r_{Tr} - r_{Va}|$ is minimized, as done in step 3. If such parameters cannot be found in the first grid search, then take $\hat{\tau}_2, \hat{\zeta}_2$, and $s_2$ such that the correlation $r_{Va}$ is maximized. Otherwise a second grid search is done, centered at $(\tau^*_2, \zeta^*_2)$, and the same criteria are used to determine the dynamic parameter estimates $\hat{\tau}_2$ and $\hat{\zeta}_2$ and a first static parameter estimate $s_2$. Let $r_{Tr}$ denote the correlation between observed and predicted BGC in the training set and $r_{Va}$ denote the correlation between observed and predicted BGC in the validation set from this second grid search.

6. Determine $\hat{a}_0, \hat{a}_1$, and $\hat{a}_2$, the static parameter estimates when the first two dynamic variables are present in the model.

Once the grid search is complete, the multiple linear regression model $y = a_0 + a_1v_1(\hat{\tau}_1, \hat{\zeta}_1) + a_2v_1(\hat{\tau}_2, \hat{\zeta}_2) + \epsilon$ is fit and the correlation between the observed and predicted BGC in the training set, denoted $r^*$, and the validation set, denoted $r^{**}$, are calculated.
Table 3.1  Bounds and $s_i$ used for the first grid search.

<table>
<thead>
<tr>
<th>Input</th>
<th>$s_i$</th>
<th>LB</th>
<th>UB</th>
<th>Input</th>
<th>$s_i$</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>60</td>
<td>5</td>
<td>.005</td>
<td>60</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>150</td>
<td>600</td>
<td>6</td>
<td>2</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1400</td>
<td>1900</td>
<td>7</td>
<td>2</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>.0025</td>
<td>10</td>
<td>60</td>
<td>8</td>
<td>.0005</td>
<td>70</td>
<td>120</td>
</tr>
</tbody>
</table>

If $r^* < r_{Tr}$ or $r^{**} < r_{Va}$, then set $\hat{a}_0$ and $\hat{a}_1$ to their previous values, found in step 4, and set $\hat{a}_2 = s_2$. If the second grid search was not performed and $r^* < r_{Tr}$ or $r^{**} < r_{Va}$, then the only change to this protocol is to set $\hat{a}_2 = 0$ instead of $s_2$. Otherwise if $r^* \geq r_{Tr}$ and $r^{**} \geq r_{Va}$, then the static parameter estimates are set to those found from fitting the multiple linear regression model. Finally set $y^2 = \hat{a}_0 + \hat{a}_1 v_1(\hat{\tau}_1, \hat{\zeta}_1) + \hat{a}_2 v_2(\hat{\tau}_2, \hat{\zeta}_2)$.

7. Repeat steps 5-6 for each of the other 6 inputs, where only the bounds of the grid searches and $s_i$ differ for each input.

8. Use a backward stepwise regression to possibly remove regressors from the model to further improve correlation.

Once the static parameters are determined for all inputs, a backward stepwise regression is performed. For this backward stepwise regression, the statistic used to determine if a dynamic variable should be removed from the model is $r_{val}$. For each dynamic variable $v_i$ that is currently in the model, i.e. for each dynamic variable such that $\hat{a}_i \neq 0$, two models are fit. One is where all model parameter estimates are held fixed, except that $\hat{a}_i$ is set to 0. For the other, the dynamic parameter estimates are held constant, but a linear regression model with every dynamic variable still in the model other than dynamic variable $v_i$ is fit. Once this is done for each input, the model with the largest $r_{val}$ is chosen. If this value of
$r_{\text{val}}$ is larger than the value of $r_{\text{val}}$ from the previous steps and the value of $r_{\text{fit}}$ is larger than $r_{\text{fit}}$ from the previous model, then the model parameters are updated. Otherwise the backward stepwise regression is complete. This is repeated until the removal of an input does not result in an increase in $r_{\text{val}}$ from the previous model.

3.4 Modeling Blood Glucose Concentrations of Non-Insulin Dependent Type 2 Diabetics

We now illustrate our methodology and compare it to other models fit to the same data. In this study, 24 type 2 diabetics who exhibit significant variation in their BGC participated in a study in order to determine if their BGC can be accurately predicted from a Wiener network using activity variables, food consumption and time of day. Since type 2 diabetes affects each subject differently, a model was fitted for each individual in the study.

In order to obtain frequent measurements of BGC, the Medtronic MiniMed Continuous Glucose Monitoring CGMS® System GoldTM (Medtronic Minimed, Northridge, CA) was used. The SenseWear® Pro3 Body Monitoring System (BodyMedia, Inc., Pittsburgh, PA) was used to measure the activity variables used in building this model. From these devices measurements of activity and BGC were obtained every five minutes. Subjects were also asked to record the food that they ingested during this time with a PDA, which used Weightmania® Pro software (Edward A. Greenwood, Inc., Cambridge, MA). Other than the necessary downtime to download the data from these devices, data were collected by these devices twenty-four hours a day for four weeks. While the SenseWear® Pro3 Body Monitoring System can measure over 30 activity variables, Rollins et al. [86] only uses seven of these, and for this work we will only use three. Of the other four variables, three of them, carbohydrates, fat, and protein, are food variables that represent the amount of
Table 3.2 A table of inputs used in the modeling work of Rollins et al. [86] Those in bold will be used in this work, and the number in parentheses denotes the input number, i.e. \( v_1 \) corresponds to carbohydrates.

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Transverse accel.-peaks (4)</td>
</tr>
<tr>
<td></td>
<td>Near body temp.</td>
</tr>
<tr>
<td></td>
<td>Longitudinal accel.-average</td>
</tr>
<tr>
<td></td>
<td>Transverse accel. - MAD (6)</td>
</tr>
<tr>
<td>Food</td>
<td>Carbohydrates (1)</td>
</tr>
<tr>
<td></td>
<td>Protein (3)</td>
</tr>
<tr>
<td>Circadian</td>
<td>Time of day (8)</td>
</tr>
</tbody>
</table>

Each consumed in grams over every five minutes. The final one, time of day, allows one to capture the circadian rhythm of each individual’s body, which has been shown to have an effect on one’s BGC [106]. It assumes values from 0, denoting midnight, to 1439, denoting 11:59 pm. A table of all inputs is given in Table 5.1.

While the AAE is used to ensure that an accurate model is obtained for each subject, the determination that a low AAE has been achieved must be evaluated on a per subject basis. This is because the range of BGC varies from person to person. Thus to adjust for this, the relative AAE (RAAE) is used to scale AAE. To calculate RAAE, first the estimated standard deviation \( \hat{\sigma}_{Y} \) of the BGC in the set of interest must be calculated:

\[
\hat{\sigma}_{Y} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2},
\]  

(3.12)

where \( n \) is the number of observations in the data set of interest. Now the RAAE can be calculated as

\[
\text{RAAE} = \frac{\text{AAE}}{\hat{\sigma}}.
\]  

(3.13)

The value of RAAE for a One Touch Ultra\textsuperscript{®} blood glucose meter (LifeScan, Inc., Milpitas,
CA) using replicated BGC measurements was found to be 0.59. Thus a model whose RAAE is near 0.6 for each subject is desirable.

The results of the Wiener networks fit to these 24 subjects are given in Table 3.3. Given the values of $r_{\text{val}}$ in the table, we will consider a model to be excellent if the correlation $r_{\text{Test}}$ between observed and predicted BGC is at least 0.5, very good if at least 0.4, good if at least 0.3, fair if at least 0.2, and poor if less than 0.2. One can see that in training and validation, the correlation between observed and predicted BGC is above 0.3 for each subject, while in the test set this correlation is deemed to be good or better for 14 of the 20 subjects and fair or better for 19 of the 20 subjects. While the RAAE is typically greater than 0.6, it is still low, indicating that these fitted models predict BGC fairly accurately. For comparison, models for subjects 1-5 were fit manually using the Solver add-in for Microsoft Excel®. For these fits, the static parameters were allowed to run to convergence. Results for these manual fits and the fits from the presented methodology are presented in Table 3.4. The results found from manually fitting the Wiener networks and this algorithm are comparable.

### 3.5 Concluding Remarks

This article presents a methodology that can find accurate parameter estimates for Wiener networks that predict frequent BGC measurements in a short period of time. This methodology uses the correlation between observed and predicted BGC in training and validation sets as well as the AAE in the training set to produce an accurate model while avoiding overfitting. It uses an intelligent grid search to estimate the dynamic parameters and fits a linear regression after the estimation of each input’s dynamic parameters in an attempt to improve the correlation between observed and predicted BGC from the Wiener network. This work indicates that the Wiener network shows promise in modeling
Table 3.3  Results from fitting four weeks of data from four subjects. Here the first 1.5 weeks of the data was used in training, the next 1.5 weeks was used in validation, and the remainder in the test set.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Training $r_{\text{fit}}$</th>
<th>RAAE</th>
<th>Validation $r_{\text{val}}$</th>
<th>RAAE</th>
<th>Test $r_{\text{test}}$</th>
<th>RAAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.303</td>
<td>0.731</td>
<td>0.532</td>
<td>0.837</td>
<td>0.513</td>
<td>0.816</td>
</tr>
<tr>
<td>2</td>
<td>0.317</td>
<td>0.737</td>
<td>0.504</td>
<td>0.660</td>
<td>0.513</td>
<td>0.714</td>
</tr>
<tr>
<td>3</td>
<td>0.375</td>
<td>0.703</td>
<td>0.476</td>
<td>0.666</td>
<td>0.468</td>
<td>0.660</td>
</tr>
<tr>
<td>4</td>
<td>0.542</td>
<td>0.594</td>
<td>0.524</td>
<td>0.626</td>
<td>0.444</td>
<td>0.702</td>
</tr>
<tr>
<td>5</td>
<td>0.651</td>
<td>0.594</td>
<td>0.677</td>
<td>0.569</td>
<td>0.321</td>
<td>0.779</td>
</tr>
<tr>
<td>6</td>
<td>0.494</td>
<td>0.670</td>
<td>0.642</td>
<td>0.591</td>
<td>0.551</td>
<td>0.646</td>
</tr>
<tr>
<td>7</td>
<td>0.391</td>
<td>0.719</td>
<td>0.426</td>
<td>0.764</td>
<td>0.429</td>
<td>0.684</td>
</tr>
<tr>
<td>8</td>
<td>0.471</td>
<td>0.672</td>
<td>0.399</td>
<td>0.788</td>
<td>0.353</td>
<td>0.824</td>
</tr>
<tr>
<td>9</td>
<td>0.504</td>
<td>0.649</td>
<td>0.496</td>
<td>0.692</td>
<td>0.402</td>
<td>0.724</td>
</tr>
<tr>
<td>10</td>
<td>0.330</td>
<td>0.686</td>
<td>0.347</td>
<td>0.782</td>
<td>0.333</td>
<td>0.721</td>
</tr>
<tr>
<td>11</td>
<td>0.490</td>
<td>0.670</td>
<td>0.369</td>
<td>0.824</td>
<td>0.002</td>
<td>0.962</td>
</tr>
<tr>
<td>12</td>
<td>0.356</td>
<td>0.732</td>
<td>0.281</td>
<td>0.732</td>
<td>0.327</td>
<td>0.757</td>
</tr>
<tr>
<td>13</td>
<td>0.480</td>
<td>0.658</td>
<td>0.549</td>
<td>0.665</td>
<td>0.393</td>
<td>0.764</td>
</tr>
<tr>
<td>14</td>
<td>0.443</td>
<td>0.656</td>
<td>0.452</td>
<td>0.663</td>
<td>0.324</td>
<td>0.739</td>
</tr>
<tr>
<td>15</td>
<td>0.449</td>
<td>0.965</td>
<td>0.432</td>
<td>0.854</td>
<td>0.509</td>
<td>1.011</td>
</tr>
<tr>
<td>16</td>
<td>0.352</td>
<td>0.742</td>
<td>0.393</td>
<td>0.911</td>
<td>0.291</td>
<td>0.742</td>
</tr>
<tr>
<td>17</td>
<td>0.444</td>
<td>0.539</td>
<td>0.729</td>
<td>0.569</td>
<td>0.220</td>
<td>1.090</td>
</tr>
<tr>
<td>18</td>
<td>0.507</td>
<td>0.621</td>
<td>0.601</td>
<td>0.613</td>
<td>0.245</td>
<td>0.749</td>
</tr>
<tr>
<td>19</td>
<td>0.315</td>
<td>0.727</td>
<td>0.432</td>
<td>0.667</td>
<td>0.299</td>
<td>0.822</td>
</tr>
<tr>
<td>20</td>
<td>0.492</td>
<td>0.666</td>
<td>0.484</td>
<td>0.765</td>
<td>0.222</td>
<td>0.916</td>
</tr>
<tr>
<td>Mean</td>
<td>0.436</td>
<td>0.687</td>
<td>0.487</td>
<td>0.712</td>
<td>0.358</td>
<td>0.791</td>
</tr>
<tr>
<td>StDev</td>
<td>0.091</td>
<td>0.085</td>
<td>0.113</td>
<td>0.100</td>
<td>0.131</td>
<td>0.119</td>
</tr>
</tbody>
</table>
Table 3.4  A comparison of model fits. Auto denotes the algorithm presented in this paper was used to find the model while Manual denotes that Wiener networks fit manually via the Solver add-in for Microsoft Excel®.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Algorithm</th>
<th>Training $r_{fit}$</th>
<th>Training RAAE</th>
<th>Validation $r_{val}$</th>
<th>Validation RAAE</th>
<th>Test $r_{Test}$</th>
<th>Test RAAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Manual</td>
<td>0.42</td>
<td>0.70</td>
<td>0.40</td>
<td>0.81</td>
<td>0.69</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>0.30</td>
<td>0.73</td>
<td>0.53</td>
<td>0.84</td>
<td>0.51</td>
<td>0.82</td>
</tr>
<tr>
<td>2</td>
<td>Manual</td>
<td>0.30</td>
<td>0.75</td>
<td>0.43</td>
<td>0.72</td>
<td>0.59</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>0.32</td>
<td>0.74</td>
<td>0.50</td>
<td>0.66</td>
<td>0.51</td>
<td>0.71</td>
</tr>
<tr>
<td>3</td>
<td>Manual</td>
<td>0.33</td>
<td>0.70</td>
<td>0.36</td>
<td>0.76</td>
<td>0.51</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>0.38</td>
<td>0.70</td>
<td>0.48</td>
<td>0.67</td>
<td>0.47</td>
<td>0.66</td>
</tr>
<tr>
<td>4</td>
<td>Manual</td>
<td>0.53</td>
<td>0.61</td>
<td>0.49</td>
<td>0.67</td>
<td>0.43</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>0.54</td>
<td>0.59</td>
<td>0.52</td>
<td>0.63</td>
<td>0.44</td>
<td>0.70</td>
</tr>
<tr>
<td>5</td>
<td>Manual</td>
<td>0.58</td>
<td>0.62</td>
<td>0.57</td>
<td>0.62</td>
<td>0.42</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>Auto</td>
<td>0.65</td>
<td>0.59</td>
<td>0.68</td>
<td>0.57</td>
<td>0.32</td>
<td>0.78</td>
</tr>
</tbody>
</table>

continuous-time BGC in non-insulin dependent people.

For this algorithm to be used in a device that noninvasively predicts BGC for a non-insulin dependent person, a few adjustments will need to be made. Firstly, the data used here measured carbohydrates, fat and protein measured to the nearest tenth of a gram for each meal. Any device that noninvasively measures BGC must have a simple way to enter food quantities. Manually measuring food quantities is time consuming, and thus those who wear the device will likely not properly estimate their food intake. This will reduce the accuracy of the model fitted by the device.

A second issue is the calibration of the device. In this work, BGC was measured every five minutes. For someone who does not typically use a CGMS, such frequent BGC measurements will be unavailable. These frequent measurements are also not possible using a glucose meter over a period of several days due to discomfort from constant finger
pricking. Thus a novel scheme must be devised to automatically monitor BGC under infrequent data collection.

A final challenge to consider is that the parameters are not time and space invariant. For a type 2 diabetic, the pancreas’ ability to produce insulin changes over time, and the body’s insulin sensitivity also changes over time. Thus the rate at which the body converts glucose to energy changes over time. This implies that the model found from this methodology to predict one’s BGC today may not accurately predict BGC a year from now. While this methodology allows one to determine a model to accurately predict BGC quickly, there may be a way to predict changes in the parameters over time.

If these challenges can be overcome, then the development of a device that can noninvasively predict BGC in non-insulin dependent humans could become reality. This device would eliminate the need for a catheter, which is used by most continuous glucose monitoring systems on the market, and thereby eliminate much of the discomfort experienced by those who use a CGMS regularly. While this device will be useful for those with non-insulin dependent type 2 diabetes, anyone who is not insulin dependent and wishes to monitor their BGC could benefit from such a device.

3.6 Acknowledgments

This work was supported under the National Institute of Health through the Illinois Institute of Technology under Grant SA395-1125-6520. The authors would like to thank BodyMedia for funding much of the data collection and allowing us to use their equipment and Jeanne Stewart for assisting in data collection.
Appendix

Pseudocode for the proposed algorithm is given below. For simplicity, the full algorithm is broken up into four algorithms. Algorithm 1 describes how the dynamic parameter estimates are obtained. Algorithm 2 describes how the static parameter estimates are obtained. Finally algorithms 3 and 4 set up the grid searches for the dynamic parameters.

The first grid search uses intelligent choices for determining the grid to search. For example, it has been shown that most carbohydrates are converted to glucose and then energy by the body within two hours of ingestion. This amount of time needed for the carbohydrates to be utilized by the system, i.e. the human body, can be viewed as the residence time. For a Wiener network, input \( i \) has residence time \( 2\tau_i\zeta_i \). Thus the grid search was designed so that the center of the grid search yielded a residence time around 120 minutes. Other food variables are known to take longer to be converted into glucose and utilized as energy, and the activity variables appear to have a short residence time in our experience. However, the dynamic parameter \( \tau_i \) for input \( i \) varies among the activity variables.

As for the second grid search, the values of \( \tau \) to be used are determined by the value of \( \tau'_i \) found from the first grid search since the candidates for \( \tau \) from the first grid search are equally spaced. The values of \( \zeta \) used for the grid search are also dependent on \( \zeta'_i \), but in some cases the grid generated from it includes negative values of \( \zeta_i \). If this is the case, the minimum value used in the grid search for \( \zeta_i \) is set to 0.0001. Then equally spaced points between 0.0001 and the maximum value as determined by \( \zeta'_i \) are used for the second grid search instead.
Algorithm 1 Summary of grid searches for dynamic parameters.

Let $y_{Tr}$ denote the observed BGC in the training set and $y_{Va}$ the observed BGC in the validation set.
Let $\hat{y}_{Tr}$ denote the predicted BGC in the training set and $\hat{y}_{Va}$ the predicted BGC in the validation set.
Let $v_i$ denote dynamic variable $i$ in the training set and $\hat{v}_i$ dynamic variable $i$ in the validation set.
$\bar{y} = [\hat{y}_{Tr} \ y_{Va}] = \bar{y}_{Tr} = \frac{1}{n} \sum_{i=1}^{n} y_i$, where $n$ is the number of observations in the training set.

Set up first grid search. See Algorithm 3.

Let $n_1$ be the length of $\tau$ and $n_2$ be the length of $\zeta$.

Let $R^f, R^v$ be $n_1 \times n_2$ matrices and $r^* = r^{**} = 0$.

while $i = 1$ to 8 do
  for $j = 1$ to $n_1$ do
    for $k = 1$ to $n_2$ do
      $r^{f1} = \text{Corr}(y, \hat{y}_{Tr} + s_i v_i^T (\tau_j, \zeta_k)), r^{v1} = \text{Corr}(y, \hat{y}_{Va} + s_i v_i^T (\tau_j, \zeta_k))$
      $r^{f2} = \text{Corr}(y, \hat{y}_{Tr} - s_i v_i^T (\tau_j, \zeta_k)), r^{v2} = \text{Corr}(y, \hat{y}_{Va} - s_i v_i^T (\tau_j, \zeta_k))$
      Choose $t$ such that $r^{f1} = \max\{r^{f1}, r^{f2}\}$.
      Set $R^f_{jk} = r^{f1}$ and $R^v_{jk} = r^{v1}$.
    end for
  end for
if $\max R^f < r^*$ then
  Choose $j, k$ such that $R^v_{jk} = \max R^v$.
  Set $\hat{\tau}_i = \tau_j$ and $\hat{\zeta}_i = \zeta_k$.
  break
end if
Choose $j, k$ such that $|R^f_{jk} - R^v_{jk}|$ is minimized, $R^f_{jk} > r^*$ and $R^v_{jk} > r^{**}$.

$r^* = R^f_{jk}, r^{**} = R^v_{jk}$

Determine new grid for grid search whose center is $(\tau_j, \zeta_k)$. See Algorithm 4.

Let this grid be $n_1 \times n_2$ and $R'^f, R'^v$ be $n_1 \times n_2$ matrices.

Let $S$ be an $n_1 \times n_2$ matrix whose entries are 1.

for $j = 1$ to $n_1$ do
  for $k = 1$ to $n_2$ do
    $r^{f3} = \text{Corr}(y, \hat{y}_{Tr} + s_i v_i^T (\tau_j, \zeta_k)), r^{v3} = \text{Corr}(y, \hat{y}_{Va} + s_i v_i^T (\tau_j, \zeta_k))$
    $r^{f4} = \text{Corr}(y, \hat{y}_{Tr} - s_i v_i^T (\tau_j, \zeta_k)), r^{v4} = \text{Corr}(y, \hat{y}_{Va} - s_i v_i^T (\tau_j, \zeta_k))$
    Choose $x$ such that $r^{v2} = \max\{r^{v2}, r^{v4}\}$.
    Set $R'^f_{jk} = r^{f2}$ and $R'^v_{jk} = r^{v2}$.
    if $r^{f4} = r^{f2}$ then
      $S_{jk} = -1$
    end if
  end for
end for
Choose $j, k$ such that $|R'^f_{jk} - R'^v_{jk}|$ is minimized, $R'^f_{jk} \geq r^*$ and $R'^v_{jk} \geq r^{**}$.

Set $r^* = R'^f_{jk}$ and $r^{**} = R'^v_{jk}$.

Set $\hat{\tau}_i = \tau_j$ and $\hat{\zeta}_i = \zeta_k$.

end while
Algorithm 2  Summary of algorithm’s determination of static parameters.

Let $\hat{y}$ be the predicted values of BGC from the grid search.
Fit the model $y = \beta_0 + \sum_{i \text{ kept}} \beta_i v_i(\hat{\tau}, \hat{\zeta}) + \epsilon$ via least squares.

Determine $\hat{y}_r^*$ and $\hat{y}_v^*$ from the least squares parameter estimates
$r_{\text{fit}} = \text{Corr}(y_r, \hat{y}_r^*)$.
$r_{\text{val}} = \text{Corr}(y_v, \hat{y}_v^*)$.
if $r_{\text{fit}} > r^*$ & $r_{\text{val}} > r^{**}$ then
  Set $\hat{y} = \hat{y}^*$ from least squares fit, $r^* = r_{\text{fit}}$ and $r^{**} = r_{\text{val}}$.
else
  Set $\hat{\beta}_i = \pm S_{jk}$
  Set $\hat{y} = \hat{y}$ found from grid search.
end if

Determine if removing regressors results in fitting the validation dataset better.

Algorithm 3  Setup for first grid search.

$\tau = [11 \ 12 \ 13 \ \ldots \ 60]$
$\zeta = [.001 \ .005 \ .01 \ .015 \ .02 \ .03 \ .08 \ .12 \ \ldots \ 2]$
$s = [1 \ 5 \ 8 \ .005 \ 2 \ 8 \ .0005]$
if $i = 2$ then
  $\tau = [153 \ 156 \ 159 \ \ldots \ 300]$
else if $i = 3$ then
  $\tau = [1410 \ 1420 \ 1430 \ \ldots \ 1900]$
  $\zeta = [.01 \ .01 + .149/56 \ .01 + 2 * .149/56 \ \ldots \ 1.50]$
else if $i = 5$ then
  $\tau = [61 \ 62 \ 63 \ \ldots \ 110]$
  $\zeta = [.01 \ .01 + .34/56 \ .01 + 2 * .34/56 \ \ldots \ .35]$
else if $i = 6$ then
  $\tau = [51 \ 52 \ 53 \ \ldots \ 100]$
else if $i = 7$ then
  $\tau = [206 \ 212 \ 218 \ \ldots \ 500]$
else if $i = 8$ then
  $\tau = [71 \ 72 \ 73 \ \ldots \ 120]$
  $\zeta = [.001 \ .001 + .08/56 \ .001 + 2 * .08/56 \ \ldots \ .081]$
end if
Algorithm 4  Setup for second grid search.

Let $\tau'_i$ and $\zeta'_i$ denote the parameter estimates chosen from the first grid search.

$\theta = [\tau'_0 - .95 \ \tau'_1 - .9 \ \cdots \ \tau'_i - .95 \ \tau'_i + .05 \ \cdots \ \tau'_i + .95]$,

if $i = 3$ then

$\theta = [\tau'_i - 9.5 \ \tau'_i - 9 \ \cdots \ \tau'_i \ \tau'_i + 0.5 \ \cdots \ \tau'_i + 9.5]$,

else if $i = 2$ then

$\theta = [\tau'_i - 2.85 \ \tau'_i - 2.7 \ \cdots \ \tau'_i \ \tau'_i + 0.15 \ \cdots \ \tau'_i + 2.85]$,

else if $i = 7$ then

$\theta = [\tau'_i - 5.7 \ \tau'_i - 5.4 \ \cdots \ \tau'_i \ \tau'_i + 0.3 \ \cdots \ \tau'_i + 5.7]$,

end if

if $i = 3$ then

if $\zeta'_i - 1.48/56 < .0001$ then

$c = \zeta'_i + 1.48/56 - .0001$

$\zeta = [ .0001 \ .0001 + c \ .0001 + 2c \ \cdots \ .0001 + 54c \ \zeta'_i + 1.48/56 ]$

else

$\zeta = [ \zeta'_i - \frac{1.48}{56} \ \zeta'_i - \frac{1.48}{56} + \frac{2.96}{5630} \ \zeta'_i - \frac{1.48}{56} + \frac{5.92}{3060} \ \cdots \ \zeta'_i - \frac{1.48}{56} + \frac{162.8}{3080} \ \zeta'_i + \frac{1.48}{56} ]$

end if

else if $i = 5$ then

$\zeta = [ \zeta'_i - \frac{33}{56} \ \zeta'_i - \frac{33}{56} + \frac{66}{3080} \ \zeta'_i - \frac{33}{56} + \frac{132}{3080} \ \cdots \ \zeta'_i - \frac{33}{56} + \frac{35.64}{3080} \ \zeta'_i + \frac{33}{56} ]$

else if $i = 8$ then

if $\zeta'_i - .079/56 < .0001$ then

$c = \zeta'_i + .079/56 - .0001$

$\zeta = [ .0001 \ .0001 + c \ .0001 + 2c \ \cdots \ .0001 + 54c \ \zeta'_i + .079/56 ]$

else

$\zeta = [ \zeta'_i - \frac{.079}{56} \ \zeta'_i - \frac{.079}{56} + \frac{158}{3080} \ \zeta'_i - \frac{.079}{56} + \frac{316}{3080} \ \cdots \ \zeta'_i - \frac{.079}{56} + \frac{8.69}{3080} \ \zeta'_i + \frac{.079}{56} ]$

end if

else

if $\zeta'_i = .001$ then

$\zeta = [ .0001 \ .0002 \ \cdots \ .0049 ]$

else if $\zeta'_i \leq .04$ then

$\zeta = [ \zeta'_i + .001 \ zeta'_i + .002 \ \cdots \ zeta'_i - .002 \ zeta'_i - .001 ]$

else if $\zeta'_i = 2$ then

$\zeta = [ 1.965 \ 1.97 \ 1.975 \ \cdots \ 2.095 \ 2.1 ]$

else

$\zeta = [ \zeta'_i + .002 \ \zeta'_i + .004 \ \cdots \ \zeta'_i - .004 \ \zeta'_i - .002 ]$

end if

end if
CHAPTER 4. THE DEVELOPMENT OF A MODEL-BASED NONINVASIVE GLUCOSE MONITORING DEVICE FOR NON-INSULIN DEPENDENT PEOPLE

A paper to be submitted to Computers and Chemical Engineering

Derrick K. Rollins, Lucas P. Beverlin, Kaylee Kotz, Nisarg Vyas, David Andre, Greg Welk, and Warren Franke

Abstract

Continuous-time glucose monitoring (CGM) effectively improves glucose control by providing frequently sampled information that allows the user to associate changes in their glucose levels with changes in their behavior. Currently, the most widely used and effective CGM devices rely on a sensor that is inserted invasively under the skin. Due to invasiveness and cost, the primary users of current CGM devices are insulin dependent people (type 1 and some type 2 diabetics). This research is concerned with the development of a non-invasive CGM device that would be used by health conscious non-insulin dependent people (including non-diabetics) that would help reduce obesity as well as the onset and the progression of type 2 diabetes. Specifically, this work seeks to develop an accurate device that uses food, activity and stress variables to infer glucose concentration at the rate of CGM devices, i.e., every 5 minutes. Using 22 test subjects with 4 weeks of data
collection each, results have been obtained to support the modeling viability necessary to build a monitoring device. Accomplishments include the ability to develop subject-specific models under several modeling challenges and restrictions. Results are presented for models developed after three days, 2 weeks and four 4 weeks that support an initial calibration period of three days with accuracy improving over time and no need for lancet measurements after 3 to 4 weeks. Thus, since the model does not drift, this device would not appear to need glucose measurements for calibration once it is fully calibrated to the user wearing the device.

Keywords: Soft Sensor, Wiener Modeling, Block-Oriented Modeling, Glucose Modeling

4.1 Introduction

Recent research has supported the belief that real-time, frequent, glucose monitoring can improve blood glucose control over infrequent monitoring provided through the use of lancet glucose meters [60, 105]. Frequent glucose measurement capability is referred to as continuous-glucose monitoring (CGM). Although not really continuous, current devices can deliver on-line glucose measurements as fast as every five minutes. Nonetheless, this is a substantial increase over lancet monitoring that only produces a few values per day, at best. CGM improves the user’s ability to achieve better glucose control by providing highly frequent, real-time, glucose concentration levels that enables the user to see relationships between glucose levels and recent activity behavior and/or food consumption. For example, a user is able to see immediately the impact of the size of a meal on the level of glucose change along with the duration.

Currently, the most widely used CGM devices, such as the Minimed Guardian® REAL-Time System (Medtronic Minimed, Northridge, CA) [9] and the SEVEN® PLUS System (DexCom, Inc. San Diego, CA) [3], rely on a sensor that is inserted invasively under
the skin. Sensors cost from $35 to $60 and last 3 days to a week. Thus, two significant drawbacks of these devices are comfort and cost. Given these drawbacks, these devices are not widely used except by insulin dependent users that depend strongly on frequent monitoring to reduce large swings in glucose variation. For this reason, these devices are less likely to be used by non-insulin dependent people, including non-diabetics, pre-diabetics, and non-insulin dependent type 2 diabetics. It is, therefore, the objective of this research to develop a glucose monitoring device with acceptable attributes for non-insulin dependent people. Our belief is that by achieving this objective, a significant advancement can be made in reducing the number of people that are diagnosed with type 2 diabetes and the number of type 2 diabetics who require insulin for controlling their blood glucose concentration.

To accomplish our objective, we feel that this CGM device must have the following attributes:

1. Completely non-invasive
2. Simple or no reliance for food entry
3. Relatively short calibration period
4. Require few to no lancet measurements for calibration
5. An accuracy that is comparable to lancet meters

More specifically, this research proposes to use an inferential sensor for BGC that uses only noninvasive inputs that updates every five minutes and is calibrated from four lancet BGC measurements per day after only a few days of data collection. Thus, since a sensor is calibrated from user data, the model developed for each person is said to be "subject-specific." Our approach to achieve this goal is to use a novel modeling method to infer
BGC using non-invasive input measurements for each subject from variables representing food, activity, circadian rhythm, and stress levels. The main component of this system is a SenseWear Pro 3 armband (BodyMedia, Inc., Pittsburgh, PA) shown in Figure 4.1. This armband will automatically collect the wearer’s activity and stress data in the form of several inputs. The food information will be entered manually by the user via the time stamp button on the armband. The model and model development algorithm will reside in the armband and will use a Wiener approach based on the work of Rollins et al. [86]. Data from a lancet glucose meter will be entered automatically or manually and will be used to develop a subject-specific model for the person wearing the device. After the model is completely developed, lancet measurements will no longer be needed for calibration. An interface device will be connected to the armband to display the glucose in five minute intervals. To our knowledge there is no truly non-invasive device or approach that uses the combined set of these types of input variables to infer glucose concentration from a model in real time.

The efficacy of this approach is demonstrated using 22 pre-diabetic and non-insulin dependent type 2 diabetic subjects. About 4 weeks of IRB approved data collection was
taken on each subject. For these subjects, modeling results will be given for three model development periods: three days, one week and two weeks. For these training periods, zero, one week, and two weeks of validation data, respectively, as well as 25 days, two weeks, and zero days of test data, respectively, are available for cases with four weeks of total data collection.

The challenges of the modeling problem include estimating a large set of dynamic and static parameters from a small set of training samples, minimizing the possibility of overfitting, a lack of initial steady state data, utilizing meals with a designation of small, medium and large in the model, as well as frequent and arbitrary removal of the armband. Through novel modifications of the Rollins et al. [86] approach, this work demonstrates an ability to overcome these challenges. Thus, this work has promising potential to develop an effective inferential continuous-time BGC sensor for the target population of non-insulin dependent people. This article presents this work by providing details of the proposed approach in the next section. After this section, the next one presents modeling results for the 22 subjects for the three training periods given above. The last section summarizes this work and discusses future work that will lead to a proof-of-concept study for the development of a prototype monitoring device.

4.2 The Modeling Approach

The basic objective of this work is the development of a "soft sensor," (also called a "virtual sensor") subject-specific, blood glucose concentration (BGC) monitoring system that can be accurately obtained with a very small amount of data. A soft sensor is basically an inferential model that is developed from process data or other measured variables that are termed inputs. This approach has seen wide applications in process monitoring and control applications in recent years [33, 63] due to advancements in computer hardware,
software, and measurement technology. While inferential modeling of BGC has been done by a number of researchers [25, 27, 77, 79, 96, 99], particularly in type 1 applications, this is the first approach that we are aware of that seeks to develop an inferential model for non-insulin dependent subjects using infrequent lancet measurements from the subject’s personal glucose meter. This is the major challenge because the frequency of BGC data for model building and inference is much less than the virtual measurement rate of 5 minutes. This limitation can potentially restrict the number of parameter estimates and the use of previously measured BGC in inference, and thus, severely impact accuracy.

The information for the development of a soft sensor comes from two sources, the response data set and the input data set. Since the information content of BGC concentration is quite limited in this approach, our proposed modeling approach strongly relies on the input data set for information on glucose behavior. More specifically, the input data set consists of meal size with three levels, and six variables from the armband. The inputs are shown in Table 4.2. The ability to map the available input/output information to accurate sensor measurements depends on the model structure, the model building procedure, and the inferential algorithm that we are calling the "Inference Engine." The model structure consists of the mathematical functions and the network that tie these functions together. The model building (i.e., identification) procedure is the process of using input/output information to estimate the values of unknown parameters in the mathematical functions. The Inference Engine is the equation used to obtain the soft sensor measurements at the desired sampling frequency. This equation represents final input selection and parameter estimation, and the use of lancet glucose measurements to enhance reliability. The purpose of the next three sections is to describe these three components of the proposed technique in detail.
Table 4.1 Input variables: Food (1); Armband (2-7)

<table>
<thead>
<tr>
<th>Input</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Food</td>
</tr>
<tr>
<td>2</td>
<td>Transverse acceleration - peaks</td>
</tr>
<tr>
<td>3</td>
<td>Heat flux - average</td>
</tr>
<tr>
<td>4</td>
<td>Longitudinal acceleration - average</td>
</tr>
<tr>
<td>5</td>
<td>Transverse acceleration - MAD</td>
</tr>
<tr>
<td>6</td>
<td>Galvanic skin response - average</td>
</tr>
<tr>
<td>7</td>
<td>Time of day</td>
</tr>
</tbody>
</table>

4.3 Modeling Structure

The modeling structure of this application must be capable of accurate parameter estimation under a small sample size \(n\), effectively handling several inputs with different dynamic behavior, and mild extrapolation. The modeling network we have chosen is the Wiener block-oriented network shown in Figure 5.1. As shown, each input \(i\) enters a separate linear dynamic block and the outputs from these blocks are collected into non-observable variables \((v_i)\). These \(v_i\) are then passed through a static block which can be any type of function. The Wiener network is defined by the attribute of allowing separate dynamic behavior for each input and this is a critical reason that it is unique for this application where the input dynamics can be quite different. This attribute is also exploited to breakdown the correlation of inputs via the passing of their weakly correlated dynamic counterparts through the static block.

The dynamic function that we have selected for this application follows the modeling work of Rollins et al. [86] and it is:

\[
\tau_i^2 \frac{d^2 v_i}{dt^2}(t) + 2\tau_i \zeta_i \frac{dv_i}{dt}(t) + v_i(t) = \tau_{ai} \frac{dx_i}{dt}(t) + x_i(t),
\]  

(4.1)
Using backward difference finite derivative approximations, Equation 5.1 gives [86]
\[ v_i(t) = \frac{2\tau_i^2 + 2\tau_i\zeta_i\Delta t}{\tau_i^2 + 2\tau_i\zeta_i\Delta t + \Delta t^2} v_i(t - \Delta t) - \frac{\tau_i^2}{\tau_i^2 + 2\tau_i\zeta_i\Delta t + \Delta t^2} v_i(t - 2\Delta t) + \frac{\tau_{ai}\Delta t + \Delta t^2}{\tau_i^2 + 2\tau_i\zeta_i\Delta t + \Delta t^2} x_i(t - \Delta t) - \frac{\tau_{ai}\Delta t}{\tau_i^2 + 2\tau_i\zeta_i\Delta t + \Delta t^2} x_i(t - 2\Delta t) \] (4.2)
which can be rewritten as
\[ v_{i,t} = \delta_{1,i} v_{i,t-\Delta t} + \delta_{2,i} v_{i,t-2\Delta t} + \omega_{1,i} x_{i,t-\Delta t} + \omega_{2,i} x_{i,t-2\Delta t} \] (4.3)
such that \( \omega_{2,i} = 1 - \delta_{1,i} - \delta_{2,i} - \omega_{1,i} \). This constraint is used to impose a unity gain restriction for the linear dynamic blocks. Here the sampling time is \( \Delta t \). In the Laplace
domain, the linear dynamic functions are

\[ G_i(s) = \frac{V_i(s)}{X_i(s)} = \frac{\tau_{ai}s + 1}{\tau_i s^2 + 2\tau_i \zeta_i s + 1} \]  

(4.4)

Note that the number of dynamic parameters associated with each input is three. This small number is a strength that we exploit to obtain parameter estimates under limited sampling as we discuss below. The function \( f(V) \) is called the static function and is a function of \( v_{i,t}, i = 1, \ldots, p \). This function can theoretically be of any form. For effectiveness under mild extrapolation and parameter estimation (as discussed below) we have chosen a linear regression model of the form:

\[ y_t = \eta_t + \epsilon_t = a_0 + a_1 v_{1,t} + \cdots + a_p v_{p,t} + \epsilon_t \]  

(4.5)

where \( \epsilon_t \) is the error term and assumed to be independently normally distributed with mean 0 and variance \( \sigma^2 \) for all \( t \).

As stated in Rollins et al. [86], the modeling objective is simply to maximize the true but unknown correlation coefficient between measured and fitted BGC. This quantity is represented by and estimated by \( r_{\text{fit}} \). Thus, under this criterion a model is considered useful, if, and only if,

\[ \rho_{y,\hat{y}} > 0. \]  

(4.6)

Since the degree of usefulness increases with \( \rho_{y,\hat{y}} \), the goal is to obtain the largest (as close to the upper limit of 1) value as possible. Due to the highly complex mapping of the parameters into the response space of \( r_{\text{fit}} \), the following indirect criterion is used in obtaining the parameter estimates as described in Rollins et al. [86]:

Maximize \( r_{\text{fit}} \) by minimizing \( \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \)

Subject to: \( \zeta_i > 0, \tau_i > 0 \) \( \forall i \),

(4.7)
4.4 Model Identification Procedure

Using the Wiener network with the functional forms of given by Equations 1-5, we have developed a procedure that can accurately estimate the $3p$ dynamic parameters and the $p + 1$ static parameters even when the number of sampling times ($n$) is much less than $4p + 1$, the total number of dynamic and static parameters. This procedure requires each input to have a separate set of dynamic parameters as met by the Wiener network but not other common networks (e.g., an Auto Regressive Moving Average with eXogenous (ARMAX) variables) [86]. Here the dynamic parameters for each input separately. This can be done by exploiting the fact that, with $a_j = 0$ except for $j = 0$ and $j = i$, $r_{fit}$ for the model

$$y_t = a_0 + a_i v_{i,t} + \epsilon_t,$$  \hspace{1cm} (4.8)

depends only on the dynamic parameters associated with input $i$, i.e., $\tau_i, \zeta_i$ and $\tau_{ai}$. A proof of this is given in Appendix A. Therefore, the proposed model identification procedure estimates the dynamic parameters for each input using Equation 4.8, one input at a time. Note that at most five parameters are estimated each time which is less than $n = 12$ for three days of data collection. After obtaining the dynamic parameters for each input, with their values fixed, the $p = 7$ (maximum value) static parameters under Equation 4.5 are estimated which is also less than $n = 12$.

To estimate parameters, the model identification procedure uses either one data set (training only) or two data sets (training and validation) depending on the amount of available data. When using one data set, the number of inputs may be restricted to the most reliable set (e.g., meal and the best arm band variable) and the optimization goal under Equation 4.5 or Equation 4.8 is to maximize $r_{fit}$ under the criterion of Equation 4.7. This estimation procedure is said to be “unsupervised” [47]. When enough data
are collected to split the data into a training and validation set, “supervised” training is used for estimating the dynamic parameters and unsupervised training is used to obtain the static parameters. To dynamically fit similar behavior in both data, the dynamic parameters are adjusted under Equation 4.8 to give similar yet high $r_{fit}$ values for the training and validation data sets. This procedure is used to guard against overfitting, i.e., fitting BGC behavior in the training set that is not due to true variation in BGC. The success of supervised training is evaluated through the use of a third set of data called the “test set” which requires enough data to split the data into three sets.

Effective use of Equation 4.3 depends on accurate initial values for the $v_i$'s. These values are needed at the start of data collection and anytime the armband is placed on the arm for use after having been removed. When the dynamics for an input are fast, the accuracy of the initial values is less of a concern because $v_i$ will stabilize relatively quickly. We have developed ways to obtain initial values for the $v_i$'s under three scenarios: 1. Start of data collection; 2. Short armband removal periods and; 3. Long armband removal periods. At the start of data collection we use values that we have obtained and evaluated from modeling the 22 subjects. For short removal periods, defined to be less than an hour, we use the $v_i$ values at the time the armband was removed. For long removal periods, defined to be greater than an hour, we will use either a standard start up set of values or their values at the time the armband was removed depending on the length of time the armband was off the arm and the dynamics of the input variable. At present, this relationship has not been finalized and is still a topic of considerable research.

4.5 Development of the Inference Engine

After obtaining a full set of parameter estimates two more refinements are done before the proposed method is commissioned for real-time monitoring. The first one is elimination
of inputs that adversely affect accuracy when combined with the other inputs. This is accomplished using a backwards elimination strategy that keeps the dynamic parameters fixed and estimates the static parameters under the model given by Equation 4.5. The first criterion for input elimination is a negative contribution to \( r_{\text{fit}} \) either in the training set or validation set. In the final model all the inputs must have a positive contribution to \( r_{\text{fit}} \). In essence, this means that it is desired that all static parameter estimates must be greater than 0. A second criterion for input \( i \) to be removed is that setting its corresponding \( a_i \) to zero increases \( r_{\text{fit}} \). Once an input is eliminated, the model is refit with the remaining inputs and the criteria are analyzed again.

The final refinement involves the use of lancet glucose to help to reduce model bias. Since these measurements are infrequent and are not measured at a constant rate, it is not possible to build a correction model based on the correlation of residuals. The correction equation that we use comes from Rollins et al. [86] where only the most recent measurement, at \( t = t^* \), is used. This equation, which represents the proposed virtual sensor, is given as

\[
\hat{y}_t = \hat{\eta}_t + (y_{t^*} - \hat{\eta}_{t^*})\lambda \frac{t - t^*}{\Delta t} \tag{4.9}
\]

subject to \( t > t^* \) and \( 0 < \lambda < 1 \), where \( \lambda \) is an adjustable constant, \( \hat{\eta}_t \) is the estimated BGC at time \( t \) under the Equation 4.5 model, \( \hat{\eta}_{t^*} \) is the estimated BGC at time \( t = t^* \) under the Equation 4.5 model, \( y_t \) is the virtual (i.e., soft) sensor value for the proposed method at time \( t \), and \( y_{t^*} \) is the lancet BGC measurement at \( t = t^* \), which is assumed to be measured without error. Note that \( y_{t^*} - \hat{\eta}_{t^*} \) represents that amount of correction and this correction diminishes as time increases based on the value of \( \lambda \), which is close to 1. Thus, by the time the next lancet measurement is taken, \( \hat{y}_t = \hat{\eta}_t \). This means that at \( t = t^* \), \( \hat{y}_t \approx \hat{\eta}_{t^*} \), at \( t = t^* + \Delta t \), \( y_t \approx \hat{y}_{t^*} \), and at \( t = t^* + k\Delta t \), with \( k \gg 1 \) and before the next lancet measurement, \( y_t \approx \hat{\eta}_t \). That is, at the time of the lancet measurement, the proposed
virtual monitor would display a value close to $\hat{\eta}_t$, the next value would be close to the lancet measurement, as time proceeds the lancet value would have less corrective influence as the monitor would rely more on the model to infer BGC. When two sets are used to estimate model parameters, $\lambda$ can be set to give the most accurate values in the validation set. When only a training set is used to estimate the model parameters, a default value can be used based on results from modeling several subjects.

### 4.6 Clinical Study Results for 22 Subjects

Using 22 test subjects with 4 weeks of data collection (in most cases and slightly under 4 weeks in other cases except for Subject 1 and 8 which had only about 3 weeks of data due to loss of data), we have obtained results to support the modeling viability necessary to build an armband monitoring device. It should be noted that these data sets were collected for another study (see Beverlin et al. [8]) and modifications had to be made to these data sets for use in this study. First, food quantities, which were in grams, had to be converted to food indices to mimic time stamping. The meal sizes were modified to represent 2, 3, or 4 time stamps, for small (e.g., snacks), medium (e.g., regular meals) and large (e.g., a meal with more than 100 carbohydrates), respectively. Two timestamps were converted to indices of 1, 0, 0, for carbohydrates, fats and proteins, respectively. Three timestamps were converted to indices of 2, 1, 1, for carbohydrates, fats and proteins, respectively. Lastly, three timestamps were converted to indices of 3, 2, 2, for carbohydrates, fats and proteins, respectively. After investigating how many of these inputs to use in the modeling, it was determined that only carbohydrates should be used. Hence input 1 in Table 4.2 corresponds to carbohydrates. Second, the lancet sampling rate was only four times per day at fixed times. Continuous-time (CT) blood glucose concentrations (BGC) were taken only at 8 am, noon, 4 pm and 8 pm unless unavailable and then the nearest value was taken with
no more than 4 values used per day. The monitoring period was taken to be from 8 am to 10 pm daily which means that this was the only period that virtual BGC were reported. Thus, the period from 10 pm to 8 am was taken to be a non-monitoring period to mimic unnecessary monitoring during the sleeping period.

We are envisioning a device that will develop the virtual sensor (i.e., Equation 4.9) using the user’s data in multiple phases. The phases that we give here are for the purpose of evaluation and are not likely to be the ones used in actual practice as the optimal phases are still under considerable research and will likely be evolutionary as more and more data become available, such as an adaptive scheme. The first phase in this evaluation will consist of a period of about three days (i.e. $n = 12$) and only food will be used to determine the model under Equation 4.8 (i.e. $\hat{\eta}_t$). Future work will evaluate the addition of other activity variables in this phase. For the purpose of this evaluation, this model can be considered to be active from day 4 until about day 14, when the second phase model will be developed. (However, results will be reported for remaining 25 days as a test set.)

The second phase model will be built under Equation 4.5 from about 14 days of data or about $n = 56$ sampling times. All inputs will be considered in this phase. The second phase model will be operational from day 15 to day 28, and producing test sets with up to 14 days of results for analysis and evaluation. After day 28, the third or final phase model for this evaluation will be developed from 28 days of data or 112 sampling times. Since there are at most 28 days of data for any subject, the data will be split into two sets, a 2 week training set and a 2 week validation set.

The results of this study are reported using four statistics. The first one is called the averaged error (AE) and is simply the average value of the residuals:

$$AE = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i).$$  \hspace{1cm} (4.10)
where \( m \) is the number of lancet measurements in the statistic being estimated. The second one is called the averaged absolute error and is similar to Equation 4.10 except that the absolute difference is used for the term in the summation as follows:

\[
\text{AAE} = \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|.
\] (4.11)

The next statistic is a scaled AAE and is called the relative AAE (RAAE). This measure of performance is determined by dividing Equation 4.11 by the standard deviation of the lancet values used to calculate AAE as follows:

\[
\text{RAAE} = \frac{\text{AAE}}{\sqrt{\frac{1}{m-1} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2}} = \frac{\text{AAE}}{\hat{\sigma}_y}.
\] (4.12)

RAAE is a relative AAE measure that accounts for large spread in the glucose variation of subjects. For replicated lancet measurement, the study in Rollins et al. [86] determined RAAE to be about 0.60 for validation. Thus, we will assume that a value around 0.60 is comparable to the performance of a glucose lancet meter. The last statistic to measure performance is \( r_{fit} \). Based on the results in Rollins et al. [86] for a type 2 subject and in Beverlin et al. [8] for these subjects, we set a goal for \( r_{fit} \) to be greater than 0.40 with a value greater than 0.60 considered excellent. Parameter estimation was done using the Excel® Solver add-in.

The first results that we present are for the case consisting of 2 weeks of training and 2 weeks of validation and are given for the model under Equation 4.5 only (i.e. \( \hat{\eta}_t \)). These results, shown in Table 4.6, are the best ones as they represent the largest training set with the largest validation set. First, evidence that the convergence criterion for training in all cases was met can be seen by the values of AE being 0.0 for all subjects. In training and validation AAE averaged about 17.1 mg/dL and 19.4 mg/dL, respectively. This difference is partly due to bias in the validation set as evidenced by the AE mean value of -2.4 mg/dL.
For some subjects AE is quite large and as high as 26.0 mg/dL for Subject 12. The average RAAE values for training and validation are 0.64 and 0.74, respectively, which are close to the target value of 0.6. Note that for training, RAAE is $\leq 0.70$ for 18 subjects and for validation, RAAE is $\leq 0.70$ for 8 subjects. RAAE ranged from a low of 0.50 to a high of 1.17 on validation cases which can be attributed greatly to the large variation in AE indicating greater bias for validation data. The values for $r_{fit}$ on training and validation are very comparable averaging 0.54 and 0.51, respectively. In addition, the modeling objective of maintaining similar $r_{fit}$ performance on both sets was met quite well.

Next, the results exclusive of food, i.e., only the armband inputs (variables 2-7 in Table 1) are discussed. These results are given in Table 4.6 for 2 weeks of training and 2 weeks of validation. In this table two sets of validation results are given. The first set is for the model under Equation 4.5 where the estimate is designated by $\hat{\eta}_t$. The second set represents the soft sensor results determined under Equation 4.9 and are designated by $\hat{y}_t$. Given that food is not included, the results are actually not that much worse than those in Table 4.6. The most critical drop is in $r_{fit}$ that went from 0.54 and 0.51 on training and validation, respectively in Table 4.6 to 0.46 and 0.43 on training and validation, respectively in Table 4.6. For two subjects (3 and 17) the Table 4.6 results are actually better than the Table 4.6 results that used all the inputs in validation under Equation 4.5. In summary, although the model with just the armband variables performed well, the improvement obtained with the addition of food was quite significant and indicates that this variable has a large enough contribution that it should be included. In addition, the armband provides variables that are very useful in estimating BGC.

The next study is for 1 week of training, 1 week of validation and 2 weeks of testing. These results are given in Table 4.4. This is the first table with testing results. The training and validation results in this table are comparable to the training and validation results in
Table 4.2  Results under Equation 4.5 for the 22 test subjects with 2 weeks of training and 2 weeks of validation. AE and AAE values are in mg/dL.

<table>
<thead>
<tr>
<th>Subject</th>
<th>2 weeks Training</th>
<th>2 weeks Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AE</td>
<td>AAE</td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td>13.31</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>11.37</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>10.46</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>8.88</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>17.27</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>12.23</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>19.55</td>
</tr>
<tr>
<td>8</td>
<td>0.00</td>
<td>16.66</td>
</tr>
<tr>
<td>9</td>
<td>0.00</td>
<td>39.11</td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>13.63</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>19.19</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>35.39</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>11.79</td>
</tr>
<tr>
<td>14</td>
<td>0.00</td>
<td>9.53</td>
</tr>
<tr>
<td>15</td>
<td>0.00</td>
<td>12.04</td>
</tr>
<tr>
<td>16</td>
<td>0.00</td>
<td>15.16</td>
</tr>
<tr>
<td>17</td>
<td>0.00</td>
<td>16.79</td>
</tr>
<tr>
<td>18</td>
<td>0.00</td>
<td>16.52</td>
</tr>
<tr>
<td>19</td>
<td>0.00</td>
<td>34.20</td>
</tr>
<tr>
<td>20</td>
<td>0.00</td>
<td>16.73</td>
</tr>
<tr>
<td>21</td>
<td>0.00</td>
<td>13.37</td>
</tr>
<tr>
<td>22</td>
<td>0.00</td>
<td>13.11</td>
</tr>
<tr>
<td>Mean</td>
<td>0.00</td>
<td>17.10</td>
</tr>
<tr>
<td>Stdev</td>
<td>0.00</td>
<td>8.34</td>
</tr>
</tbody>
</table>
Table 4.3  Armband (inputs 2-8 in Table 1) modeling results for the 22 test subjects with 2 weeks of training and 2 weeks of validation. AE and AAE values are in mg/dL. AE is not reported for Training since it is 0 for all subjects.

<table>
<thead>
<tr>
<th>Sub.</th>
<th>2 weeks Training AAE</th>
<th>RAAE</th>
<th>( r_{\text{fit}} )</th>
<th>2 weeks Validation AE</th>
<th>AAE</th>
<th>RAAE</th>
<th>( r_{\text{fit}} )</th>
<th>2 weeks Validation AAE</th>
<th>RAAE</th>
<th>( r_{\text{fit}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.54</td>
<td>0.69</td>
<td>0.41</td>
<td>2.57</td>
<td>21.66</td>
<td>0.69</td>
<td>0.40</td>
<td>2.56</td>
<td>21.65</td>
<td>0.69</td>
</tr>
<tr>
<td>2</td>
<td>13.08</td>
<td>0.62</td>
<td>0.64</td>
<td>13.26</td>
<td>20.09</td>
<td>0.75</td>
<td>0.60</td>
<td>10.36</td>
<td>18.12</td>
<td>0.68</td>
</tr>
<tr>
<td>3</td>
<td>10.70</td>
<td>0.67</td>
<td>0.51</td>
<td>11.37</td>
<td>14.67</td>
<td>0.85</td>
<td>0.53</td>
<td>7.69</td>
<td>11.99</td>
<td>0.69</td>
</tr>
<tr>
<td>4</td>
<td>9.49</td>
<td>0.79</td>
<td>0.32</td>
<td>7.26</td>
<td>10.58</td>
<td>0.87</td>
<td>0.39</td>
<td>7.26</td>
<td>10.58</td>
<td>0.87</td>
</tr>
<tr>
<td>5</td>
<td>17.79</td>
<td>0.68</td>
<td>0.42</td>
<td>0.24</td>
<td>14.36</td>
<td>0.70</td>
<td>0.41</td>
<td>0.24</td>
<td>14.35</td>
<td>0.70</td>
</tr>
<tr>
<td>6</td>
<td>12.40</td>
<td>0.63</td>
<td>0.48</td>
<td>-2.17</td>
<td>15.66</td>
<td>0.77</td>
<td>0.47</td>
<td>-2.12</td>
<td>15.70</td>
<td>0.77</td>
</tr>
<tr>
<td>7</td>
<td>19.55</td>
<td>0.69</td>
<td>0.51</td>
<td>7.64</td>
<td>13.99</td>
<td>0.66</td>
<td>0.52</td>
<td>7.44</td>
<td>12.70</td>
<td>0.60</td>
</tr>
<tr>
<td>8</td>
<td>16.86</td>
<td>0.64</td>
<td>0.42</td>
<td>-0.16</td>
<td>20.71</td>
<td>0.66</td>
<td>0.49</td>
<td>0.02</td>
<td>20.67</td>
<td>0.66</td>
</tr>
<tr>
<td>9</td>
<td>42.34</td>
<td>0.66</td>
<td>0.45</td>
<td>26.89</td>
<td>43.42</td>
<td>0.85</td>
<td>0.41</td>
<td>18.18</td>
<td>40.66</td>
<td>0.79</td>
</tr>
<tr>
<td>10</td>
<td>14.28</td>
<td>0.63</td>
<td>0.55</td>
<td>1.75</td>
<td>17.62</td>
<td>0.66</td>
<td>0.49</td>
<td>1.24</td>
<td>16.95</td>
<td>0.64</td>
</tr>
<tr>
<td>11</td>
<td>19.60</td>
<td>0.64</td>
<td>0.53</td>
<td>0.05</td>
<td>24.42</td>
<td>0.70</td>
<td>0.43</td>
<td>-0.20</td>
<td>24.09</td>
<td>0.69</td>
</tr>
<tr>
<td>12</td>
<td>37.70</td>
<td>0.70</td>
<td>0.38</td>
<td>27.37</td>
<td>48.94</td>
<td>0.88</td>
<td>0.33</td>
<td>15.75</td>
<td>42.51</td>
<td>0.76</td>
</tr>
<tr>
<td>13</td>
<td>11.76</td>
<td>0.66</td>
<td>0.38</td>
<td>2.75</td>
<td>11.75</td>
<td>0.76</td>
<td>0.29</td>
<td>1.73</td>
<td>11.07</td>
<td>0.71</td>
</tr>
<tr>
<td>14</td>
<td>9.28</td>
<td>0.67</td>
<td>0.28</td>
<td>-3.99</td>
<td>9.75</td>
<td>0.75</td>
<td>0.33</td>
<td>-3.78</td>
<td>9.13</td>
<td>0.70</td>
</tr>
<tr>
<td>15</td>
<td>12.50</td>
<td>0.66</td>
<td>0.51</td>
<td>11.43</td>
<td>20.89</td>
<td>0.63</td>
<td>0.58</td>
<td>11.38</td>
<td>20.82</td>
<td>0.63</td>
</tr>
<tr>
<td>16</td>
<td>17.75</td>
<td>0.72</td>
<td>0.43</td>
<td>-3.53</td>
<td>14.96</td>
<td>0.65</td>
<td>0.49</td>
<td>-2.92</td>
<td>14.34</td>
<td>0.62</td>
</tr>
<tr>
<td>17</td>
<td>19.09</td>
<td>0.69</td>
<td>0.43</td>
<td>-9.76</td>
<td>25.84</td>
<td>0.77</td>
<td>0.50</td>
<td>-8.36</td>
<td>25.19</td>
<td>0.74</td>
</tr>
<tr>
<td>18</td>
<td>17.09</td>
<td>0.69</td>
<td>0.47</td>
<td>15.78</td>
<td>19.47</td>
<td>1.21</td>
<td>0.38</td>
<td>15.00</td>
<td>18.75</td>
<td>1.17</td>
</tr>
<tr>
<td>19</td>
<td>41.75</td>
<td>0.59</td>
<td>0.60</td>
<td>14.54</td>
<td>37.58</td>
<td>0.68</td>
<td>0.58</td>
<td>14.54</td>
<td>37.57</td>
<td>0.68</td>
</tr>
<tr>
<td>20</td>
<td>17.05</td>
<td>0.72</td>
<td>0.37</td>
<td>11.45</td>
<td>18.06</td>
<td>0.89</td>
<td>0.34</td>
<td>11.43</td>
<td>18.04</td>
<td>0.89</td>
</tr>
<tr>
<td>21</td>
<td>14.78</td>
<td>0.6311</td>
<td>0.53</td>
<td>-2.56</td>
<td>18.73</td>
<td>0.86</td>
<td>0.19</td>
<td>-2.58</td>
<td>18.75</td>
<td>0.86</td>
</tr>
<tr>
<td>22</td>
<td>13.77</td>
<td>0.67</td>
<td>0.55</td>
<td>10.64</td>
<td>16.38</td>
<td>0.96</td>
<td>0.34</td>
<td>-6.13</td>
<td>14.60</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Mean: 18.33 | 0.67 | 0.46 | -1.88 | 20.89 | 0.78 | 0.43 | -1.00 | 19.92 | 0.75 | 0.46
StDe: 9.59 | 0.04 | 0.09 | 11.54 | 10.15 | 0.14 | 0.10 | 8.95 | 9.32 | 0.13 | 0.11
Table 4.6 with 2 weeks of training and 2 weeks of validation. For several of the subjects (e.g., subjects 2, 11, 15-17 and 19) the results under the model given by Equation 4.5 ($\hat{\eta}_t$) are excellent. However, there are a number of subjects where $r_{\text{fit}} < 0.2$ on the test set (e.g., subjects 4, 13, 14, 21 and 22). In terms of improving AAE, the use of Equation 4.9 ($\hat{y}_t$) show a significant but modest improvement in the $r_{\text{fit}}$ results in Tables 4.6 and 4.4. However, a more critical reason for the use of the soft senor equation given by Equation 4.9 is illustrated in Figure 3 for subject 10 based on the results in Table 4.6. The left plot shows the fit under Equation 4.5 and the right plot show the fit under Equation 4.9. As illustrated by the right plot, the use of measured glucose provides a correction within 5 minutes that is close to the most recently measured glucose. Since glucose does not change too rapidly most often for non-insulin dependent subjects, the sensor will reflect the variations in BGC behavior quite well in a continuous monitoring fashion.

The final set of results is given for 3 days of training under Equation 4.8 for food only. In addition, for all these subjects, $\zeta_i = 0.2$ and $\tau_{ai} = 0$. This was done to increase the degrees for freedom to estimate the more critical parameter $\tau_i$ and to simplify the optimization. The best choice for these values is future research work. As Table 4.6 shows, the results indicate that the model, while significantly worse than the best results in Table 4.6, are really quite promising for building the model with only 12 samples.

4.7 Concluding Remarks

This article presented preliminary work on the development of a virtual sensor for blood glucose concentration (BGC) with the objective of using it to develop a monitoring system that would be used by non-insulin dependent subjects. This device would require users to wear an armband that is widely used by this targeted group currently and manually entering meal sizes through the use of a button on the armband. This device would require
Table 4.4 Modeling results for the 22 test subjects with 1 week of training and 1 week of validation. AE and AAE values are in mg/dL. AE is not reported for Training since it is 0 for all subjects.

<table>
<thead>
<tr>
<th>Sub.</th>
<th>Equation 4.5 Model ( \hat{\eta} )</th>
<th>1 week Training</th>
<th>1 week Validation</th>
<th>2 week Testing</th>
<th>2 week Testing RAAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>( 0.51 )</td>
<td>17.2</td>
<td>0.52</td>
<td>17.2</td>
<td>10.4</td>
</tr>
<tr>
<td>5</td>
<td>( 0.74 )</td>
<td>0.32</td>
<td>-0.8</td>
<td>14.0</td>
<td>0.62</td>
</tr>
<tr>
<td>6</td>
<td>( 0.80 )</td>
<td>16.0</td>
<td>0.78</td>
<td>16.1</td>
<td>21.5</td>
</tr>
<tr>
<td>7</td>
<td>( 0.79 )</td>
<td>0.45</td>
<td>-4.8</td>
<td>17.7</td>
<td>0.67</td>
</tr>
<tr>
<td>8</td>
<td>( 0.73 )</td>
<td>0.32</td>
<td>-4.9</td>
<td>15.6</td>
<td>0.69</td>
</tr>
<tr>
<td>9</td>
<td>( 0.70 )</td>
<td>14.9</td>
<td>0.78</td>
<td>15.4</td>
<td>0.51</td>
</tr>
<tr>
<td>10</td>
<td>( 0.60 )</td>
<td>13.5</td>
<td>0.34</td>
<td>15.4</td>
<td>0.19</td>
</tr>
<tr>
<td>11</td>
<td>( 0.63 )</td>
<td>13.0</td>
<td>0.59</td>
<td>15.4</td>
<td>0.06</td>
</tr>
<tr>
<td>12</td>
<td>( 0.69 )</td>
<td>12.8</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>13</td>
<td>( 0.63 )</td>
<td>12.5</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>14</td>
<td>( 0.60 )</td>
<td>12.2</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>15</td>
<td>( 0.57 )</td>
<td>11.9</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>16</td>
<td>( 0.54 )</td>
<td>11.6</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>17</td>
<td>( 0.51 )</td>
<td>11.3</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>18</td>
<td>( 0.48 )</td>
<td>11.0</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>19</td>
<td>( 0.45 )</td>
<td>10.7</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>20</td>
<td>( 0.42 )</td>
<td>10.4</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>21</td>
<td>( 0.39 )</td>
<td>10.1</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
<tr>
<td>22</td>
<td>( 0.36 )</td>
<td>9.8</td>
<td>0.74</td>
<td>15.4</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Mean: 18.0, 0.61, 0.54, 19.4, 0.79, 0.51, 1.2, 22.5, 0.85, 0.38, 1.1, 21.9, 0.41, 0.37.
Table 4.5  Modeling results for the 22 test subjects with 3 days of training and no validation. AE and AAE values are in mg/dL.

<table>
<thead>
<tr>
<th>Sub.</th>
<th>Model from Equation 4.5</th>
<th>25 days testing</th>
<th>Model from Equation 4.9</th>
<th>25 days testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 days training</td>
<td></td>
<td>25 days testing</td>
<td></td>
</tr>
<tr>
<td>Sub.</td>
<td>AAE</td>
<td>RAEE</td>
<td>r&lt;sub&gt;fit&lt;/sub&gt;</td>
<td>AE</td>
</tr>
<tr>
<td>1</td>
<td>17.3</td>
<td>0.71</td>
<td>0.31</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>14.9</td>
<td>0.76</td>
<td>0.44</td>
<td>-0.1</td>
</tr>
<tr>
<td>3</td>
<td>8.0</td>
<td>0.70</td>
<td>0.23</td>
<td>-5.8</td>
</tr>
<tr>
<td>4</td>
<td>7.0</td>
<td>0.46</td>
<td>0.79</td>
<td>5.5</td>
</tr>
<tr>
<td>5</td>
<td>11.4</td>
<td>0.64</td>
<td>0.52</td>
<td>8.5</td>
</tr>
<tr>
<td>6</td>
<td>9.7</td>
<td>0.66</td>
<td>0.60</td>
<td>-3.2</td>
</tr>
<tr>
<td>7</td>
<td>29.6</td>
<td>0.74</td>
<td>0.35</td>
<td>16.5</td>
</tr>
<tr>
<td>8</td>
<td>8.2</td>
<td>0.66</td>
<td>0.39</td>
<td>-1.2</td>
</tr>
<tr>
<td>9</td>
<td>32.1</td>
<td>0.46</td>
<td>0.77</td>
<td>28.4</td>
</tr>
<tr>
<td>10</td>
<td>16.1</td>
<td>0.65</td>
<td>0.30</td>
<td>15.1</td>
</tr>
<tr>
<td>11</td>
<td>21.4</td>
<td>0.83</td>
<td>0.10</td>
<td>13.9</td>
</tr>
<tr>
<td>12</td>
<td>51.4</td>
<td>0.85</td>
<td>0.09</td>
<td>12.5</td>
</tr>
<tr>
<td>13</td>
<td>15.1</td>
<td>0.73</td>
<td>0.32</td>
<td>-7.5</td>
</tr>
<tr>
<td>14</td>
<td>9.3</td>
<td>0.71</td>
<td>0.51</td>
<td>2.9</td>
</tr>
<tr>
<td>15</td>
<td>18.3</td>
<td>0.67</td>
<td>0.50</td>
<td>13.1</td>
</tr>
<tr>
<td>16</td>
<td>17.5</td>
<td>0.63</td>
<td>0.67</td>
<td>-8.7</td>
</tr>
<tr>
<td>17</td>
<td>29.2</td>
<td>0.77</td>
<td>0.21</td>
<td>1.9</td>
</tr>
<tr>
<td>18</td>
<td>8.1</td>
<td>0.64</td>
<td>0.08</td>
<td>0.0</td>
</tr>
<tr>
<td>19</td>
<td>40.5</td>
<td>0.48</td>
<td>0.74</td>
<td>19.1</td>
</tr>
<tr>
<td>20</td>
<td>21.5</td>
<td>0.76</td>
<td>0.27</td>
<td>-4.0</td>
</tr>
<tr>
<td>21</td>
<td>15.6</td>
<td>0.58</td>
<td>0.66</td>
<td>-0.9</td>
</tr>
<tr>
<td>22</td>
<td>11.8</td>
<td>0.64</td>
<td>0.59</td>
<td>2.5</td>
</tr>
<tr>
<td>Mean</td>
<td>18.8</td>
<td>0.67</td>
<td>0.43</td>
<td>-4.5</td>
</tr>
<tr>
<td>StDe</td>
<td>11.5</td>
<td>0.11</td>
<td>0.22</td>
<td>9.9</td>
</tr>
</tbody>
</table>
frequent lancet measurements per day as current invasive continuous-time monitoring systems require. However, for this system, once the model is fully developed, which will likely require 2 to 4 weeks, lancet measurements will be less critical for accuracy and may not be necessary for some subgroups such as non-diabetic subjects.

Future work will involve running clinical studies under the protocol that subjects will follow when wearing the device such as time stamping for meal size and using only their glucose meter to collect data. If these studies show promise and continued improvement in the modeling technique, we hope to develop a prototype armband and evaluate it on several subjects.

While we have overcome many challenges such as the use of a food index, the lack of initial conditions, frequent and long term removal of the armband, and multiple input, subject-specific modeling under infrequent sampling, there are still several challenges to overcome. These challenges include obtaining starting values for parameters for estimation, gaining a better understanding on the bounds of each parameter, as well as the development of an
automatic estimation algorithm that will reside in the armband and apply the estimation method to develop the virtual sensor from on-line data. These are all areas of future research that we have begun to work to research and the results are quite promising.

4.8 Acknowledgments

This work was supported under the National Institute of Health through the Illinois Institute of Technology under Grant SA395-1125-6520. The authors would like to thank BodyMedia for funding much of the data collection and allowing us to use their equipment as well as Jeanne Stewart for assisting in data collection.

Appendix

The purpose of this appendix is to provide a mathematical proof that \( r_{\text{fit}} \), under the simple linear regression model given by Equation 4.8, does not depend on the model coefficients \( a_0 \) and \( a_i \) but only on the explanatory variable \( v_{i,t} \).

With \( \eta_t = a_0 + a_i v_{i,t} \), \( r_{\text{fit}} \) is mathematically given by

\[
\begin{align*}
    r_{\text{fit}} = r_{y,\eta} &= \frac{\sum_{j=1}^{n} (y_j - \bar{y})(\eta_j - \bar{\eta})}{\sqrt{\sum_{j=1}^{n} (y_j - \bar{y})^2} \cdot \sqrt{\sum_{j=1}^{n} (\eta_j - \bar{\eta})^2}} \\
    &= \frac{\sum_{j=1}^{n} (y_j - \bar{y}) (a_0 + a_i v_{i,j} - a_0 - a_i \bar{v}_i)}{\sqrt{\sum_{j=1}^{n} (y_j - \bar{y})^2} \cdot \sqrt{\sum_{j=1}^{n} (a_0 + a_i v_{i,j} - a_0 - a_i \bar{v}_i)^2}} \\
    &= \frac{a_i \sum_{j=1}^{n} (y_j - \bar{y}) (v_{i,j} - \bar{v}_i)}{a_i^2 \sqrt{\sum_{j=1}^{n} (y_j - \bar{y})^2} \cdot \sqrt{\sum_{j=1}^{n} (v_{i,j} - \bar{v}_i)^2}} \\
    &= \frac{a_1}{|a_1|} r_{y,v_{i}}
\end{align*}
\]

Thus, with \( a_i > 0 \), \( r_{\text{fit}} = r_{y,v_i} \) and for \( a_i < 0 \), \( r_{\text{fit}} = -r_{y,v_i} \). This result means that if the correlation \( r_{\text{fit}} \) between blood glucose concentration (BGC) and \( v_i \) is positive, \( a_1 \) can be set
to any positive value and $r_{\text{fit}}$ will depend only on the behavior of $v_i$ which is independently controlled by the values of the dynamic parameters associated with $v_i$. Conversely, if the correlation of BGC and $v_i$ is negative, $a_1$ can be set to any negative value and $r_{\text{fit}}$ will be greater than 0 and independently controlled by the values of the dynamic parameters associated with $v_i$. 
CHAPTER 5. A NEW METHOD FOR PREDICTING FUTURE BLOOD GLUCOSE CONCENTRATIONS IN NON-INSULIN DEPENDENT TYPE 2 DIABETICS

A paper to be submitted to Computers and Chemical Engineering

Lucas P. Beverlin, Derrick K. Rollins, Kaylee Kotz, Nisarg Vyas, David Andre, Greg Welk, and Warren Franke

Abstract

The ability to forecast the onset of hypoglycemia or hyperglycemia is very important for managing the blood glucose concentration (BGC) of diabetics. Currently, commercially available continuous glucose monitoring systems can only predict future BGC to warn the user of the onset of hypoglycemia or hyperglycemia up to 30 minutes into the future. The purpose of this work is to present a new methodology for constructing $(1 - \alpha)100\%$ forecast intervals for the prediction of BGC up to 1 hour before it would occur. In this work, a Wiener network is first used to predict BGC using food and activity variables as well as time of day. This model is then used with a $k$-steps ahead prediction (KSAP) model in order to predict BGC up to 1 hour into the future. This fitted KSAP model is used to construct $(1 - \alpha)100\%$ forecast intervals for one’s BGC up to 1 hour into the future. While in this work we forecast the BGC of non-insulin dependent type 2 diabetics, such
an approach could be used in conjunction with a model that utilizes insulin infusion to predict the future BGC of an insulin-dependent diabetic.

5.1 Introduction

Diabetes mellitus is a condition where the body’s ability to control its blood glucose concentration (BGC) has been impaired or destroyed. Unfortunately it is a growing problem throughout the world, as Zhang et al. [114] report that the global health expenditure for diabetes is projected to be at least 375 billion US dollars in 2010 and at least 500 billion US dollars in 2030.

There are two main types of diabetes mellitus. Both of them affect the body’s production of insulin, a hormone produced by the \( \beta \) cells of the pancreas that signals cells to uptake glucose for energy or fat storage. Type 1 diabetes mellitus is a condition where the \( \beta \) cells are destroyed by the body. What triggers the body to destroy the \( \beta \) cells is still being investigated, though it is believed to be an autoimmune response [21]. Type 2 diabetes mellitus is characterized by poor control of the body’s BGC. This can be caused by cells developing insulin resistance and/or the \( \beta \) cells’ inability to produce a sufficient amount of insulin to promote glucose uptake. Those with type 2 diabetes are at higher risk for several other health problems, such as obesity, high blood pressure, nephropathy, neuropathy and blindness [21]. While the onset of type 1 diabetes generally occurs before the age of 30 and type 2 diabetes after the age of 30, an increasing number of people, particularly of Pacific Island or south Asian descent are experiencing the onset of type 2 diabetes in their 20s [114].

It has been shown that tight glucose control significantly reduces the risk of complications in both type 2 diabetics [105] and in type 1 diabetics [72, 113]. To improve glucose control, type 2 diabetics have several options. Since many newly diagnosed type 2 diabet-
ics are overweight, altering dietary and exercise habits are typically the first prescription [5]. Many type 2 diabetics tend to consume an excess of carbohydrates at a meal, which can lead to overly high BGC, or hyperglycemia, as well as obesity. Exercise has also been shown to increase insulin sensitivity [81]. If the body cannot produce enough insulin to promote glucose uptake or if one’s insulin sensitivity has become too weak, then diet and exercise alone cannot tightly control BGC. In this case medications and/or insulin may be used to help control BGC.

To assess their current BGC, diabetics use a glucose meter. However, each measurement requires a finger prick, and thus monitoring one’s BGC constantly throughout the day using a glucose meter is unfeasible due to discomfort. Over the last fifteen years, diabetics have seen the advent and evolution of a continuous glucose monitoring system (CGMS). Typically these devices return predictions of BGC every five minutes. Once data have been downloaded from the device, diabetics can view a profile of the changes in their BGC over the course of several days. A new feature found on some newer CGMS models, such as the Minimed Guardian® REAL-Time System (Medtronic Minimed, Northridge, CA) [9], the SEVEN® PLUS System (DexCom, Inc. San Diego, CA) [3] and the 5-day FreeStyle Navigator (Abbott Laboratories, Abbott Park, IL) [109], is the ability to warn the user if he or she is currently hyperglycemic or hypoglycemic, which is the situation where one’s BGC is abnormally low. This allows the subject to take immediate action in order to return their BGC to a healthy level. However, to allow for better control, one needs to predict the onset of hypoglycemia or hyperglycemia before it occurs in order to take preventative measures. To our knowledge, two of these devices, the Minimed Guardian® and the 5-day FreeStyle Navigator, can predict a hypoglycemic or a hyperglycemic episode up to 30 minutes before it occurs. According to Bode et al. [9], for the Minimed Guardian®, this amount of time can be set by the user to be anywhere from 5 to 30 minutes in 5 minute intervals. However,
there is roughly a 10 minute lag in this device from collecting the sample to measure BGC and determining the level of BGC [53]. In order to truly predict BGC at a fixed amount of time into the future, this lag must be eliminated.

The purpose of this article is to present a new methodology for predicting future BGC. To this end a model that predicts future BGC from current measurements must be constructed. First the Wiener network will be used to predict one’s current BGC, with no lag, based on food and activity variables as well as time of day. Then this model will be used with a $k$-steps ahead prediction (KSAP) model in order to accurately predict BGC in the future. Finally a $(1 - \alpha)100\%$ prediction interval for one’s BGC at a given time in the future will be constructed. Such an interval henceforth will be referred to as a forecast interval, since we are interested in the prediction of a response variable at some point in the future, though Chatfield [17] uses the term interval forecast.

The proposed methodology is presented with the following outline. First a detailed description of the Wiener network for multiple inputs and a single output is given to establish the problem context. After this section a description of the $k$-steps ahead prediction (KSAP) model is given and methods for determining a $(1 - \alpha)100\%$ forecast interval for the response variable in such a model is given. The details of the methodology and an example to illustrate our methodology’s ability to forecast BGC is given in the fifth section. Concluding remarks and some ideas for future work are given in the last section.

5.2 The Wiener Network

The Wiener network has a powerful structure for modeling nonlinear dynamic systems. A block diagram with $p$ inputs and one output is given in Figure 5.1. Each input $x_i$ is first passed through a dynamic linear block, denoted $g(x_i)$ and converted into its corresponding dynamic variable $v_i$. In Rollins et al. [86], the following second-order-plus-lead with dead
Figure 5.1  A graphical representation of the Wiener model used in Rollins et al. [86].

\[
\begin{align*}
&x_1 \rightarrow g(x_1) \rightarrow v_1 \\
&x_2 \rightarrow g(x_2) \rightarrow v_2 \rightarrow v \rightarrow f(v) \rightarrow \hat{y} \\
&\vdots \\
&x_p \rightarrow g(x_p) \rightarrow v_p
\end{align*}
\]

time differential equation is used:

\[
\tau_i^2(t, x_i) \frac{d^2 v_i}{dt^2}(t) + 2\tau_i(t, x_i)\zeta_i(t, x_i) \frac{dv_i}{dt}(t) + v_i(t, x_i) = \tau_{ai}(t, x_i) \frac{dx_i}{dt}(t - \theta_i) + x_i(t - \theta_i),
\]

(5.1)

where \(\tau_i\) is a time constant, \(\zeta_i\) is a damping coefficient, \(\tau_{ai}\) is a lead coefficient, \(\theta_i\) denotes dead time, and \(t\) represents time. For simplicity, assume that the dynamic parameters are time and space invariant, i.e. \(\tau_i(t, x_i) = \tau_i\) and \(\zeta_i(t, x_i) = \zeta_i\). Also \(v_i(t, x_i)\) will be written as \(v_i(t)\) henceforth. The vectors \(\tau_a\), \(\tau\), and \(\zeta\) will denote all lead coefficients, time constants, and damping coefficients, respectively.

In order to calculate \(v_i(t)\), a recursive definition for \(v_i(t)\) must be determined. This can be done by using a backward difference approximation for the derivatives in Equation 5.1:

\[
\frac{dv_i}{dt}(t) \approx \frac{v_i(t) - v_i(t - \Delta t)}{\Delta t}
\]

(5.2)
and
\[ \frac{d^2 v_i}{dt^2}(t) \approx \frac{v_i(t) - 2v_i(t - \Delta t) + v_i(t - 2\Delta t)}{\Delta t^2} \tag{5.3} \]

It should be noted that since an instantaneous change in \( x_i \) at time \( t \) cannot result in an instantaneous change in \( v_i \) at time \( t \) under this structure. Thus one must approximate \( x_i(t - \theta_i) \) with \( x_i(t - \theta_i - \Delta t) \) and \( \frac{dx_i}{dt}(t - \theta_i) \) with \( x_i(t - \theta_i - \Delta t) \) and \( x_i(t - \theta_i - 2\Delta t) \):
\[ \frac{dx_i}{dt}(t - \theta_i) \approx \frac{x_i(t - \theta_i - \Delta t) - x_i(t - \theta_i - 2\Delta t)}{\Delta t}. \tag{5.4} \]

By substituting Equations 5.2 - 5.4 into Equation 5.1,
\[ v_i(t) = \frac{2\tau_i^2 + 2\tau_i \zeta_i \Delta t}{\tau_i^2 + 2\tau_i \zeta_i \Delta t + \Delta t^2} v_i(t - \Delta t) - \frac{\tau_i^2}{\tau_i^2 + 2\tau_i \zeta_i \Delta t + \Delta t^2} v_i(t - 2\Delta t) \]
\[ + \frac{\tau_{ai} \Delta t + \Delta t^2}{\tau_i^2 + 2\tau_i \zeta_i \Delta t + \Delta t^2} x_i(t - \Delta t - \theta_i) - \frac{\tau_{ai} \Delta t}{\tau_i^2 + 2\tau_i \zeta_i \Delta t + \Delta t^2} x_i(t - 2\Delta t - \theta_i) \tag{5.5} \]

The resulting \( v_i \)'s are then passed through a static nonlinear block, denoted \( f(v) \) in Figure 5.1. The function \( f \) is typically a nonlinear function with respect to the \( v_i \)'s. Rollins et al. [86] uses 11 inputs and the nonlinear function
\[ f(v_1(t), \ldots, v_{11}(t)) = a_0 + \sum_{i=1}^{11} a_i v_i(t) + \sum_{j=1}^{11} b_j v_j^2(t) + \sum_{k=1}^{10} \sum_{l=k+1}^{11} c_{k,l} v_k(t) v_l(t). \tag{5.6} \]

In this work, the quadratic and second-order interaction terms will be omitted, and three of the inputs used in their work will be omitted here, thus leaving the static block to be represented by a linear regression model. Hence the predictions from the Wiener network are given by
\[ \hat{\eta}(t) = \hat{a}_0 + \sum_{i=1}^{8} \hat{a}_i v_i(t|\hat{\tau}_{ai}, \hat{\tau}_i, \hat{\zeta}_i). \tag{5.7} \]

Let \( a \) denote the vectors corresponding to all linear static parameters. They will be collectively referred to as the static parameters. Thus the final resulting model is
\[ y(t) = \eta(t) + \epsilon(t), \tag{5.8} \]
where $\epsilon(t)$ is a normally distributed error term with mean 0 and variance $\sigma^2$. The error terms are assumed to be independent of one another.

The ability of the Wiener network to assign unique dynamic behavior to each input is a major advantage it has over other models that have been used to model BGC. While autoregressive models have been used to predict BGC [83, 96], these models only depend on previous values of BGC. These models do not utilize inputs that could greatly impact BGC, such as the amount of carbohydrates consumed. Other models that do allow for an exogenous input that have been used to predict BGC include autoregressive models with exogenous variables (ARX models) [28, 79], autoregressive moving average models with exogenous inputs (ARMAX models) [25, 27], and nonlinear ARMAX (NARMAX) models that are linear in their parameters [99]. Since all three of these models share the same shortcoming, we will illustrate this by considering a typical ARX model with two inputs, each with lag 2.

$$y_m = \alpha_0 + \alpha_1 y_{m-1} + \alpha_2 y_{m-2} + \beta_{11} x_{1,m-1} + \beta_{12} x_{1,m-2} + \beta_{21} x_{2,m-1} + \beta_{22} x_{2,m-2} + \epsilon_m, \quad (5.9)$$

where $m$ denotes a timepoint corresponding to time $t$. For simplicity, the error term $\epsilon_m$ will be suppressed, as it is not important in the following derivation. Thus the model can be written in the form of a single differential equation:

$$c_1 \frac{d^2 y}{dt^2}(t) + c_2 \frac{dy}{dt}(t) + y(t) = d_1 \frac{dx_1}{dt}(t) + d_2 x_1(t) + d_3 \frac{dx_2}{dt}(t) + d_4 x_2(t), \quad (5.10)$$

where each $c_i, i = 1, 2$ and $d_i, i = 1, 2, 3, 4$, are constants. Note that the coefficient in front of $y(t)$ in Equation 5.10 is set to 1 so that the number of constants is equal to the number of parameters in the ARX model without the intercept. This can be seen by recalling the backward difference approximations for the first and second derivatives given in Equations 5.2, 5.3 and 5.4. Here however we will assume that $\theta_i = 0$. Substituting these as well as
\(x_i(t - \Delta t)\) for \(x_i(t)\) into Equation 5.10,

\[
\frac{c_1 y(t) - 2y(t - \Delta t) + y(t - 2\Delta t)}{\Delta t^2} + \frac{c_2 y(t) - y(t - \Delta t)}{\Delta t} + y(t) = \frac{d_1 x_1(t - \Delta t) - x_1(t - 2\Delta t)}{\Delta t} + d_2 x_1(t - \Delta t) + \frac{d_3 x_2(t - \Delta t) - x_2(t - 2\Delta t)}{\Delta t} + d_4 x_2(t - \Delta t)
\]

(5.11)

To simplify notation, we will rewrite this to use subscripts to denote timepoints, i.e. \(y(t) = y_m, y(t - \Delta t) = y_{m-1}\):

\[
\frac{c_1 y_m - 2y_{m-1} + y_{m-2}}{\Delta t^2} + \frac{c_2 y_m - y_{m-1}}{\Delta t} + y_m = \frac{d_1 x_{1,m-1} - x_{1,m-2}}{\Delta t} + d_2 x_{1,m-1} + \frac{d_3 x_{2,m-1} - x_{2,m-2}}{\Delta t} + d_4 x_{2,m-1}
\]

(5.12)

Now rearranging terms and solving for \(y_m\) yields

\[
\left(\frac{c_1}{\Delta t^2} + \frac{c_2}{\Delta t} + 1\right) y_m - \left(\frac{2c_1}{\Delta t^2} + \frac{c_2}{\Delta t}\right) y_{m-1} + \frac{c_1}{\Delta t^2} y_{m-2} = \left(\frac{d_1}{\Delta t} + d_2\right) x_{1,m-1} - \frac{d_1}{\Delta t} x_{1,m-2} + \left(\frac{d_3}{\Delta t} + d_4\right) x_{2,m-1} - \frac{d_3}{\Delta t} x_{2,m-2}
\]

(5.13)

\[
(c_1 + c_2 \Delta t + \Delta t^2) y_m - (2c_1 + c_2 \Delta t) y_{m-1} + c_1 y_{m-2} = (d_1 \Delta t + d_2 \Delta t^2) x_{1,m-1} - d_1 \Delta t x_{1,m-2} + (d_3 \Delta t + d_4 \Delta t^2) x_{2,m-1} - d_3 \Delta t x_{2,m-2}
\]

(5.14)

\[
y_m = \frac{2c_1 + c_2 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-1} - \frac{c_1}{c_1 + c_2 \Delta t + \Delta t^2} y_{m-2} + \frac{d_1 \Delta t + d_2 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} x_{1,m-1} - \frac{d_1 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} x_{1,m-2} + \frac{d_3 \Delta t + d_4 \Delta t^2}{c_1 + c_2 \Delta t + \Delta t^2} x_{2,m-1} - \frac{d_3 \Delta t}{c_1 + c_2 \Delta t + \Delta t^2} x_{2,m-2}
\]

(5.15)

One can see that the denominator of each coefficient of an input is \(c_1 + c_2 \Delta t + \Delta t^2\). Thus the only parameter that determines the relationship between \(x_i\) and \(y\) are the \(d_i\)’s. This is very constritive, in that if we set \(c_1 = \tau^2\) and \(c_2 = 2\tau\zeta\), this forces the residence time of each input to be \(2\tau\zeta\). Thus to use an ARX, ARMAX, or NARMAX model for predicting
BGC, one must assume that each input to be used in the model has the same residence time. Given there is some evidence that this is untrue \cite{70}, we would prefer to use a model that does not have this limitation. This model flexibility as well as low parameterization were the main reasons the Wiener network is used in this work.

5.3 The KSAP Model

In order to predict future BGC, a $k$-steps ahead prediction (KSAP) model will be used, as given in Rollins et al. \cite{86}. A step in this situation corresponds to a change in time of $\Delta t$ minutes. In this work $\Delta t = 5$ since the measurements of each input and output are reported every five minutes. Thus, for example, a 6 steps ahead prediction model predicts BGC thirty minutes after the current measurements have been taken.

In order to motivate the KSAP model, a prewhitening model is defined as (Rollins, et al. \cite{85})

\begin{equation}
    y_t = \eta_t + N_t,
\end{equation}

where

\begin{equation}
    N_t = \frac{\theta_q(B)}{\phi_p(B)} a_t,
\end{equation}

\begin{align}
    \theta_q(B) &= 1 - \theta_1 B^1 - \theta_2 B^2 - \cdots - \theta_q B^q, \\
    \phi_p(B) &= 1 - \phi_1 B^1 - \phi_2 B^2 - \cdots - \phi_p B^p,
\end{align}

$B$ is a backward shift operator, i.e. $B^r x_t = x_{t-r}$ for $r = 1, 2, \ldots$, and $a_t$ is an error term at time $t$ whose mean is 0 and variance is $\sigma_a^2$ where $a_i$ is independent of $a_j$ for $i \neq j$. Note that $N_t$ is an ARMA($p, q$) noise term. Let

\begin{equation}
    \frac{\phi_p(B)}{\theta_q(B)} = \Pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \cdots
\end{equation}

\begin{equation}
    \end{equation}
Multiplying both sides of Equation 5.16 by Π(B) yields

\[ \Pi(B)y_t = \Pi(B)\eta_t + a_t. \]  

(5.19)

\[ (1 - \pi_1 B - \pi_2 B^2 - \cdots) y_t = (1 - \pi_1 B - \pi_2 B^2 - \cdots) \eta_t + a_t \]  

(5.20)

\[ y_t = \eta_t + (\pi_1 B + \pi_2 B^2 + \cdots)(y_t - \eta_t) + a_t \]  

(5.21)

\[ y_t = \eta_t + (\Pi(B) - 1)e_t + a_t, \]  

(5.22)

where \( \hat{\Pi}(B) = \hat{\phi}(B) \hat{\theta}(B) \) and \( e_t = y_t - \eta_t \). Thus from Equation 5.22 a one step ahead prediction (OSAP) estimate for \( y_t \) is given by

\[ \hat{y}_t = \hat{\eta}_t + (\hat{\Pi}(B) - 1)\hat{e}_t = \hat{\eta}_t + \left( \frac{\hat{\phi}(B)}{\theta(B)} - 1 \right) \hat{e}_t, \]  

(5.23)

where \( \hat{\Pi}(B) = \hat{\pi}_1 B + \hat{\pi}_2 B^2 + \cdots \) and \( \hat{e}_t = y_t - \hat{\eta}_t \).

The KSAP model will have a similar structure to the OSAP model. Instead of predicting one timestep ahead, i.e. at \( t \), the KSAP model predicts BGC at timestep \( t + k - 1 \). Note that a KSAP model predicts \( y_{t+k-1} \) rather than \( y_{t+k} \) since \( y_{t+k-1} \) is \( k \) steps ahead of the most recent BGC measurement available, which is at timestep \( t - 1 \). With the most recent output \( y_{t-1} \) the KSAP model proposed here gives the following prediction for \( y_{t+k-1} \)

\[ \hat{y}_{t+k-1} = \hat{\eta}_{t+k-1} + (\hat{\Pi}^*(B) - 1)\hat{e}_t \]  

(5.24)

with model

\[ y_{t+k-1} = \eta_{t+k-1} + N^*_t + \beta_{t+k-1} \]  

(5.25)

where \( N^*_t = \Pi^*(B)a_t, \Pi^*(B) = \frac{\phi^*(B)}{\theta^*(B)} = 1 - \pi_1^* B - \pi_2^* B^2 - \cdots, \pi_i^* \) are parameters to be estimated, \( \hat{\Pi}^*(B) = 1 - \hat{\pi}_1^* B - \hat{\pi}_2^* B^2 - \cdots, \hat{\pi}_i^* \) are estimates of \( \pi_i^* \), and \( \beta_{t+k-1} \) is an estimate of the bias at timestep \( t + k - 1 \). This bias term is included because the residuals from a KSAP model will not be white noise. This is due to the serial correlation that is present
in \(\hat{\Pi}(B) - 1\) that would not be present in \(\hat{\Pi}(B) - 1\) from the OSAP model. To eliminate this bias, an intercept term \(\phi_0\) is added to this model and estimated, i.e.

\[
y_{t+k-1} = \eta_{t+k-1} + N^*_t + \phi_0. \tag{5.26}
\]

There are advantages to using a KSAP model, which shall be referred to as model 2, as opposed to a Wiener network that does not use outputs, which shall be called model 1, or a model that only uses outputs, such as an autoregressive model, which shall be denoted model 3. The first advantage is that model 2 incorporates both inputs and outputs in order to improve prediction. Model 1 only uses inputs to predict BGC, while model 3 only uses outputs. Secondly, the outputs used in Model 1 can assist in correcting for a lack of fit that is inherent to modeling BGC in non-insulin dependent diabetics. Third, the predictive ability of model 2 far outperforms that of the autoregressive model, which can be seen in Figure 5.2. As \(k\) increases, the predictive ability of model 2 decreases, but its performance will not be worse than that of model 1, while the predictive ability of model 3 degrades quickly as \(k\) increases.

### 5.4 The Proposed Methodology

In this section the proposed methodology for constructing \((1 - \alpha)100\%\) forecast intervals for BGC will be presented. To illustrate this methodology, data collected from two subjects, denoted subjects 1 and 2 in this work, in a study in Beverlin et al. [8] were used. In that work, the Medtronic MiniMed Continuous Glucose Monitoring CGMS® System Gold™ (Medtronic Minimed, Northridge, CA) was used to measure BGC every 5 minutes. The SenseWear® Pro3 Body Monitoring System (BodyMedia, Inc., Pittsburgh, PA) was used to measure the activity variables that were used by the Wiener model every five minutes, and subjects recorded their food data using Weightmania® Pro software (Edward
Figure 5.2 The observed relationship between $k$, the use of inputs and/or outputs in the model and $r_{fit}$. Taken from [86].

A. Greenwood, Inc., Cambridge, MA) on a PDA. The inputs used by the Wiener network is given in Table 5.1. The Wiener networks were fit in the same manner as presented in that paper. The KSAP models were fit to the final two weeks of the data for each subject. Predictions at $k = 1, 6$ and 12 steps ahead will be analyzed.

First, since $N_t^*$ is an infinite sequence, it should be truncated in order to limit the number of parameters to be estimated. To do this, models were fitted with 1, 2, 3, …, terms, i.e.

\begin{align}
    y_{t+k-1} &= \eta_{t+k-1} + \phi_0 + \phi_1 e_{t-1} + a_t, \\
    y_{t+k-1} &= \eta_{t+k-1} + \phi_0 + \phi_1 e_{t-1} + \phi_2 e_{t-2} + a_t,
\end{align}

and so on. Terms were added until a model was fit where at least one $\phi_i$ failed to be significantly different from zero. The parameter $\phi_i$ was deemed significant if, for the hypothesis test $H_0 : \phi_i = 0$ versus $H_a : \phi_i \neq 0$, the value of $\frac{\hat{\phi}_i}{\sigma_{\phi_i}}$, which is assumed to be a random sample from a standard normal distribution, is larger than 1.96, which corresponds
Table 5.1  A table of inputs used in the modeling work of Rollins et al. [86] Those in bold will be used in this work, and the number in parentheses denotes the input number, i.e. \( v_1 \) corresponds to carbohydrates.

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Transverse accel.-peaks (4)</td>
</tr>
<tr>
<td></td>
<td>Energy expenditure (7)</td>
</tr>
<tr>
<td></td>
<td>Heat flux (5)</td>
</tr>
<tr>
<td></td>
<td>Transverse acceleration - MAD (6)</td>
</tr>
<tr>
<td>Food</td>
<td>Carbohydrates (1)</td>
</tr>
<tr>
<td></td>
<td>Fat (2)</td>
</tr>
<tr>
<td></td>
<td>Protein (3)</td>
</tr>
<tr>
<td>Circadian</td>
<td>Time of day (8)</td>
</tr>
</tbody>
</table>

to \( p < .05 \) for this test. The last model where all \( \phi_i \)'s were deemed significant was taken as the model of choice. The number of error terms used for each subject and each \( k \) are given in Table 5.2. For every model, either two or three error terms were used for KSAP modeling.

When fitting the resulting KSAP model for each \( k \) to subject \( i \), the residuals do appear to be skewed from a normal distribution with mean 0 and some variance \( \sigma_{i,k}^2 \), and the skewness appears to increase as \( k \) increases. Histograms of the residual plots are given in Figure 5.3. To construct an approximate \((1 - \alpha)100\%\) forecast interval, we will use a similar calculation as a calculation of a \((1 - \alpha)100\%\) prediction interval in nonlinear regression, but the critical value that is typically used will be increased to account for the skewness. To further simplify the calculation of this forecast interval, the large sample size allowed the dynamic parameters to be accurately estimated. Due to the low variance estimates of the dynamic parameters, it was assumed in the calculation of the forecast interval that the dynamic parameters were constants. This allows the dynamic variables found in the Wiener network to be used as regressors, as the inputs were also assumed to be measured without error. Thus an errors-in-variable approach, as given in Fuller [35] or Seber and Wild [90], or factoring in the variance of the dynamic parameters to the forecast interval,
is not necessary.

From this, an approximate \((1-\alpha)100\%\) forecast interval, conditional on \(v_{t+k-1}, e_{t-1}, \dotsc, e_{t-s}\) for a single future observation \(Y_{i,t+k-1}\) from a KSAP model can be constructed as

\[
\hat{y}_{i,t+k-1} \pm t_{\alpha/2,n-p}\hat{\sigma}_{i,k} \sqrt{1 + c_{i,k}}, 
\]

(5.29)

where \(p\) is the number of static parameters in the model, \(t_{\alpha/2,n-p}\) is the \((1 - \alpha/2)100\%\) percentile of a \(t_{n-p}\) distribution, \(c_{i,k} = w'_{i,t+k-1}(V'V)^{-1}w_{i,t+k-1},\)

\[
w'_{i,t+k-1} = [v_{1,t+k-1} v_{2,t+k-1} \dotsc v_{p,t+k-1} e_{t-1} \dotsc e_{t-q}], 
\]

(5.30)

\[
V = [1_n v_1 \dotsc v_p e_1 \dotsc e_q], 
\]

(5.31)

\(q\) is the number of terms in the KSAP model (either 2 or 3), \(1_n\) is a vector of length \(n\) whose entries are 1, \(n\) is the number of observations used to fit the model, \(e_i\) is the vector of residuals of lag \(i\), and \(\hat{\sigma}_{i,k}\) is the point estimate of the variance of \(Y_{i,t+k-1}\). Since we are interested in obtaining a simultaneous confidence band for all values in the range of the dynamic variables and residuals, we chose to use a confidence band approach, as presented in Bates and Watts [6]. Thus, for the remainder of this work, an approximate \((1-\alpha)100\%\) forecast interval for \(Y_{i,t+k-1}\) is

\[
\hat{y}_{i,t+k-1} \pm \sqrt{pF_{p,n-p,\alpha}}\hat{\sigma}_{i,k} \sqrt{1 + c_{i,k}}, 
\]

(5.32)

where \(F_{p,n-p,\alpha}\) is the \((1 - \alpha)100\%\) percentile of an \(F_{p,n-p}\) distribution.

In having a large sample size, \(c_{i,k}\) is also so small as to be negligible. This is seen in Table 5.2, in that the coverage of the 95% forecast intervals without \(c_{i,k}\), denoted Coverage 1, are almost identical to those with \(c_{i,k}\), given under Coverage 2. This can also be seen in that we obtain highly accurate estimates of the static parameters due to the large sample size. Thus the estimated variance of the static parameters \(\hat{\sigma}^2(V'V)^{-1}\) is negligible. Hence
to simplify calculations, one may remove \( c_{i,k} \) from Equation 5.32, which yields the following formula for a \((1 - \alpha)100\%\) forecast interval:

\[
\hat{y}_{i,t+k-1} \pm \sqrt{p F_{p,n-p,\alpha} \hat{\sigma}_{i,k}}.
\]  

(5.33)

It should be noted that a similar approach was taken to calculate \((1 - \alpha)100\%\) confidence intervals for individual \( Y_{i,t+k-1} \) in Gilchrist [39]. Chatfield [16] notes that Gilchrist’s method may not achieve the desired covered for small \( n \) and large \( k \). For this work, \( n \) is very large, typically over 3000, and \( k \) is no larger than 12. Thus achieving the desired coverage seems plausible.

Plots of the models where predictions are made before and after a meal are given in Figure 5.4. Not surprisingly, the forecast intervals become wider as \( k \) increases. While the coverage is not perfect, one can see that the observed BGC that fall outside of the 95% forecast interval are only slightly higher than the upper bound of the forecast interval. It would be far more worrisome if some observed BGC fell well below the lower bound of the 95% forecast interval, as the health implications of hypoglycemia are diabetic shock, coma, and even death. This does not appear to happen throughout the two weeks of predictions, regardless of \( k \).

5.5 Concluding Remarks

This article presents a methodology for constructing \((1 - \alpha)100\%\) forecast intervals for BGC up to 1 hour into the future. This methodology first utilizes a Wiener network to predict BGC. This fitted model is then used in a KSAP model in order to improve the accuracy of predictions. Finally a \((1 - \alpha)100\%\) forecast interval is determined from the KSAP model in order to predict future BGC.
Figure 5.3 Histograms of the residuals for 1, 6, and 12 steps ahead for two subjects.

One weakness of this approach is that if one is predicting BGC too far into the future, there is no way for this model to take into account food consumption or exercise that occurs between the current time and the time for the prediction. In this work, since all the food and activity data were collected prior to fitting the model, it was known when meals and exercise would take place, and thus our model could account for these future events. If future meals and exercise are unknown, then the performance of these models, particularly as $k$ increases, will suffer.
Table 5.2  Results from the calculation of forecast intervals. Terms is the number of error terms used in the KSAP model for predicting BGC \( k \) steps ahead for subject \( i \), Coverage denotes the proportion of observations \( y_{t+k-1} \) contained in the 95% forecast intervals calculated from Equation 5.32.

<table>
<thead>
<tr>
<th>Subject 1</th>
<th>Subject 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>Terms</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

While this work was performed on data taken from non insulin-dependent type 2 diabetics, such a method could be used, in conjunction with a model that incorporates insulin infusion, for predicting BGC in diabetics who are insulin dependent. For these people, this KSAP model that utilizes insulin infusion could warn them ahead of time if their BGC will become too low or too high. From this, they could take countermeasures to prevent hypoglycemia or hyperglycemia from even occurring. This would be very useful for type 1 diabetics who are concerned with abnormal BGC during the overnight hours, as an alarm could sound to wake them in order to allow them to take corrective action.

Unfortunately, for this data, assessing the ability of this approach to detect hypoglycemia is difficult. If the signal from the sensor to the CGMS is weak, this can result in the CGMS reporting lower BGC than what the actual BGC is. This can result in the CGMS incorrectly reporting hypoglycemia due to weak signal strength. This also causes error in the model because of the error in the observed BGC from the CGMS. While data cleaning was performed to remove erroneous BGC measurements from the CGMS, there were few observations of hypoglycemia in the data used by Beverlin et al. [8]. A further complaint raised in the literature [20, 37, 57] is that while the accuracy of a CGMS has
increased over the past 10 years, they are still not as accurate as lancet glucose meters, which can only give infrequent BGC measurements.

The ultimate goal of this research is to develop a method of forecasting future BGC for use in a device that measures BGC noninvasively that can warn the user up to an hour ahead of time of a possible hypoglycemic or a hyperglycemic episode. Such a device, particularly if false alarms can be minimized, would improve the quality of life for diabetics everywhere by allowing the user to take corrective action to reduce the occurrence of
abnormal levels of BGC. While the results in this work are promising, the forecast intervals for large $k$ must become narrower in order to be used effectively in such a device.

5.6 Acknowledgments

This work was supported by the Juvenile Diabetes Research Foundation, Grant 5-2010-7. The authors would like to thank BodyMedia for funding much of the data collection and allowing us to use their equipment and Jeanne Stewart for assisting in data collection.
CHAPTER 6. CONCLUDING REMARKS AND FUTURE WORK

6.1 Concluding Remarks

Many people have attempted to model one’s blood glucose concentration (BGC) over the last 50 years. While several of these models have had success, there are strengths and weaknesses associated with each model. The model used in this work, the Wiener network, has many strengths that other models do not have. The first is that the dynamic parameters for each input of the Wiener network have phenomenological meaning. The second is that the Wiener network assigns unique dynamics to each input. This is important in modeling BGC, as food nutrients, activity, and time of day all affect BGC differently. Many other models that have been used do not utilize inputs, and of those that have, none to our knowledge use all the inputs that the Wiener network in this work uses.

One of the weaknesses of the Wiener network is that it is a nonlinear model. Thus iterative methods will be needed in order to fit the Wiener network. In this work an algorithm was developed that could determine accurate parameter estimates in order to create an accurate model in a short period of time. This algorithm can also determine which inputs should be included in the Wiener network to maximize the correlation between the predicted BGC from the Wiener network and the observed BGC. Thus this algorithm was used to produce models that could accurately predict a non-insulin dependent type 2 diabetic’s BGC.
Although the Wiener network was used to accurately predict BGC in non-insulin dependent type 2 diabetics, there are other hurdles that must be overcome in order to use such a model in a device that can noninvasively measure one’s BGC. One hurdle is that there may be model bias. To reduce this bias and increase its accuracy, the model is calibrated four times a day using a measurement of BGC from a lancet glucose meter. Another is the inclusion of food in the model. A timestamping procedure was devised in order to approximate food intake rather than measure it to the nearest gram as was done for the models fitted using the algorithm described in Chapter 3. The final hurdle is that most users will not frequently measure their BGC. Due to a lack of data, some parameters were set to fixed values to reduce parameterization in order to determine an accurate model with a very limited amount of data. With these hurdles overcome, the Wiener network was able to accurately monitor BGC throughout the day with only four weeks of infrequent data collection.

Another function that could be utilized in such a noninvasive device is the ability to predict a user’s BGC in the future. If the user has a desired range for their BGC, he or she could set an alarm to sound if predictions of future BGC indicate that their BGC may leave the specified range. The Wiener network is used with a $k$-steps ahead prediction model to improve prediction of future BGC. This allows us to use both inputs and outputs in order to improve prediction. By utilizing this model, this work developed an approximate $(1 - \alpha)100\%$ forecast interval for BGC up to 1 hour into the future.

### 6.2 Future Work

While the results of this work will bring us closer to the goal of developing a device that can noninvasively measure BGC accurately, there is still much work to be done. The first focus is to improve the accuracy of our model. While the models fit in this work yield
predictions of BGC that exhibit high correlation with observed BGC, there may be some measurable inputs that are not being utilized in these models. Identifying such inputs will increase the ability of these models to track changes in BGC and reduce the average absolute error. It is also possible that a better model structure that uses current inputs may be identified.

In this work, the Wiener network has been used to accurately monitor one’s BGC under infrequent measurements of BGC. However, there are several possible improvements that could be made. Firstly, these models were fit manually. To be used in the noninvasive device for measuring BGC, an algorithm must be devised to determine accurate parameter estimates with a small amount of data. Thus certain parameters may need to be fixed in order for the algorithm to find a model that accurately predicts BGC. Determining starting parameter estimates that allow the algorithm to quickly find parameter estimates that yield an accurate model is another topic of future research.

Another focus on future research is to improve the forecasting capability of the KSAP model. This capability could allow it to be used to warn the user of the onset of hypoglycemia or hyperglycemia in the near future. While the KSAP model produced narrow forecast intervals when \( k \) was small, the width of the forecast interval increased rapidly as \( k \) increased. Thus, currently predicting the onset of hypoglycemia or hyperglycemia an hour or more ahead of time is difficult, as the width of the forecast interval for large \( k \) is too wide to make a sound judgment on determining whether or not one will have abnormal BGC at that future time. Modifications to the KSAP model may be necessary in order to improve performance.

An advantage that this work had that may not be available under free-living conditions is the knowledge of when future meals will be consumed or future exercise will take place. In this work, since all data was collected before constructing the forecast intervals, it was
known if the user would consume a meal or exercise within the next hour, and thus it was taken into account in predicting future BGC. This information may not be available to the model used in this noninvasive device under free-living conditions. Thus functionality must be added to the device that allows the user to warn the device of future food consumption or exercise, as simply increasing the width of the forecast interval to account for this uncertainty would decrease the ability of the device to accurately predict future hypoglycemia or hyperglycemia.

This work represents progress toward the ultimate goal of the development of a device that noninvasively predicts BGC in non-insulin dependent people. Such a device would be useful to anyone who desires to learn more about how their BGC is impacted by their diet and exercise regimen as well as time of day. This device would improve the well-being of anyone who wears such a device.
Bibliography


