UNIVERSITY OF CALIFORNIA
RIVERSIDE

Simultaneous State and Parameter Estimation of Glucose Metabolism in Type 1 Diabetic Subjects.

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Mechanical Engineering

by

Vashishtha Janakkumar Bhatt

June 2019

Thesis Committee:

Dr. Thomas Stahovich, Chairperson

Dr. Fabio Pasqualetti

Dr. Mona Eskandari

The Thesis of Vashishtha Janakkumar Bhatt is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

ABSTRACT OF THE THESIS

Simultaneous State and Parameter Estimation of Glucose Metabolism
In Type 1 Diabetes Subjects

by

Vashishtha Janakkumar Bhatt

Master of Science, Graduate Program in Mechanical Engineering
University of California, Riverside, June 2019
Dr. Thomas Stahovich, Chairperson

Approximately 1.5 million people live with type 1 diabetes in the US alone. The chronic disease must be managed by adopting a strict glycemic control regimen. We have developed a prediction algorithm to aid in bolus calculation and meal planning. By continuously estimating the parameters of a physiological metabolism model, the algorithm can predict blood glucose concentrations 30-minutes in advance. Such a prediction system allows a subject to prevent hypoglycemic episodes. The algorithm uses continuous glucose monitor (CGM) measurements along with meal and insulin inputs to compute the best fit on a patient-specific model. The prediction accuracy is evaluated by computing the root mean squared error (RMSE) between predicted 30-minute concentration and the actual value. An average RMSE value of 18 mg/dl is achieved on the datasets. Also, over 83% of the predictions are within zone A of the Clarke error grid. Further, the system is implemented on a web server that interfaces with Dexcom's share-API. The web service makes prediction data available to care-providers in real time for timely intervention in hypoglycemia events.

Table of Contents

List of Figures

List of Tables

1. Introduction

Type 1 Diabetes (T1D) Mellitus is an autoimmune disorder that affects the body's ability to produce insulin and thereby, metabolize glucose. Approximately 1.5 million people live with this condition in the US alone. While it is commonly known as 'Juvenile Diabetes', more adults live with the condition than children (National Center for chronic Disease Prevention and Health Promotion, 2017). Managing T1D requires keeping constant watch on blood glucose concentration and artificial insulin intake. Thus, managing the condition can become an overwhelming task for the newly diagnosed, especially children. With the increased availability of Continuous Glucose Monitoring (CGM) and computerized insulin pumps in the last decade, focus has shifted towards accurately anticipating glucose response to carbohydrate and insulin intake (David Rodbard, 2016). Research in this domain has ranged from advanced machine learning techniques to a control-theory approach to understanding glucose metabolism. This project focuses on the latter.

Based on real patient data, several mathematical formulations are identified that can replicate real blood glucose (BG) behavior. These physiological models are expressed in the observer-canonical form and their parameters are fit to biological behavior. While, the blood glucose behavior simulated by a physiological model is repeaTable for a given set of inputs, real blood glucose behavior shows considerable variability across subjects and time. These deviations could be attributed to uncertainties in carbohydrate intake and unmodeled system behavior.

Simultaneous state and parameter estimation of physiological model parameters can be achieved through Kalman Filters. CGM measurements are used to recursively

update the model physics and the updated model is used to generate predictions 30 minutes into the future.

The algorithm is implemented as a web service which can be set up with ease. The design and implementation borrow many elements from the open source ‘NightScout’ system (NightScout, 2019). It uses the Dexcom share API to download CGM data through HTTP POST requests. The website also provides 12-hour reports on prediction accuracy and glycemic control.

The current prediction system also lays the necessary foundation work for a closed loop BG regulation system in concert with automated insulin pumps. Such “artificial pancreas” systems can lower the incidence of hypoglycemia and excessive time spent in hyperglycemia. Improved glycemic control is a pre-requisite for minimizing the complications from life-long diabetes (Group, 1993). Such a system would also help calm anxious nerves of newly diagnosed children and their parents.

The next chapter focuses on compartmental models of glucose metabolism in type 1 diabetics and the ever-evolving complexity of these models from a signal processing standpoint. Chapter 3 introduces existing approaches to hypoglycemia prediction and discusses the metrics used to evaluate prediction performance. Chapter 4 discusses the development of CGM devices and their accuracy metrics. The filtering and prediction algorithms are developed in chapter 5. A discussion of prediction results and their limitations is provided in chapter 6. Finally, chapter 7 provides an overview of the web service ‘SweetSpot’ developed in this project.

2. Compartmental Models

2.1. Introduction

Academic interest in computer assisted insulin therapy dates to the mid-1980s (Deutsh & Lehmann, 1996). Given the wide variations in physiology across subjects as well as time-varying behavior patterns, optimal diabetes treatment is possible only with regular monitoring and patient-specific models. Work in this domain is split across physiological modelling (compartment models) and algorithm-based approaches. Furthermore, algorithmic approaches are supplemented through large scale data collection. 'Knowledge Discovery through Data' has found a lot of appeal in the diabetes community (Lebech, Johansen, & Hejlesen, 2016).

Compartmental models of the glucoregulatory system are built from sub models of various physiological processes within the body. These include but are not limited to endogenous production of glucose in the liver, the extraction of glucose by the kidneys and various insulin dependent utilization mechanisms. One of the earliest full model implementations can be seen in AIDA (Lehmann & Deutsch, 1992). Before AIDA, pharmacodynamic models were developed to simulate the action of subcutaneous insulin on glucose dynamics (Berger & Rodbard, 1989). Figure 1 provides a screenshot of a typical AIDA simulation with 5 meals and different insulin intakes. The compartment models used in the present work are described in the following sub-sections.

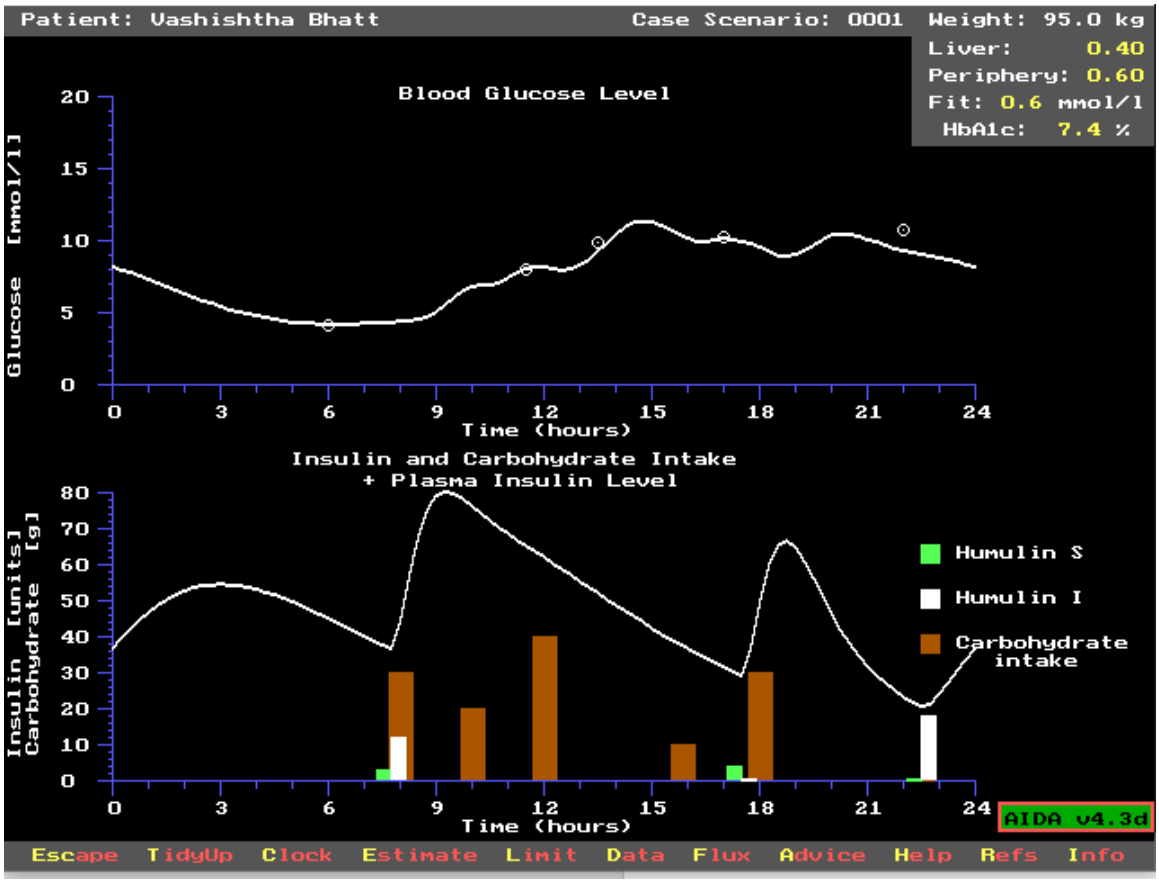

Figure 1: AIDA simulation for a single day

2.2. Gut Sub compartment

The gut subsystem mimics carbohydrate ingestion and glucose uptake. Several simulation models have been proposed. Lehman et al. proposed a single compartment model of glucose absorption that followed a trapezoidal gastric emptying profile (Lehmann & Deutsch, 1992). The amount of glucose in the gut, G_{gut} following the ingestion of a meal containing Ch millimoles of carbohydrate is defined as:

$$\frac{dG_{gut}}{dt} = G_{empt} - k_{abs}.G_{gut} \quad (1)$$

k_{abs} is the rate constant of glucose absorption into the blood stream. The rate of gastric emptying G_{empt} has a trapezoidal profile as shown in Figure 2 and describes the movement of ingested carbohydrate through the compartment.

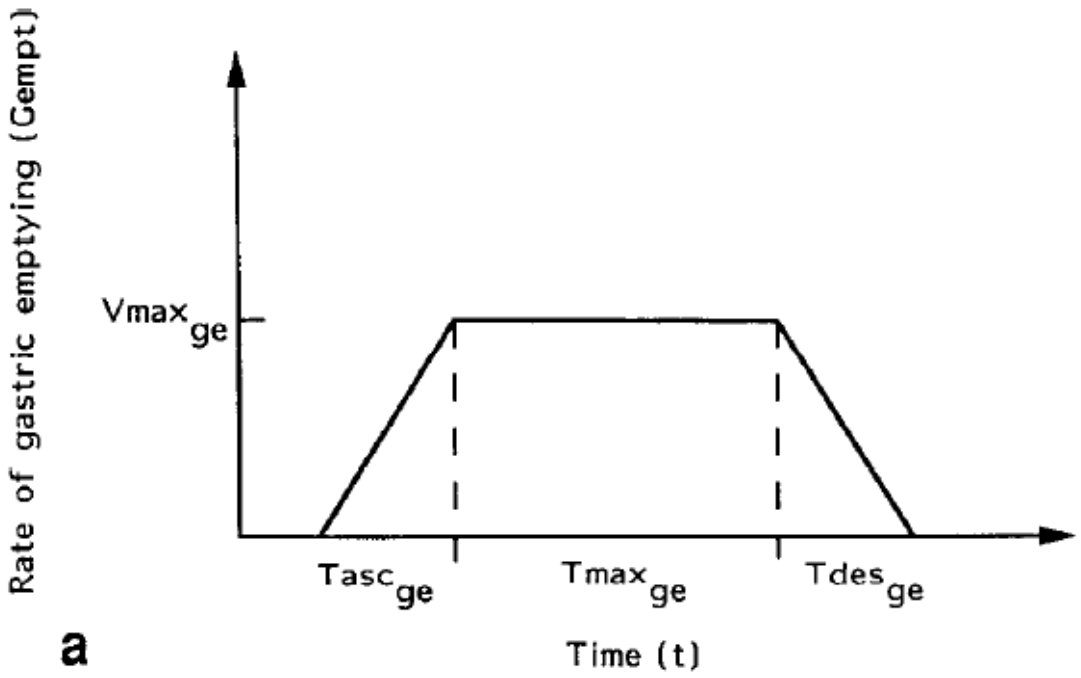

Figure 2: Rate of gastric emptying. *(Lehmann & Deutsch, 1992)*

T_{asc} and T_{des} are the respective lengths of the ascending and descending branches of the gastric emptying curve and have default values of 30 minutes. V_{max} is the maximal rate of gastric emptying and is set to 120 mmol/h. Furthermore, for small quantities, the rate of gastric emptying will not plateau out and was defined as:

$$T_{asc} = T_{des} = \frac{2\ Ch}{V_{max}} \tag{2}$$

Man, et al. compared several models of glucose absorption against oral glucose tolerance test (OGTT) data (Man, Camilleri, & Cobelli, 2006). The database consisted of 41 subjects with varying level of glucose tolerance. During the oral glucose tolerance tests,

75 g of glucose was administered, and researchers collected blood samples at 5-minute intervals up to 6 hours after ingestion. The Rate of Appearance (Ra) of glucose was then calculated.

The present study uses a 2-compartment model proposed by Man et al. with a constant rate of emptying across the gut compartments. The model treats carbohydrate ingestion as step inputs. A descriptive image of the model is provided in Figure 3.

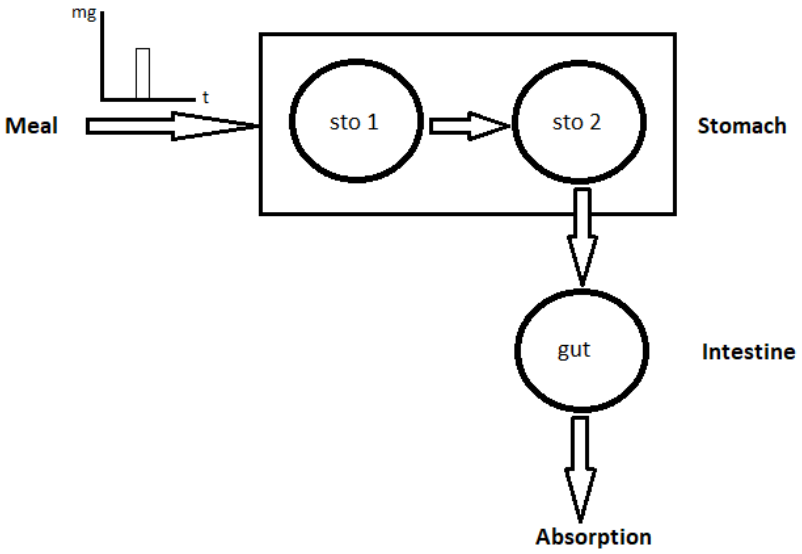

Figure 3: Teo-compartment model of glucose absorption from gut.

The model describes carbohydrate transit through the stomach and small intestine. It consists of a linear chain of three compartments. The discretized model Equations are:

$$\begin{bmatrix} \boldsymbol{q_{sto1}} \\ \boldsymbol{q_{sto2}} \\ \boldsymbol{q_{sto}} \\ \Delta_{\boldsymbol{gut}} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{q_{sto1}}(1-k_{emp})+\boldsymbol{U_c} \\ \boldsymbol{q_{sto2}}(1-k_{emp}) + \boldsymbol{q_{sto1}}k_{emp} \\ \boldsymbol{q_{sto}}(1-k_{abs}) + \boldsymbol{q_{sto2}}k_{emp} \\ \boldsymbol{q_{sto}}k_{abs}f \end{bmatrix}_{k} \quad (3)$$

State variables are **boldfaced** and k is the time step. This is a linear time invariant system where the carbohydrate input (in mg) is denoted as $\boldsymbol{U_c}$. q_{sto1}, q_{sto2} and q_{sto} is the carbohydrate content in each of the compartments (in mg). Δ_{gut} denotes the rate of appearance of glucose (in mg) in the blood stream. The system response to a 30g carbohydrate input for different values of k_{abs} and $k_{emp} = 0.9$ is shown in Figure 4.

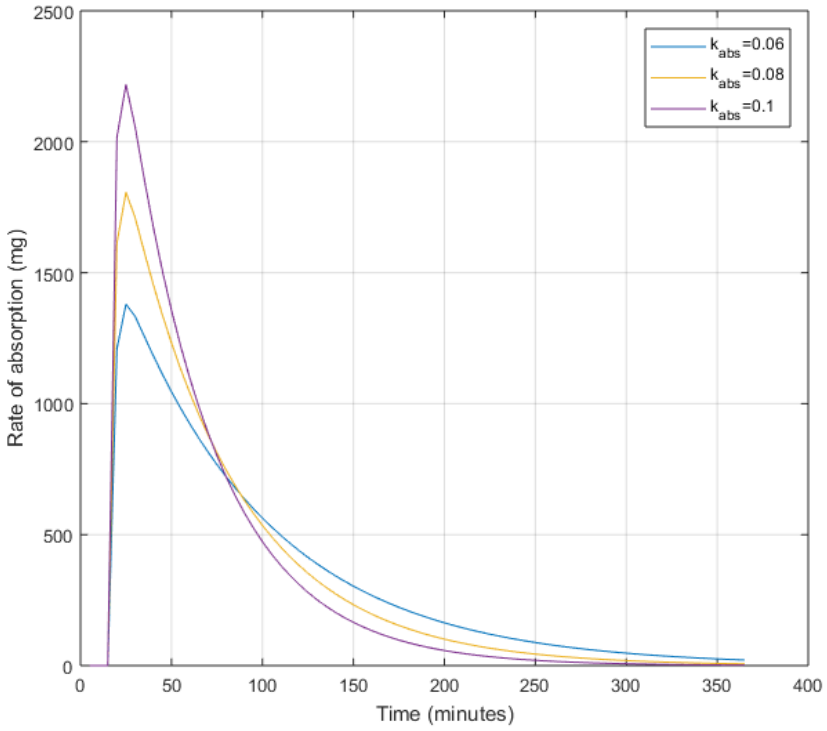

Figure 4: System response to 30g carbohydrate intake at time t=0.

2.3. Insulin sub compartment

The insulin system incorporates the uptake of artificial insulin through subcutaneous injection. For type 1 diabetics, the secretion of insulin from the pancreas is not modeled. Insulin degradation occurs both in the liver and the periphery, however,

insulin clearance is modeled as a single rate constant. A single compartment model of insulin is used for this study (Duke, 2010).

$$\begin{bmatrix} \boldsymbol{I_{sub}} \\ \boldsymbol{I_c} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{I_{sub}}(1-\alpha_{ir}) + \boldsymbol{U_I} \\ \boldsymbol{I_c}(1-\alpha_{ic}) + \dfrac{f_I S \boldsymbol{I_{sub}} \alpha_{ir}}{V_I bm} \end{bmatrix}_{k+1} \tag{4}$$

The parameters α_{ir} and α_{ic} are rate constants of insulin absorption from subcutaneous tissue and clearance respectively. For simplicity, the model does not consider variations in insulin type. U_I denotes the insulin input in micro Insulin units µu. The 'Insulin Unit (u)' is the most basic measure of insulin. Injected insulin preparations are denoted in concentrations of u/ml denoting the amount of insulin present per milli liter of liquid. I_{sub} is the total amount of insulin present in subcutaneous spaces. I_c denotes the insulin concentration in blood with units µu/ml. V_I is the total volume of the insulin compartment in ml/kg. bm denotes the body mass of the subject in kg. An extra parameter S is added to represent insulin sensitivity of the subject on a scale of 0 to 1. Figure 5 shows the insulin concentration vs. time for a bolus of 10u taken at time t=0.

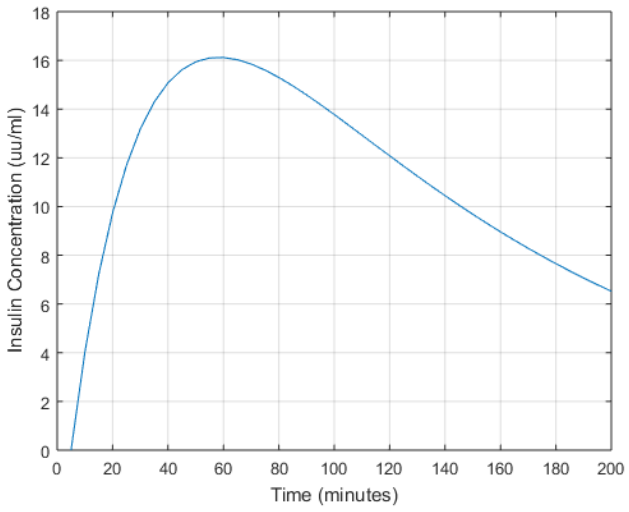

Figure 5: System response to a 10u bolus at time t=0.

2.4. Glucose sub compartment

The glucose sub compartment models the dynamic effects of insulin, carbohydrate and liver on blood sugar levels. Like for the meal subsystem several attempts have been made over the last two decades to make accurate physiological models for this. These models span a wide range of complexity. This study uses the model defined by Plis et al. (Plis, Shubrook, & Schwartz, 2014). The model strikes a good balance between fidelity and mathematical simplicity.

$$\begin{bmatrix} \Delta_{\boldsymbol{egp}} \\ \Delta_{\boldsymbol{dep}} \\ \Delta_{\boldsymbol{ind}} \\ \Delta_{\boldsymbol{clr}} \\ \boldsymbol{g_m} \end{bmatrix}_{k+1} = \begin{bmatrix} \frac{-\boldsymbol{g_m}\alpha_{egp1}}{Q} + \alpha_{egp2} e^{\frac{-\boldsymbol{I_c}}{\alpha_{egp3}}} \\ -\alpha_{dep1}\boldsymbol{I_c}\left(\frac{\boldsymbol{g_m}}{Q} + \alpha_{dep2}\right) \\ -\alpha_{ind}\sqrt{\frac{\boldsymbol{g_m}}{Q}} \\ -\alpha_{clr1}\left[\frac{\boldsymbol{g_m}}{Q}\right]^{\alpha_{clr2}} \\ \boldsymbol{g_m} + \Delta_{\boldsymbol{abs}} + \Delta_{\boldsymbol{egp}} + \Delta_{\boldsymbol{dep}} + \Delta_{\boldsymbol{ind}} + \Delta_{\boldsymbol{clr}} \end{bmatrix}_k \tag{5}$$

The endogenous production of glucose by the liver is represented by Δ_{egp} in mg. It is modeled a negative linear function of glucose mass in blood g_m governed by rate constant α_{egp1}. It also varies exponentially with Insulin concentration. Δ_{dep} denotes the extraction of glucose (in mg) from blood due to insulin action in mg at a rate of $-\alpha_{dep1}$ due to insulin action. Δ_{ind} (mg) accounts for the glucose utilization by the central nervous system at a rate of $-\alpha_{ind}$. Several authors have modeled renal clearance of glucose as a piecewise function (Man, et al., 2014).

$$\Delta_{clr} = \begin{cases} -\propto_{clr} (g_c - 115), g_c > 115 \\ 0, \ otherwise \end{cases} \tag{6}$$

The piecewise function is not differentiable, which can pose a problem for training models. As a remedy, in the present work, a first order polynomial approximation is instead used. While a higher order polynomial reduces the error with respect to the piecewise version, it increases the potential for linearization error during EKF implementation. A first-order polynomial is therefore a good tradeoff.

$$\Delta_{clr} = -\propto_{clr1} {g_c}^{\propto_{clr2}} \tag{7}$$

The total mass of glucose in the body g_m (mg) is the sum of all the compartment contributions. Glucose mass is related to concentration g_c (mg/dl) through volumetric parameter V_g (dl/kg) which denotes the volume of glucose in blood (in dl) per kg of body weight.

$$Q = V_g bm \tag{8}$$

Figure 7 provides an overview of the complete compartmental model used in the present work. The default values of the model parameters are listed in Table 1 along with their units of measure. The values have been discretized to 5 minutes. To check model behavior, a simulation is run for randomly selected days in the dataset one at a time. The model parameters in literature for insulin clearance and carbohydrate absorption were further adjusted by trial and error to ensure minimum variance between model-predicted and actual signals. A comparison between model predicted and actual BG is provided in Figure 7.

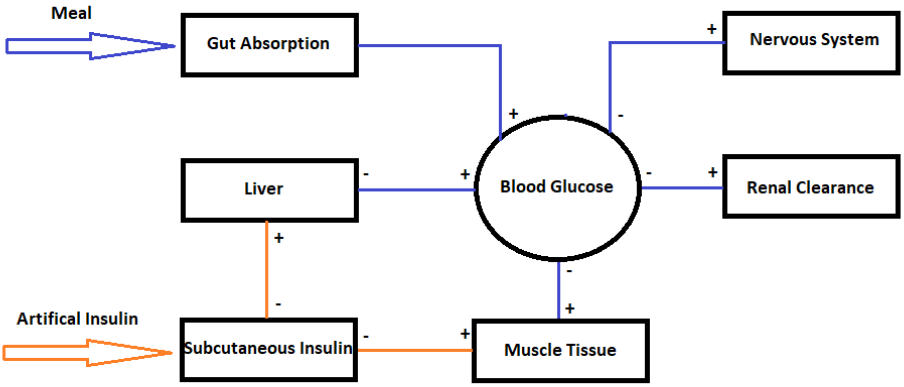

Figure 6: Complete system model.

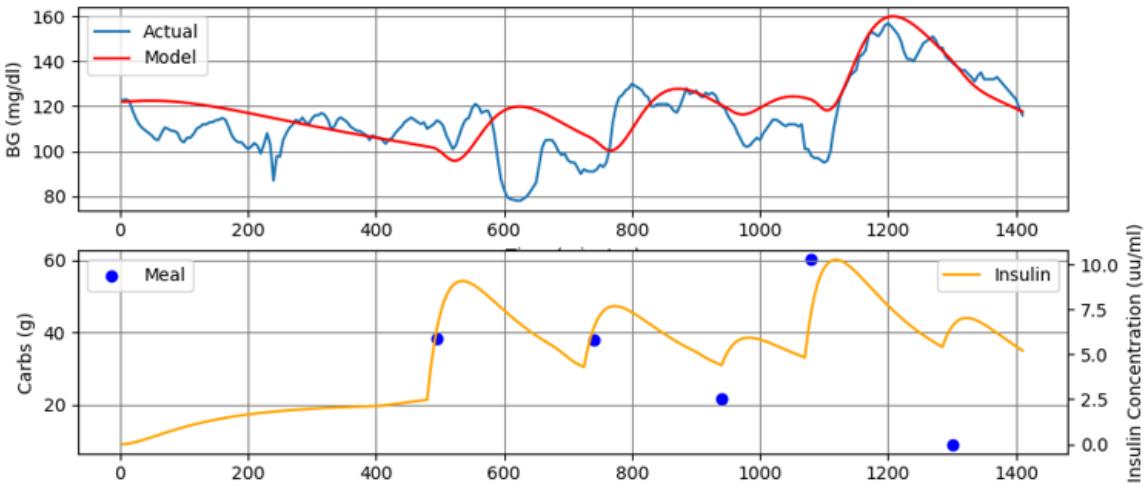

Figure 7: Model simulation vs. actual glucose concentration. ***Upper panel****: Glucose concentration* g_c. ***Lower panel:*** *Insulin concentration* I_c *and meal inputs* U_c

Table 1: Model parameter values and units.

Parameter	Value	Explanation
$\propto_{ir}$	0.04	Rate of insulin absorption.
$\propto_{ic}$	0.165	Rate of insulin clearance.
V_I	1.42 ml/kg	Insulin volume.
k_{emp}	0.9	Rate of gut emptying.
k_{abs}	0.06	Rate of carb absorption.
S	1	Insulin sensitivity.
$\propto_{dep1}$	0.128	Rate of insulin dependent utilization
f	0.2	Amplification factor
$\propto_{dep2}$	90	Fitting parameter
$\propto_{ind}$	6	Rate of insulin independent utilization
$\propto_{egp1}$	0.165	Rate of endogenous glucose production.
$\propto_{egp2}$	225	Fitting parameter
$\propto_{egp3}$	15	Fitting parameter
$\propto_{clr1}$	0.04	Rate of renal clearance
$\propto_{clr2}$	1.676	Exponent of renal clearance
f_I	0.00005	Amplification factor
V_G	2.2 dl/kg	Glucose volume

The complete system model in observable canonical form is given by Equations 9 and 10. g_c denotes the blood glucose concentration in mg/dl.

$$\begin{bmatrix} \boldsymbol{I_{sub}} \\ \boldsymbol{I_c} \\ \boldsymbol{q_{sto1}} \\ \boldsymbol{q_{sto2}} \\ \boldsymbol{q_{sto}} \\ \Delta_{\boldsymbol{gut}} \\ \Delta_{\boldsymbol{egp}} \\ \Delta_{\boldsymbol{dep}} \\ \Delta_{\boldsymbol{ind}} \\ \Delta_{\boldsymbol{clr}} \\ \boldsymbol{g_m} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{I_{sub}}(1-\alpha_{ir}) + \boldsymbol{U_I} \\ \boldsymbol{I_c}(1-\alpha_{ic}) + \dfrac{S\boldsymbol{f_I}\boldsymbol{I_{sub}}\alpha_{ir}}{V_I bm} \\ \boldsymbol{q_{sto1}}_{(1-k_{emp})+\boldsymbol{U_c}} \\ \boldsymbol{q_{sto2}}(1-k_{emp}) + \boldsymbol{q_{sto1}}k_{emp} \\ \boldsymbol{q_{sto}}(1-k_{abs}) + \boldsymbol{q_{sto2}}k_{emp} \\ \boldsymbol{q_{sto}}k_{abs}f \\ \dfrac{-\boldsymbol{g_m}\alpha_{egp1}}{Q} + \alpha_{egp2}e^{\frac{-\boldsymbol{I_c}}{\alpha_{egp3}}} \\ -\alpha_{dep1}\boldsymbol{I_c}\left(\dfrac{\boldsymbol{g_m}}{Q} + \alpha_{dep2}\right) \\ -\alpha_{ind}\sqrt{\dfrac{\boldsymbol{g_m}}{Q}} \\ -\alpha_{clr1}\left[\dfrac{\boldsymbol{g_m}}{Q}\right]^{\alpha_{clr2}} \\ \boldsymbol{g_m} + \Delta_{\boldsymbol{abs}} + \Delta_{\boldsymbol{egp}} + \Delta_{\boldsymbol{dep}} + \Delta_{\boldsymbol{ind}} + \Delta_{\boldsymbol{clr}} \end{bmatrix}_k \tag{9}$$

$$[\boldsymbol{g_c}]_k = \left[\frac{\boldsymbol{g_m}}{Q}\right]_k \tag{10}$$

3. Hypoglycemia Prediction

3.1. Introduction

Timely detection of hypoglycemia helps T1D patients prevent and minimize the adverse impacts of low blood glucose. Clinicians define hypoglycemia as blood glucose concentrations at or below 70 mg/dl. Signs and symptoms of hypoglycemia include fatigue, anxiety, sweating and hunger. Patients may also feel irriTable and notice paling of skin (Mayo Clinic, 2018). Hypoglycemia in T1D patients usually occurs as a result of overcompensation (by artificial insulin) to dangerously high insulin levels (hyperglycemia). Even in healthy people, hypoglycemia can occurs after meals because the body produces more insulin than needed. This type of hypoglycemia is called reactive/postprandial hypoglycemia. Remining in a hypoglycemic state can result in unconsciousness as the brain needs glucose to function. Moreover, degraded motor control in a hypoglycemic state can be catastrophic when driving or operating heavy machinery. Long term T1D and its associated higher occurrence of hypoglycemic events can lead to hypoglycemia-unawareness. The body and brain no longer produce signs and symptoms associated with hypoglycemia, making detection and intervention difficult (diaTribe Learn, 2018). Timely hypoglycemia detection is therefore necessary for any computer-aided T1D support system. Moreover, the system must exhibit low false-positives rates as this would lead to a trust deficit between the user and the algorithm. This chapter will focus on previous academic work in this domain. A review of such work provides a baseline for evaluating the system developed in this work.

3.2. Scoring Prediction Accuracy

Blood glucose predictions can be scored on a Clarke error grid which was developed in 1987 to quantify clinical accuracy of self-estimated blood glucose versus values obtained from a meter. The x-axis is the reference concentration (mg/dl) (meter-value) and y-axis is the estimated concentration (mg/dl) (Clarke, Frederick, Carter, & Pohl, 1987). The scatter plot evaluation is divided across five zones.

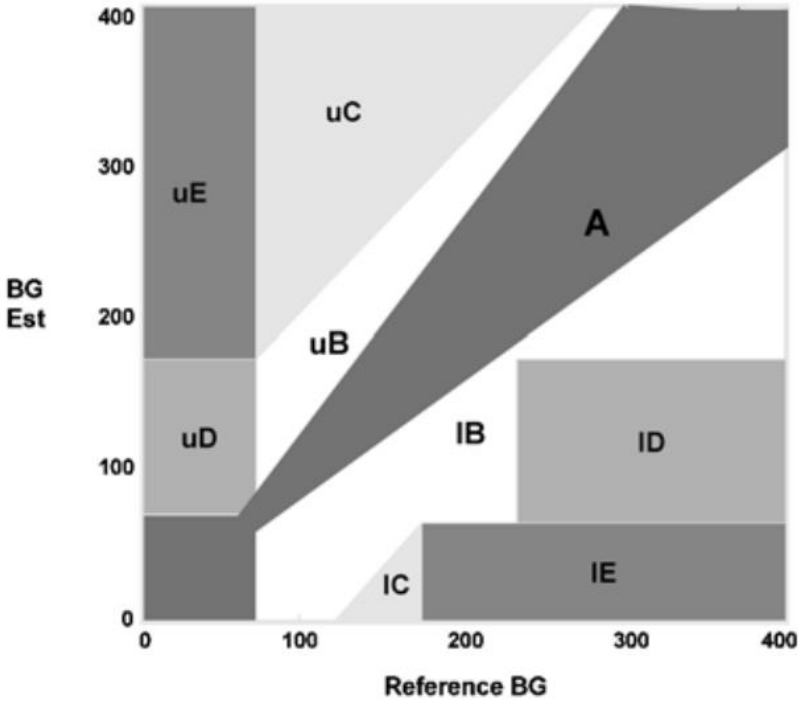

Figure 8: The original Clarke Error Grid Analysis (Clarke, Frederick, Carter, & Pohl, 1987).

For a prediction system:

1. Zone A represents predictions that are within 20% of the true (sensor) value.
2. Zone B represents predictions that differ from the true value by more than 20% but are inconsequential for treatment decisions.

3. Zone C represents predictions that lead to unnecessary treatment.
4. Zone D represents predictions that are dangerous as they indicate a failure to detect hypo/hyper glycemia.
5. Zone E represents predictions that would lead a patient to confuse treatment for hypoglycemia and hyperglycemia.

To aid in evaluation of continuous glucose monitors, a rate-error grid was developed as shown in in Figure 9. Together with the original Clarke error grid, this evaluation system is known as the CG-EGA (Continuous Glucose – Error Grid Analysis) (Wentholt, Hoekstra, & DeVries, 2006).

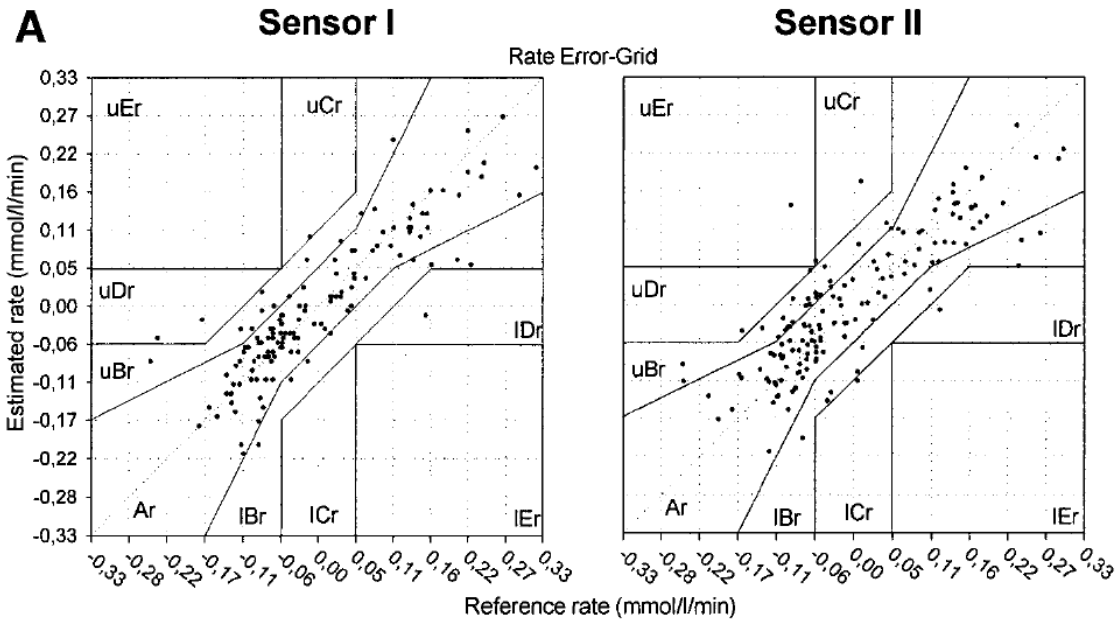

Figure 9: Rate-error grid analysis (Wentholt, Hoekstra, & DeVries, 2006).

Historically, work on blood glucose prediction has employed statistical metrics for performance evaluation. This makes benchmarking and comparison difficult. The most common metric is the root of the mean of squared errors (RMSE).

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(bg_{predicted,k} - bg_{actual,k}\right)^2} \quad (11)$$

$bg_{predicted,k}$ is the predicted value of blood glucose at time step *k* and $bg_{actual,k}$ is the actual value. For a given day, RMSE can be evaluated on the predicted blood glucose (at a future timestep) and actual blood glucose (at the same timestep).

Another popular error metric concerns the relative absolute difference between the predicted and actual blood glucose values.

$$RAD_k = \left|\frac{\left(bg_{predicted,k} - bg_{actual,k}\right)}{bg_{actual,k}}\right| \quad (12)$$

For the entire dataset, the RAD is averaged and expressed as a percentage.

$$MAD = \frac{1}{N}\left(\sum_{k=1}^{N} RAD_k\right) \times 100 \quad (13)$$

3.3. Review of Previous Work

3.3.1. Autoregressive Estimation

One of the simplest predictors that can be designed around CGM data is an Autoregressive (AR) model. There are numerous projects that span from simple AR estimators to more complex ARMAX (Autoregressive – Moving Average with Exogenous Input) systems.

Let k denote the timestamp of a CGM sample. A 2nd order AR has the following form.

$$g_k = a_0 + a_1 g_{k-1} + a_2 g_{k-2} + w_k \quad (13)$$

The order of an AR model is also known as lag. A higher lag AR model can be applied to identify processes that evolve over a longer time period. The weights a_1, a_2 and a_0 are determined by least-squares curve fitting. w_k is the process white noise. Even a 1st order model can predict crossing the hypoglycemia threshold 20-25 minutes ahead of time (Sparacino, et al., 2007). Sparacino et al. added an exponential forgetting-factor to past measurements that allows the weights to rapidly trach changes in the CGM signal. The prediction algorithm is described in Figure 10.

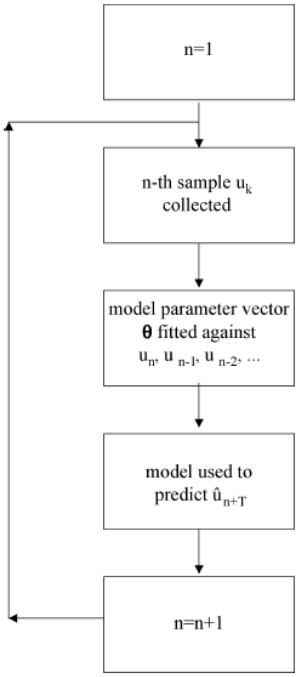

Figure 10: AR parameter estimation and glucose prediction (Sparacino, et al., 2007).

A comparison between actual blood glucose and predictions made 30 minutes ahead of time is shown in Figure 11. It is noted that the predicted signal tracks the actual CGM signal closely with exceptions at the peaks and nadirs. The average delay time for predictions during negative trends was found to be 12 minutes in the worst-case scenario.

The median RMSE of predictions error was 17.83 mg/dl for the entire dataset of 28 subjects.

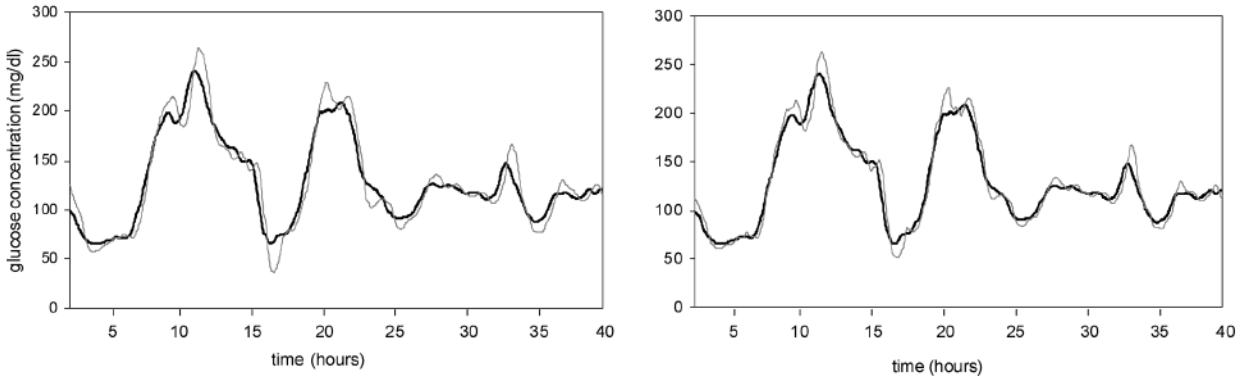

Figure 11: A comparison of polynomial model (left) and 1st order AR model (right). Actual BG in dark. 30 minutes predicted BG in gray (Sparacino, et al., 2007).

An autoregressive moving average (ARMA (p, q)) model is described in Equation 14. Here, p and q are the orders of the AR and MA parts of the model.

$$g_k = \sum_{i=1}^{p} a_i g_{k-i} + \sum_{i=1}^{q} w_{k-i}\theta_k + w_k \qquad (14)$$

w_k are the noise terms and θ_k are parameters of the moving average process. An ARMA model thus has p+q+1 parameters which must be estimated. These are estimated via Maximum Likelihood Estimation. Plis et al. used an ARMA model to predict blood glucose levels at 30-minute and 50-minute prediction horizons (Plis, Shubrook, & Schwartz, 2014). The results were compared to a physician predicted baseline in terms of RMSE. The ARMA model did not fare significantly better than the AR model with a mean RMSE over the dataset of 22.9 mg/dl. Oruklu et al. compared ARMA (3,1) predictions to 3rd order AR predictions and found the ARMA model significantly improved prediction accuracy for type-2 diabetes subjects (Oruklu, Cinar, Quinn, & Smith, 2009). For a 30-

minute prediction horizon, the ARMA model predicted over 81% of glucose values within zone A of the CG-EGA point error grid. Moreover, the system predicted 0% false positives (zone uD) for hypoglycemic events.

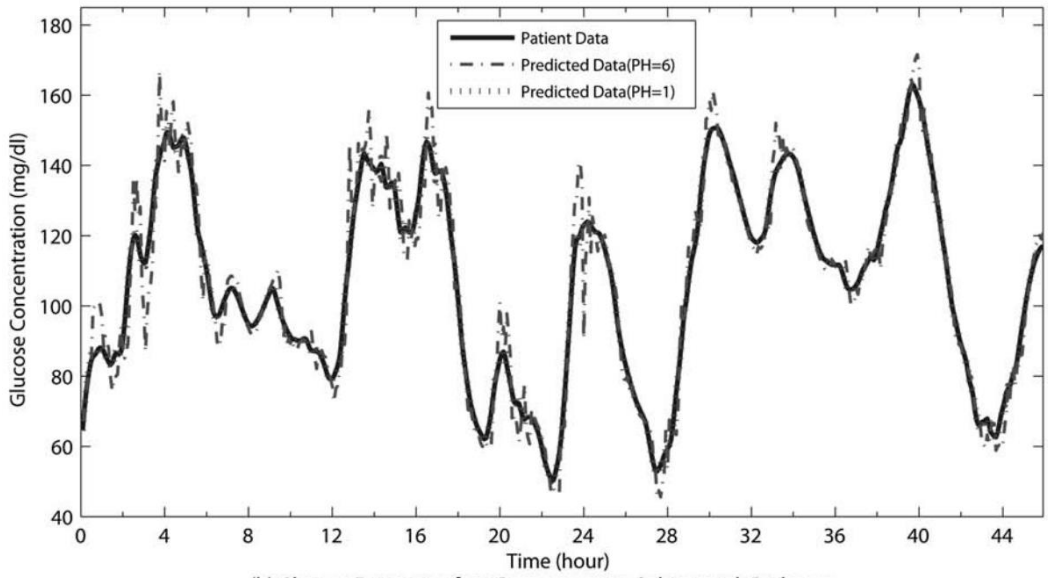

Figure 12: Actual glucose values versus predictions at 5 minutes (PH=1) and 30 minutes (PH=6) (Oruklu, Cinar, Quinn, & Smith, 2009).

Furthermore, Autoregressive models with Exogenous Inputs (ARX) consider the effect of carbohydrate and insulin intake.

3.3.2. Bayesian Estimation Through Kalman Filters

Bayesian estimation methods rely on the Bayes rule, which gives the probability of an event given prior knowledge of the conditions related to the event. Kalman filters provide sequential minimum mean squared estimation (MMSE) of a signal which is embedded in noise. The signal is characterized by a dynamical model. If the signal and noise are jointly gaussian, then the Kalman Filter is an optimal MMSE estimator (Kay, 1993). Kalman filtering is a two-step process. First, the state is evolved through a transition

function along with the uncertainty in its estimate. Second, the measured signal is compared with the prediction from the first step. The error is used to calculate the 'Kalman-Gain' which is then used to update the state estimate based on the relative uncertainty in process evolution and sensor measurement. The Extended Kalman Filer (EKF), Adaptive-Extended Kalman Filter (AEKF) and Unscented Kalman Filters (UKF) used in this study are discussed in detail in Chapter 5.

With the evolution of complex physiological modelling coupled with CGM data, Kalman filtering became an attractive option for prediction and de-noising algorithms. Knobbe et al. used an Extended Kalman Filter algorithm for CGM signals that is consistent across sensors based on optical and electrochemical processes (Knobbe & Buckingham, 2005). In their paper on CGM data smoothing, Fachchinetti et al. note that the signal to noise ratio in CGM signals is not constant, therefore, filter parameters must be continuously adjusted for noise (Facchinetti, Sparacino, & Cobelli, 2009). They compared the Kalman filter approach to the original moving average approach on simulated and real-patient data and found significant improvements.

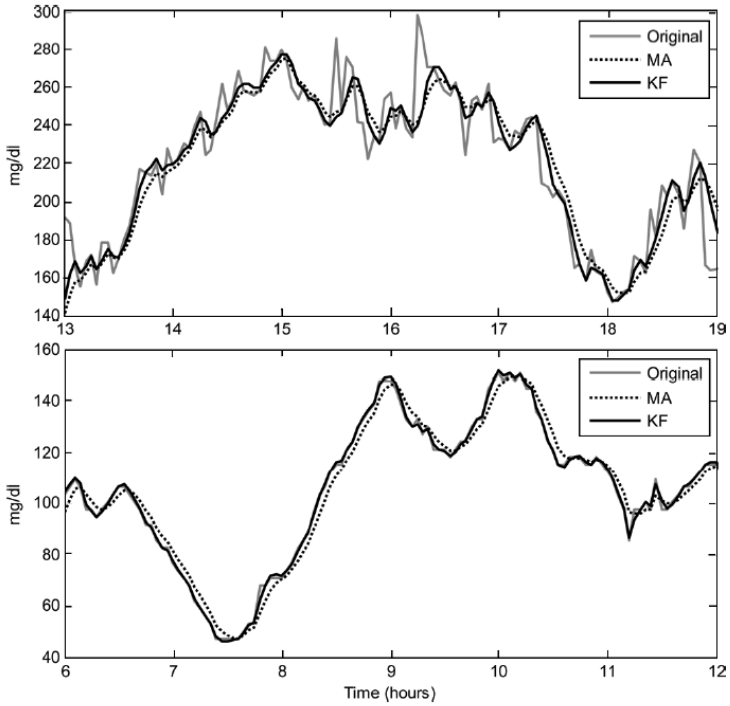

Figure 13: A comparison of Kalman filter (KA) and moving average (MA) denoising (Facchinetti, Sparacino, & Cobelli, 2009).

Kalman filters can be used for predictions by evolving the updated state vector through the state-transition function repeatedly until the prediction horizon is reached. Palerm et al.. applied a linear-time-invariant glucose dynamics model to predict glucose concentrations 30-minutes ahead and generate warning-alarms for hypoglycemia. The study achieved a sensitivity and specificity of 90% and 79% respectively. Furthermore, their algorithm allowed for a user-defined warning threshold depending on tolerance of false-positives (Palerm & Bequette, 2007). To the author's best knowledge, this is the earliest use of this technique. In his PhD thesis, Duke applied an EKF to his physiological model (a variant of which is used in this study) and predicted blood glucose values up to 45 minutes into the future (Duke, 2010). The predicted results were evaluated on the CG-

EGA and achieved a prediction accuracy of 84% and 65% for 15-minute and 45-minute prediction horizons. The study also compared results from AR and ARX models. A variant of the physiological model which incorporated exercise was also developed.

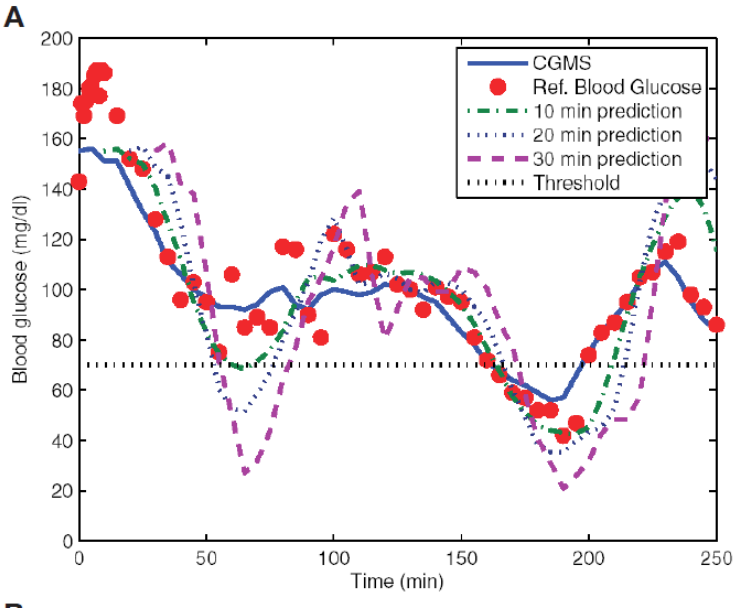

Figure 14: Effect of prediction horizons on predictor performance (Palerm & Bequette, 2007).

In both Duke's and Palerm's study, the parameters of the model remained constant. However, there is both an intra-patient and inter-patient variation in response to glucose and insulin inputs. Therefore, filtering approaches must combine state and parameter estimation. This is a difficult problem due to the large number of parameters and state variables even in conservative physiological models. Wang et al.. tackled this problem by modelling glucose absorption from the gut as finite-impulse-response function. Glucose and insulin metabolism are defined as an AR process. A second order Extended Kalman Filter is used to recursively estimate the AR weights. The weights are assumed to evolve as function of noise alone. Furthermore, the consistency of the Kalman filter is evaluated through the Normalized Innovation Squared metric. The algorithm was applied

on both simulated and real patient data. The prediction system achieved a MAD of 10% and 20% respectively for simulated and real data (Wang, et al., 2014).

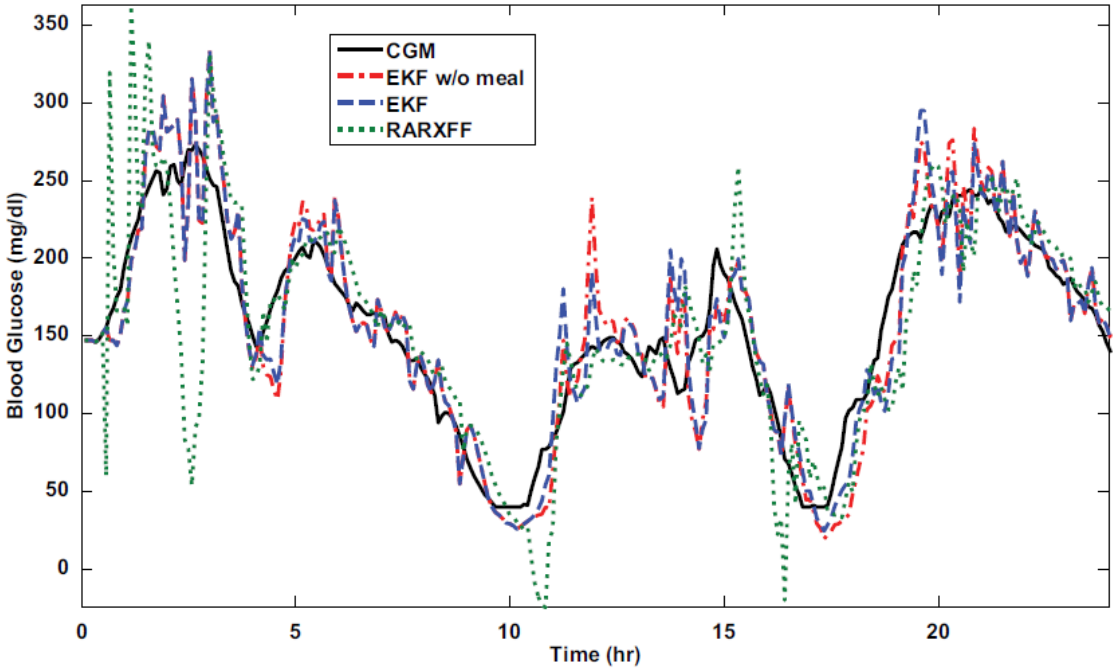

Figure 15: CGM sensor readings versus 30-minute predictions (Wang, et al., 2014).

3.3.3. Machine Learning Methods

The application of machine learning tools in healthcare has picked up pace in the last decade. This is due to the availability of cheap computing power and storage. Hypoglycemia detection problems can be approached as classification problems (Hypo Yes/No) or regression problems as in previously discussed approaches. Research has mostly focused on the latter using recurrent neural networks (RNN).

Dubosson et al.. trained a classifier for detecting incidences of reactive hypoglycemia on real and simulated datasets. In their paper, they note that the problem is difficult due to imbalanced datasets and incorrectly labeled/missing data. A Random

forest classifier achieved F1 scores of 2.41%. The best performing classifier was a Support Vector Machine with an F1 score of 13.7%. Due to imbalanced datasets, the false positive rates were too high for the prediction system to be practical (Dubosson, Beatriz, & Schumacher, 2017).

Plis et al.. used a Support Vector Regressor (SVR) to identify the parameters of Duke's physiological model. The gaussian kernel SVR was trained on features which encoded the difference between predicted and actual blood glucose from the model. The SVR achieved a 30-minute prediction accuracy of 19.5 mg/dl. For hypoglycemia detection, the SVR system achieved F1 score of 30%. However, when compared to physician and ARMA RMSE values of 19.8 mg/dl and 22.9 mg/dl, the SVR performed relatively well (Plis, Shubrook, & Schwartz, abcd, 2014).

Pappada et al. developed a feed-forward neural network trained on 17 patients and tested its predictions on a 50-180-minute prediction horizons (Pappada, Cameron, & Rosman, 2008). The predictions were scored with MAD. They also provide breakdown of MAD in hypoglycemic and hyperglycemic regions. The prediction system achieved an overall MAD of 18.7% at the 100-minute prediction horizon. Moreover, the system failed at detecting hypoglycemia at the 100-minute horizon. The system achieves an overall MAD of 6.7% for the 50-minute prediction horizon but still struggles to detect hypoglycemia. This is due to the highly imbalanced nature of CGM datasets as subjects actively avoid hypoglycemia events.

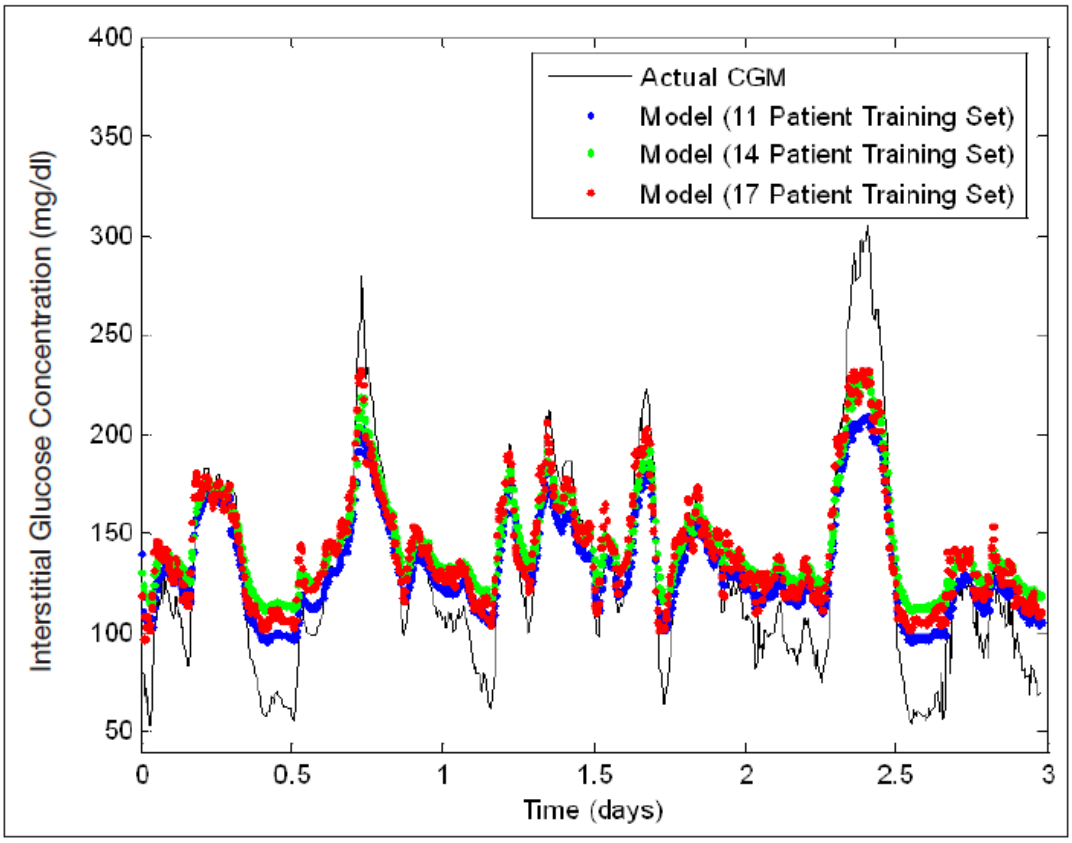

Figure 16: Predictions over a 100-minute horizon versus actual CGM data for different training sets (Pappada, Cameron, & Rosman, 2008).

Gandia et al. developed a feed-forward neural network with time-lagged data like that of Pappada et al. A network used a lag of 20, which translates to a time history of 100 minutes. The predictions were evaluated at horizons of 15,30 and 45 minutes. The trained network was also benchmarked against a 1st order AR model developed by Sparacino et al. For a prediction horizon of 30 minutes, the network achieved a mean RMSE of 17.45 mg/dl versus 20.27 mg/dl for the AR model on the same dataset. For a prediction horizon of 45 minutes, the system achieved an RMSE of 25.08 mg/dl versus 30.30 mg/dl achieved by AR. The system was also evaluated on CGM data from different device manufacturers.

The difference in prediction RMSE was within 1 SD of the mean (Perez-Gandia, et al., 2010).

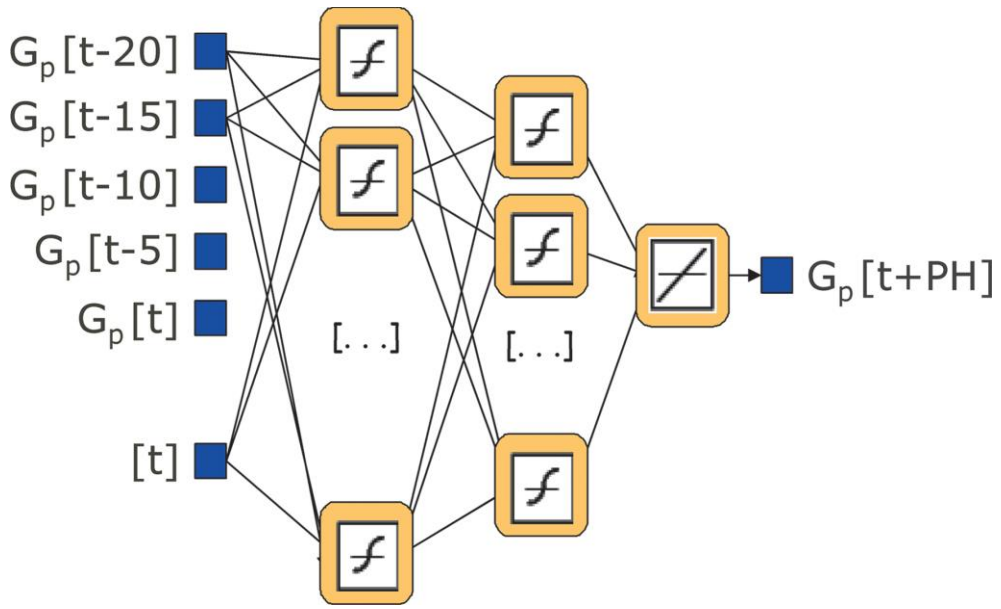

Figure 17: Neural Network Architecture (Perez-Gandia, et al., 2010).

Beyond just CGM samples, neural network models have been developed which consider compartment model predictions and meal inputs (Mougiakakou, Prountzou, & Nikita, 2005). These methods, however, are beyond the scope of this work

4. Continuous Glucose Monitoring

4.1. Introduction

The Diabetes Control and Complications Trial (DCCT) demonstrated that tight glycemic control can be achieved through frequent self-monitoring of blood glucose (Group, 1993). Glucose finger sticks were introduced in the 1980's to help patients accurately track blood glucose. Since their introduction, these porTable glucose meters have become smaller, more accurate and faster. While, extremely useful, finger sticks require pricking the skin to draw a blood sample. While not considered painful, this 'wet' procedure causes anxiety to those newly diagnosed (especially children). It also increases the risk of infections for those with weaker immune systems. This puts patients and caregivers at risk of contracting infections such as hepatitis B, HIV and hepatitis C (Geaghan, 2014). A universal compliance standard does not exist for finger stick meters but the International Organization for Standardization suggests a 15 mg/dl error margin for glucose levels under 75 mg/dl and readings within 20% (zone A of CEGA) for glucose levels above 75 mg/dl (Olansky & Kennedy, 2010).

Figure 18: Typical finger stick glucose meter (health24, 2017).

4.2. Continuous measurement.

Enzymatic measurement of glucose concentration based on hexokinase is the gold standard widely used in clinical laboratories. Current test-strip systems use glucose oxidase, glucose dehydrogenase and nicotinamide adenine (Olansky & Kennedy, 2010). Aleppo et al. conducted a randomized trial with T1D subjects using CGM and traditional finger stick (BGM) measurements. During the study, 276 participants were divided into groups of CGM-only and CGM+BGM. Efficacy of glycemic control was evaluated for both groups. It was noted that no severe hypoglycemic events occurred in the CGM-only group. Moreover, users in the CGM-only group showed the same level of glycemic control as the CGM+BGM group proving the safety and reliability of CGM meters. The study also provided a vast set of CGM data with meal information. This dataset is used in the present study (Aleppo, et al., 2017). The first CGM was approved by the FDA in 1999. Currently, there are at least 4 systems that see popular use.

1. Dexcom's G-series.
2. Medtronic's MiniMed.
3. Abbott Laboratories Freestyle Navigator.
4. Eversense Laboratories.

4.2.1. The Sensor

Currently available CGM's measure blood glucose in the interstitial fluid through minimally invasive means. The sensor rests just under the skin and does not puncture blood vessels. The initial generations of devices required a warmup time ranging from a few hours to a day (Klonoff, 2005). The sensor uses the enzyme glucose oxidase which reacts with glucose to form hydrogen peroxide. The hydrogen peroxide reacts with

platinum inside the sensor to generate an electrical signal (Diabetes Forecast, 2014). Chen et al. note that there are three generations of glucose biosensors (Chen, Zhao, Hong, Zhu, & Qian, 2017). The first generation relied on the production-detection of hydrogen peroxide. The second generation employ electron-acceptors to solve the oxygen deficiency. Research on the third generation of glucose sensors is focused on getting rid of artificial mediators and the glucose enzyme. The exhaustion of the glucose enzyme requires repeated sensor removals and insertion. This typically occurs at a frequency of 3 to 7 days. Recently, the development of carbon nanotubes and graphene-based electrodes have been intensively studied (Zhu, Gancedo, Flewitt, Xie, & Moussy, 2012). The chemistry behind the first-generation sensors is described in Equations 15 and 16. The reduction of flavin adenine dinucleotide (FAD) in the glucose oxidase (GOD) enzyme results in the reduced form of the enzyme $(FADH_2)$. The re-oxidation of flavin produces hydrogen peroxide as a byproduct.

$$GOD(FAD) + glucose \rightarrow GOD(FADH_2) + gluconolactone \quad (15)$$

$$GOD(FADH_2) + O_2 \rightarrow GOD(FAD) + H_2O_2 \quad (16)$$

The measurements of hydrogen peroxide help in the simple design and miniaturized fabrication of the sensor. The sensor needle also has a protective layer to prevent an immune response. A more detailed discussion on sensor chemistry is beyond the scope of this work.

4.2.2. Monitor and Signal Processing

Modern CGM systems are connected wirelessly to the sensor apparatus. This allows for easy removal and changes. Some systems have eliminated the need for a dedicated monitor and send data directly to the subject's smartphone over Bluetooth.

Some CGM systems are also bundled with insulin pumps to streamline the user interface. An example of the ‘MiniMed’ system by Medtronic is shown in Figure 19. Interfacing the CGM with a pump facilitates the development of closed loop insulin delivery systems, which are often called “artificial pancreas” systems.

Figure 19: The 'MiniMed' 630G by Medtronic.

CGMs measure glucose concentration in interstitial fluid rather than in blood. Thus, CGM measurements can lag the actual blood glucose concentration which presents a challenge to CGM accuracy. Moreover, the data is corrupted by random noise. Guerra et al. presented a deconvolution based approach relying on a linear regression model to account for the delay in glucose diffusion from blood to interstitial tissue (Guerra, Facchinetti, Sparacino, & Cobelli, 2012). The algorithm was tested on synthetic and real datasets and the results were evaluated on CG-EGA. A noTable aspect about this work

was that it treated the CGM device as a black box with no access to the raw electrode signals. The system achieved a score of 95% on zones A and B of the CG-EGA in the hypoglycemia region. It also outperformed the Kalman filter algorithm developed by Sparacino et al. discussed in chapter 3.

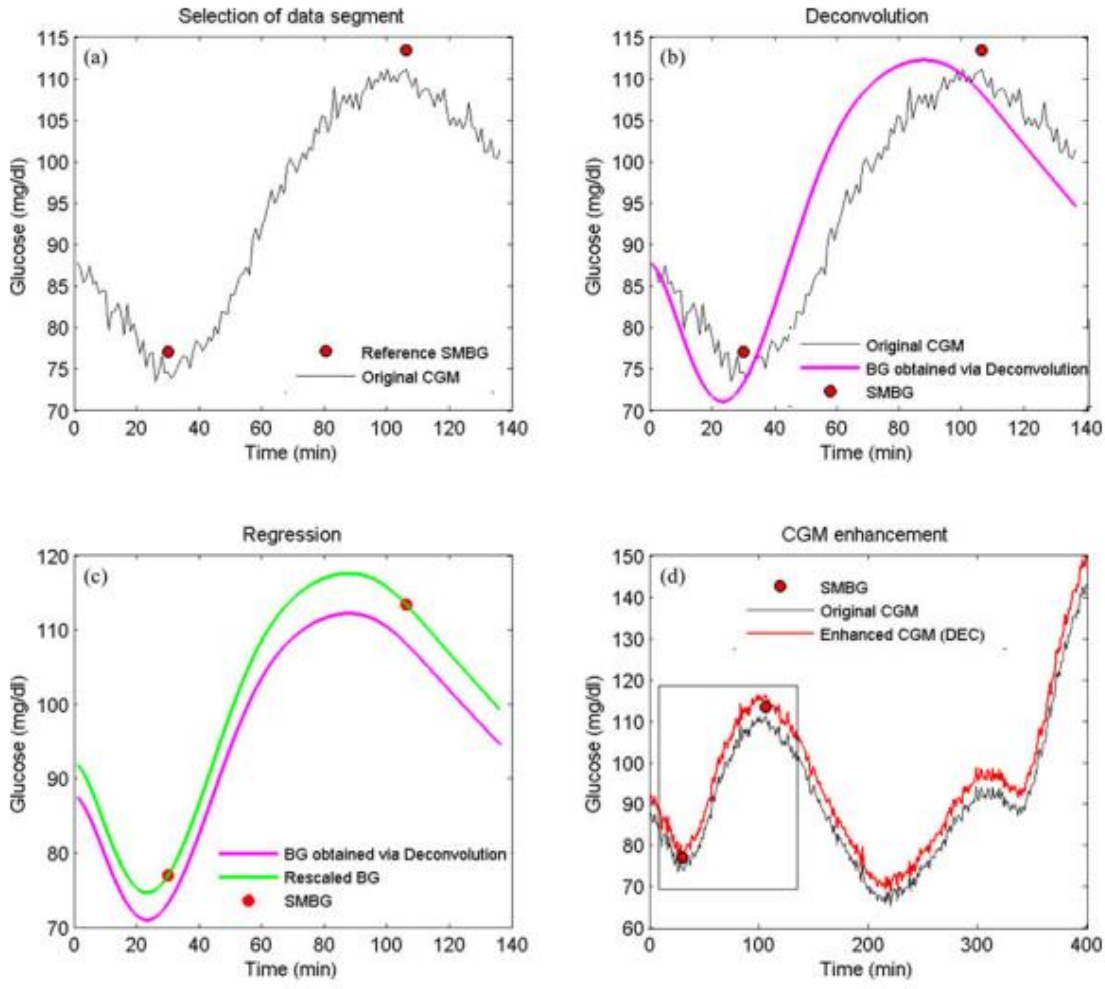

Figure 20: An online system for improving CGM accuracy through deconvolution and SMBG measurements (Guerra, Facchinetti, Sparacino, & Cobelli, 2012).

5. Estimation & Prediction from Kalman Filters

5.1. Introduction

This study focuses on the use of Kalman filters to generate the minimum mean squared error approximation of the model described by Equations 9 and 10. A review of previous work using this method is presented in Chapter 3. Multiple variations of Kalman filter based predictors are examined to identify the best performing algorithm. Figure 21 shows the generic linear Kalman filtering process. Let the state vector be *x* with an external noise covariance *Q*. Let *P* be the prediction error covariance and *R* the sensor noise covariance. The update and measurement Equations are present in matrices *A* and *H*. The Kalman Gain matrix *G* provides the necessary factor to reconcile between the measured signal *z* and expected signal *y*.

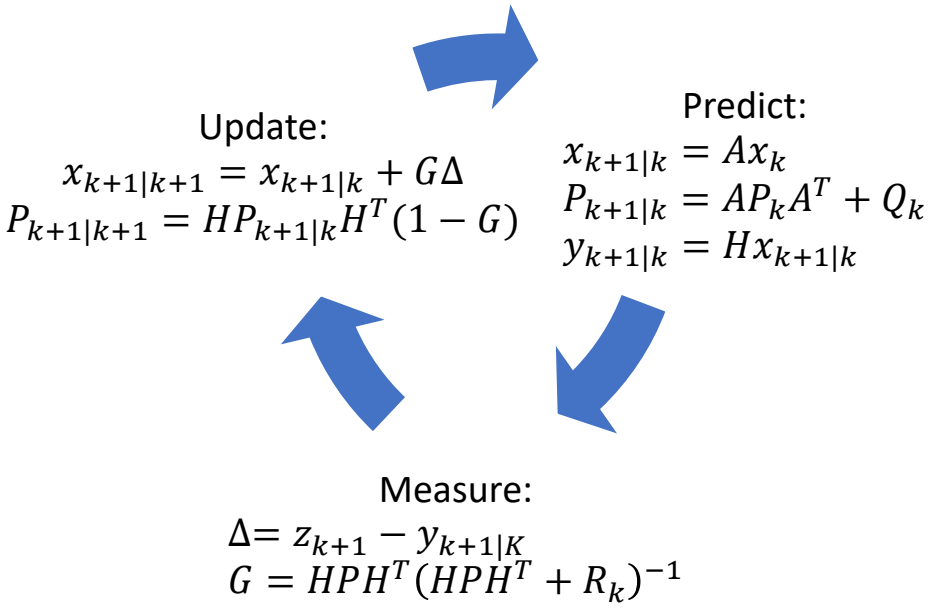

Figure 21: The general Kalman filtering process

The filtering process begins with predicting the state estimate and error at the next timestep. The pipe operator '|' in the Equations provides context on how the state/measurement estimate is calculated. For example, the subscript 'k+1|k' indicates that the estimate at the next timestep (k+1) was calculated from the previous estimate at time k. Once, measurement from the sensor becomes available at time k+1, the error Δ is calculated. The Kalman Gain is computed from the prediction error and sensor noise. The prediction is now 'updated' using the Kalman gain and Δ as correction factors. The update Equation is like a recursive least square estimator. The filtering algorithm assumes a gaussian distribution of uncertainty. Figure 22 shows how the Kalman filtering minimizes the uncertainty of mean between two gaussian distributions.

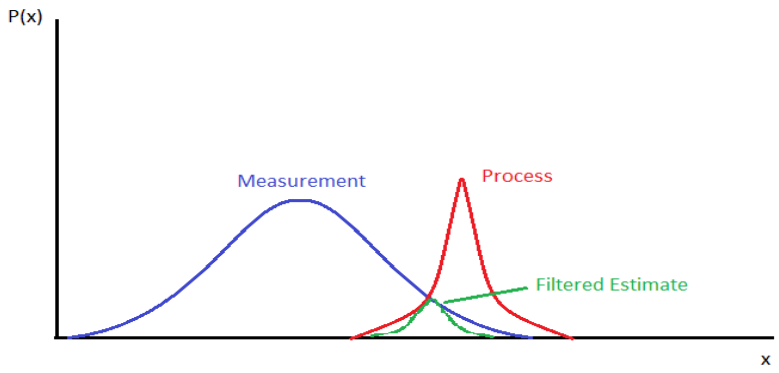

Figure 22: Probability distribution of state, measurement and filtered estimate.

5.2. An Adaptive Extended Kalman Filter for joint state & parameter estimation.

The system being studied is non-linear in nature, moreover, the dependencies between the state variables and changing parameters make it time-variant. This is problem is not tracTable with the typical Kalman filter discussed in section 5.1. An Extended Kalman filter innovates on the first order Taylor series approximation of the state transition function. This is called the Jacobian. The parameters of the model are assumed

to vary due to noise. They are then taken as additional states in the system. Let *x* (n x 1) denote the vector of states and θ (p x 1) denote the vector of parameters. The augmented state is *z* (n + p x 1) with w and η representing the uncorrelated Gaussian white noise in the process and parameters respectively. The covariance matrix of process and parameter noise is Q (n x n) and ϕ (p x p) respectively.

$$\widehat{z_{k+1|k}} = \begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} f(x_k, u_k, \theta_k) \\ \theta_k \end{bmatrix} + \begin{bmatrix} w_k \\ \eta_k \end{bmatrix} \tag{17}$$

u is the vector of inputs (insulin and carbohydrates). The measurement function *h* has associated sensor noise v and noise covariance R (1 x 1). The measurement y is scalar as only the glucose concentration is measured.

$$\widehat{y_{k+1|k}} = h\left(\widehat{z_{k+1|k}}\right) + v_k \tag{18}$$

$$F_k = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial \theta} \\ 0 & I \end{bmatrix}_{x_{k|k}, \theta_{k|k}} \quad and \; H_{k+1} = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial \theta} \end{bmatrix}_{x_{k+1|k}, \theta_{k+1|k}} \tag{19}$$

Step 1: The state is predicted from Equations 17 and 18. The covariance of the prediction is calculated from Equation 20.

$$P_{k+1|k} = F_k P_{k|k} F_k^T + \psi_k \tag{20}$$

$$\psi_k = \begin{bmatrix} Q_k & 0 \\ 0 & \phi_k \end{bmatrix} \tag{21}$$

Step 2: The measurement is used to calculate the innovation Δ. The Kalman gain G is also calculated.

$$\Delta_{k+1} = y_{k+1} - \widehat{y_{k+1|k}} \tag{22}$$

$$G_{k+1} = H_{k+1}P_{k+1|k}H_{k+1}^{T}\left(H_{k+1}P_{k+1|k}H_{k+1}^{T} + R_k\right)^{-1} \quad (23)$$

Step 3: Update the state estimate and process noise covariance with the Kalman gain and innovation respectively. The covariance of prediction is updated accordingly.

$$\widehat{z_{k+1|k+1}} = \widehat{z_{k+1|k}} + G_{k+1}\Delta_{k+1} \quad (24)$$

$$P_{k+1|k+1} = [I - G_{k+1}H_{k+1}]P_{k+1|k} \quad (25)$$

$$\psi_{k+1} = \alpha\psi_k + (1-\alpha)\left(G_{k+1}\Delta_{k+1}\Delta_{k+1}^{T}G_{k+1}\right) \quad (26)$$

The algorithm assumes that external process noise encapsulates the time-varying nature of the parameters. α is a forgetting factor that controls how rapidly ψ is updated. Rather than a constant noise covariance, such a system allows the filter to track glycemic disturbances more accurately. The updated glucose concentration is now calculated.

$$\widehat{y_{k+1|k+1}} = h\left(\widehat{z_{k+1|k+1}}\right) \quad (27)$$

Given the filtered state estimate $\widehat{z_{k+1|k+1}}$, a prediction can be made *PH* minutes into the future by computing Equations 17 and 18 $m = PH/D$ times for timesteps *k+1* to *k+1+m*. *D* is the time discretization which is set to 5 minutes in the present study. Therefore, for a *PH* of 30 minutes, *m=6*.

5.3. The Unscented Kalman filter for state estimation

The extended Kalman filter relies on linearization of the state transition function when calculating the prediction error covariance. Given the strong non-linearities in the physiological model, The Unscented Kalman filter (UKF) might be a better choice. The UKF relies on sigma-points to estimate the true shape of the probability distribution rather than a simple linearization. While this does increase the required computational overhead,

it yields a more accurate estimate of error in the predicted state. Multiple strategies exist for the calculation of sigma points. This study uses the popular method described by Merwe and Van (Merwe & Wan, 2003). Consider a system described by Equations 28 and 29.

$$x_{k+1} = f(x_k, u_k) + w_k \tag{28}$$

$$y_k = h(x_k) + v_k \tag{29}$$

Step 1: Let X_k $(n \times (2n+1))$ be a matrix of sigma points. *n* is the number of state variables in *f* and each column in X_k is an individual state. To initialize the system:

$$\lambda = \alpha^2(n + \kappa) - n \tag{30}$$

α determines the spread of the sigma points while κ is a secondary scaling parameter. Let W_k $(1 \times (2n+1))$ be the row vector of sigma point weights.

The sigma points for the latest state estimate x_k are now calculated. The column index of the matrices X_k and $\sqrt{(n+\lambda)P_k}$ are indicated in square brackets.

$$X_k[i] = x_k + \left(\sqrt{(n+\lambda)P_k}\right)[i], for\ i = 1\ to\ n \tag{31}$$

$$X_k[i+n] = x_k - \left(\sqrt{(n+\lambda)P_k}\right)[i], for\ i = 1\ to\ n \tag{32}$$

$$X_k[0] = x_k \tag{33}$$

Each sigma point is now propagated through the non-linear process model *f.* Their weights are calculated in Equation 35 and 36. This is used to calculate the predicted state $x_{k+1|k}$ and covariance of the prediction.

$$X_{k+1|k}[i] = f(X_k[i])\ for\ i = 0\ to\ 2n \tag{34}$$

$$Y_{k+1|k}[i] = h\big(X_{k+1|k}[i]\big) \tag{35}$$

$$W_k[0] = \frac{\lambda}{n+\lambda} \tag{36}$$

$$W_k[i] = \frac{1}{2(n+\lambda)} \tag{37}$$

$$\widehat{x_{k+1|k}} = \sum_{i=0}^{2n} W_k[i] X_{k+1|k}[i] \tag{38}$$

$$P_{k+1|k} = \sum_{i=0}^{2n} W_k[i]\big(X_{k+1|k}[i] - \widehat{x_{k+1|k}}\big)\big(X_{k+1|k}[i] - \widehat{x_{k+1|k}}\big)^T + Q_k \tag{39}$$

Step 2: The measured signal is now used to calculate the innovation, innovation covariance *C* and the cross-covariance between the process and measurement B. This is then used to calculate the Kalman gain G.

$$\Delta_{k+1} = y_{k+1} - h\big(\widehat{x_{k+1|k}}\big) = y_{k+1} - \widehat{y_{k+1|k}} \tag{40}$$

$$C_{k+1} = \sum_{i=0}^{2n} W_k[i]\big(Y_{k+1|k}[i] - \widehat{y_{k+1|k}}\big)\big(Y_{k+1|k}[i] - \widehat{y_{k+1|k}}\big)^T + R_k \tag{41}$$

$$B_{k+1} = \sum_{i=0}^{2n} W_k[i]\,\big(X_{k+1|k}[i] - \widehat{x_{k+1|k}}\big)\big(Y_{k+1|k}[i] - \widehat{y_{k+1|k}}\big)^T \tag{42}$$

$$G_{k+1} = B_{k+1} C_{k+1}^{-1} \tag{43}$$

Step 3: Finally, the filtered state estimate and error covariance are calculated.

$$\widehat{x_{k+1|k+1}} = \widehat{x_{k+1|k}} + G_{k+1}\Delta_{k+1} \tag{44}$$

$$P_{k+1|k+1} = P_{k+1|k} + G_{k+1} C_{k+1} G_{k+1}^T \tag{45}$$

The matrix $\left(\sqrt{(n+\lambda)P_k}\right)$ is calculated through the Cholesky decomposition.

Like the EKF, the UKF can be extended to simultaneous state and parameter estimation by augmenting the state vector with the parameter vector. Also, innovation covariance matching can be used to arrive at an adaptive filtering scheme as in Equation 26. The filtered estimate of the state vector is now propagated through 6 timesteps to arrive at the 30-minute prediction.

6. Prediction and Evaluation Algorithms

6.1. Introduction

A variety of prediction algorithms were created and tested to identify which perform better at blood-glucose prediction and hypoglycemia prediction. Given the non-linear nature of the physiological model, both extended and unscented Kalman filter was used in the prediction algorithm. These filters were tested both in state-estimation and dual-estimation mode. Adaptive Kalman filtering was also tested given that the nature of process noise is unknown. Lastly, to correct for reduced prediction accuracy during sudden drops in blood glucose, a mixed algorithm was also tested which combined an extended Kalman filter with an unscented filter as well as an infinite impulse response filter. The details of the various algorithms are described in the section below starting with a description of the skeletal filtering and prediction algorithm in section 6.2, followed by the various modifications. A naming system is developed to aid the reader. This is shown in Table 2. This convention will be used to refer to the prediction system from here onwards: <NOISE>_<FILTER>_<MODEL>_<ESTIMATION>_<INPUTS>.

Table 2: Naming scheme

Noise	Filter	Model	Estimation	Inputs
Constant <C>	Linear <KF>	Pallerm <PAL>	State <S>	None <N>
Process <A>	Extended <EKF>	Current <CR>	Dual <SP>	Carb <C>
	Sigma <UKF>			Insulin <B>
	AR <IIR>			

6.2. The Skeletal Prediction Algorithm

For a given subject on a given day, the prediction and evaluation system operate as per the general algorithm given in Table 3. *PH* is the prediction horizon in minutes and *D* is the discretization parameter (5 minutes). vectors are **boldfaced** and zero-indexed in square brackets [].

Table 2: Skeletal prediction algorithm

1. Given timestamp *k* in and state z_k
2. Get measurement **CGM**[k+1], inputs **Carb**[k] **Bolus**[k] and **Basal**[k]
3. *filter_predict*(**CGM**[k+1], **Carb**[k], **Bolus**[k], **Basal**[k])
3.1. Calculate $\widehat{z_{k+1\|k}} = f(z_k, \boldsymbol{Carb}[k], \boldsymbol{Bolus}[k], \boldsymbol{Basal}[k])$ and $P_{k+1\|k}$
3.2. Calculate $\widehat{y_{k+1\|k}} = h\left(\widehat{z_{k+1\|k}}\right)$ and $H_{k+1\|k}$
3.3. Calculate innovation $\Delta_{k+1} = y_{k+1} - \widehat{y_{k+1\|k}}$ and gain G_{k+1}
3.4. Update state estimate $\widehat{z_{k+1\|k+1}} = \widehat{z_{k+1\|k}} + G_{k+1}\Delta_{k+1}$
3.5. Update error covariance $P_{k+1\|k+1} = [I - G_{k+1}H_{k+1}]P_{k+1\|k}$
3.6. Let $z_{current} = \widehat{z_{k+1\|k+1}}$, for m = 0 to *PH/D*
3.6.1. $z_{next} = f(z_{current}, Carb = 0, Bolus = 0, Basal = \boldsymbol{Basal}[\boldsymbol{k}])$
3.6.2. $\boldsymbol{pbg}[m] = h(z_{next})$
3.6.3. $z_{current} = z_{next}$
3.7. *return* ***pbg***

6.2.1. Failure Detection

It was observed that during sudden drops in blood glucose levels occurring at or below 90 mg/dl, the filters do not converge to the new state/parameter estimates quickly enough before the hypoglycemic threshold. To account for this, a correction term is added to the generic scheme for future timesteps that adds the innovation error to the predicted blood glucose as per Equation 46. This error correction is applied to all filtering schemes and added before step 3.6.2.

$$if\ \Delta_{k+1} < -3\ and\ CGM[k+1] < 90, then\ \boldsymbol{pbg}[m] = \boldsymbol{pbg}[m] + \Delta_{k+1} \tag{46}$$

6.2.2. Dual Estimation and Adaptive Noise Covariance

For a simple state estimator, the skeletal prediction algorithm shown in Table 3 remains largely unchanged. Note that there is no parameter to augment, therefore $z_k = x_k$ and $\psi_k = Q_k$. During adaptive estimation, the line $\psi_{k+1} = \alpha\psi_k + (1-\alpha)\left(G_{k+1}\Delta_{k+1}\Delta_{k+1}^T G_{k+1}\right)$ must be added to step 3.5.

For the dual estimation scheme, the insulin dependent glucose utilization rate α_{dep1} is augmented to the state vector as shown in Equation 47.

$$z_k = \begin{bmatrix} x_k \\ \alpha_{dep1} \end{bmatrix}\ and\ z_{k+1|k} = \begin{bmatrix} f(x_k, u_k) \\ \alpha_{dep1} \end{bmatrix} \tag{47}$$

Steps 3.1 through 3.5 provide a general overview of the Kalman filter. To perform Extended Kalman filtering, the steps 3.1 through 3.5 must include Equations 17 through 27. To perform Unscented Kalman filtering, Equations 28 through 45 must be included instead.

6.2.3. Mixed Prediction Algorithm

To achieve a win-win situation across all performance metrics, experiments were also performed with a mixed prediction algorithm which contains a baseline scheme and a failure-mode scheme. The generic algorithm is now run twice at each timestep (once for each filter) and two ***pbg*** vectors are returned. A failure mode identification like the one developed in section 6.2.1 is applied. Equation 48 shows an example in which the baseline scheme is an adaptive dual estimation EKF (dAEKF) and the failure-mode scheme is an infinite impulse response filter (IIR).

$$if\ \boldsymbol{CGM}[k+1] < 90\ and\ \frac{dCGM}{dt}_{k+1} < 0\ then\ use\ \boldsymbol{pbg_{IIR}}\ else\ use\ \boldsymbol{pbg_{dAEKF}} \tag{48}$$

6.3. Evaluation of Predictions

For *PH=30*, *PH/D=6* and index *m=5* denotes the blood glucose prediction 30 minutes (or 6 timesteps) into the future. To evaluate prediction performance at any horizon *m*, the RMSE and MAD can be calculated for the entire day containing ***T****=288* timesteps as shown in Equations 49 and 50. Note, vectors are zero-indexed, therefore filtering begins at timestep *k=1* and $pbg_k[5]$ indicates a prediction for the *(k+6)*th timestep.

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T-(m+1)} (\boldsymbol{pbg_k}[m] - \boldsymbol{CGM}[k+(m+1)])^2} \tag{49}$$

$$MAD = \frac{1}{T}\left(\sum_{k=1}^{T-(m+1)} \left|\frac{(\boldsymbol{pbg_k}[m] - \boldsymbol{CGM}[k+(m+1)])}{\boldsymbol{CGM}[k+(m+1)]}\right|\right) \times 100 \tag{50}$$

The predicted $\boldsymbol{pbg_k}[m]$ and reference blood glucose $\boldsymbol{CGM}[k+(m+1)]$ is also plotted on the CG-EGA and the percentage of total points in each zone are recorded for the entire day.

6.4. Benchmark Models

Palerm et al. developed a 2nd order linear model of glucose metabolism to build a hypoglycemia classifier. The model dynamics are given by Equations 46 and 47 (Palerm & Bequette, 2007). The blood glucose concentration is represented by *g* and its rate of change represented by *d.* Further, the 2nd order derivative of glucose concentration is represented by *f*.

$$\begin{bmatrix} g \\ d \\ f \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g \\ d \\ f \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_k \tag{51}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + v_k \tag{52}$$

The first order AR model used by Sparacino et al. is shown in Equation 13 with higher order terms neglected. Both these models were evaluated on Dataset 1 and the overall results for each complete day given a prediction horizon of 30 minutes is shown in Table 4. The process noise covariance for Equation 46 is kept constant at $\frac{Q}{R} = 0.00125$ as per the methodology of Palerm et al. The 30-minute predictions are compared in Figure 23. Both prediction systems perform well overall with the AR model having a significant edge over the physiological model when detecting hypoglycemia episodes. The AR model

also has a higher overall RMSE for the dataset. Both the models are comparable when evaluating percentage of predictions in zone A.

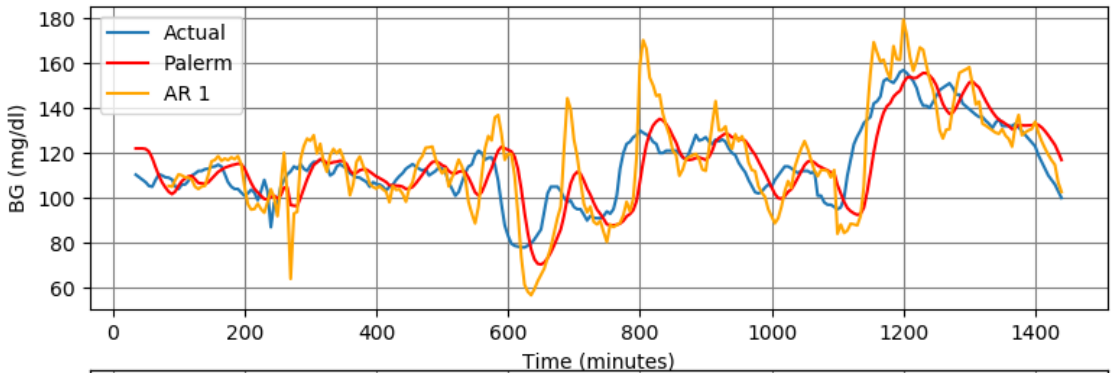

Figure 23: Comparison of 1st order AR and physiological model at PH=30 minutes.

The model developed in this study is evaluated through an Extended Kalman filter. The process noise was kept constant as per the methodology of Palerm et al and failure detection was enabled. A significantly better performance on all evaluation metrics with exception to zone D predictions was observed. This shows that detailed physiological modelling improves prediction accuracy. The drawback of detailed physiological modelling is increased computational cost.

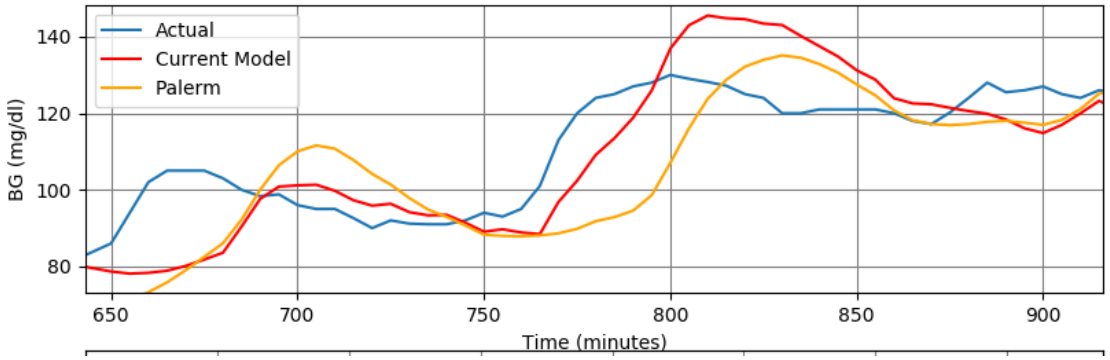

Figure 24: Comparison of physiological models. Current Model=C_EKF_CR_S_CB. Palerm=C_KF_PAL_S_N.

Table 4: Comparison of prediction systems at PH=30 minutes. Overall mean (SD) on Dataset 1.

Model	Zone A %	Zone B%	Zone D%	RMSE
IIR	75.89 (5.65)	23.34 (5.40)	0.23 (0.55)	26.29 (4.20)
C_KF_PAL_S_N	80.74 (5.63)	18.47 (5.20)	0.79 (1.35)	18.75 (3.59)
C_EKF_CR_S_CB	85.22 (6.35)	14 (5.46)	0.77 (1.57)	16.12 (3.56)

Accuracy increases as the prediction horizon is shortened. Figure 25 plots the zone A % for each prediction model with varying prediction horizons.

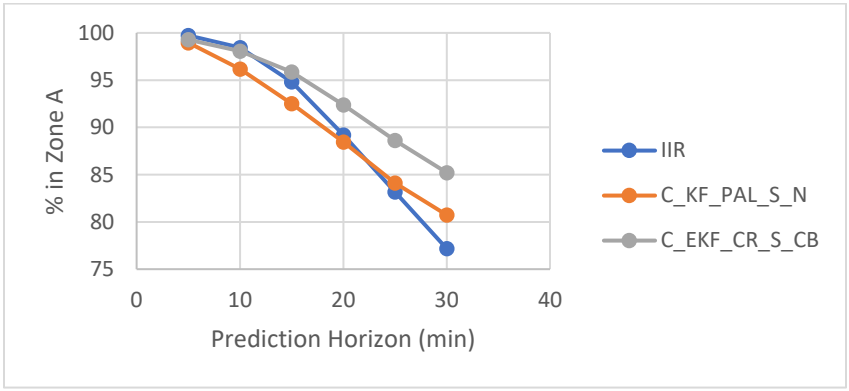

Figure 25: Prediction Accuracy vs. Prediction Horizon for prediction system

7. Discussion of Findings

7.1. Introduction

This study was conducted on two datasets. The first is from a single subject that recorded bolus insulin, basal insulin rate and detailed nutrition intake along with CGM and heart rate readings from a Dexcom G5 device over a 30-day period in free living-conditions. This dataset will be hereby referred to as Dataset 1. The second dataset was compiled by Aleppo et al. for their study on CGM use in free living conditions. This consists of CGM readings of 285 participants along with carbohydrate intake and bolus data collected over 6 months. This data was further processed and divided participant-wise and day-wise. Days that contained missing CGM readings for over 4 hours were discarded. No such restriction was imposed on carbohydrate and bolus information. Therefore, while consistent CGM data was ensured, missing carbohydrate and meal data was allowed. This dataset will be referred to as Dataset 2. For a given subject on a given day, a file containing all the measurement and input vectors discretized to 5-minute intervals was created. The files (in csv file format) are organized with the input and measurement vectors on the columns and timestamps on the rows. An example is provided in Table 2.

Table 5: Measurement and Input vectors.

T (min)	CGM (mg/dl)	Carb (g)	Bolus (u)	Basal (u)	Heart Rate (bpm)
740	133	0	0	0.039583333	65
745	135	0	2.05	0.039583333	65
750	134	0	0	0.039583333	83
755	129	0	0	0.039583333	88
760	127	0	0	0.039583333	77.5
765	129	0	0	0.039583333	73
770	129	32.8	0	0.039583333	73.28571429

7.2. Adaptive Noise Estimation

To implement the adaptive filtering scheme developed in this study, the parameter α required careful tuning. For a prediction horizon of 30 minutes, the parameter was varied to identify the lowest possible overall RMSE on the dataset. Other metrics, such as zone A%, zone D% etc. were found to be unaffected while tuning α. Table 6 shows the results from the α tuning experiment.

Table 3: Tuning α for AEKF estimation

α tuning A_EKF_CR_S_CB					
ALPHA	Zone A %	Zone B%	Zone C%	Zone D%	RMSE
0.1	85.14	14	0	0.85	16.16
0.2	85.16	13.97	0	0.85	16.11
0.3	85.21	13.93	0	0.85	16.05
0.4	85.23	13.91	0	0.85	16.01
0.5	85.25	13.89	0	0.85	15.98
0.6	85.3	13.84	0	0.85	15.95
0.7	85.32	13.82	0	0.85	15.94
0.8	85.39	13.77	0	0.83	15.94
0.9	**85.41**	**13.77**	**0**	**0.8**	**15.95**

To ensure Kalman filter consistency at the selected α, an additional check was introduced. The Innovation Magnitude Bound Test checks that Innovation Δ is unbiased and 95% of its values lie within 2σ. For a given day, the 30-minute predictions from A_EKF_CD_S_CB algorithm is plotted along with the carbohydrate and insulin inputs and the innovation from the filter in Figure 26.

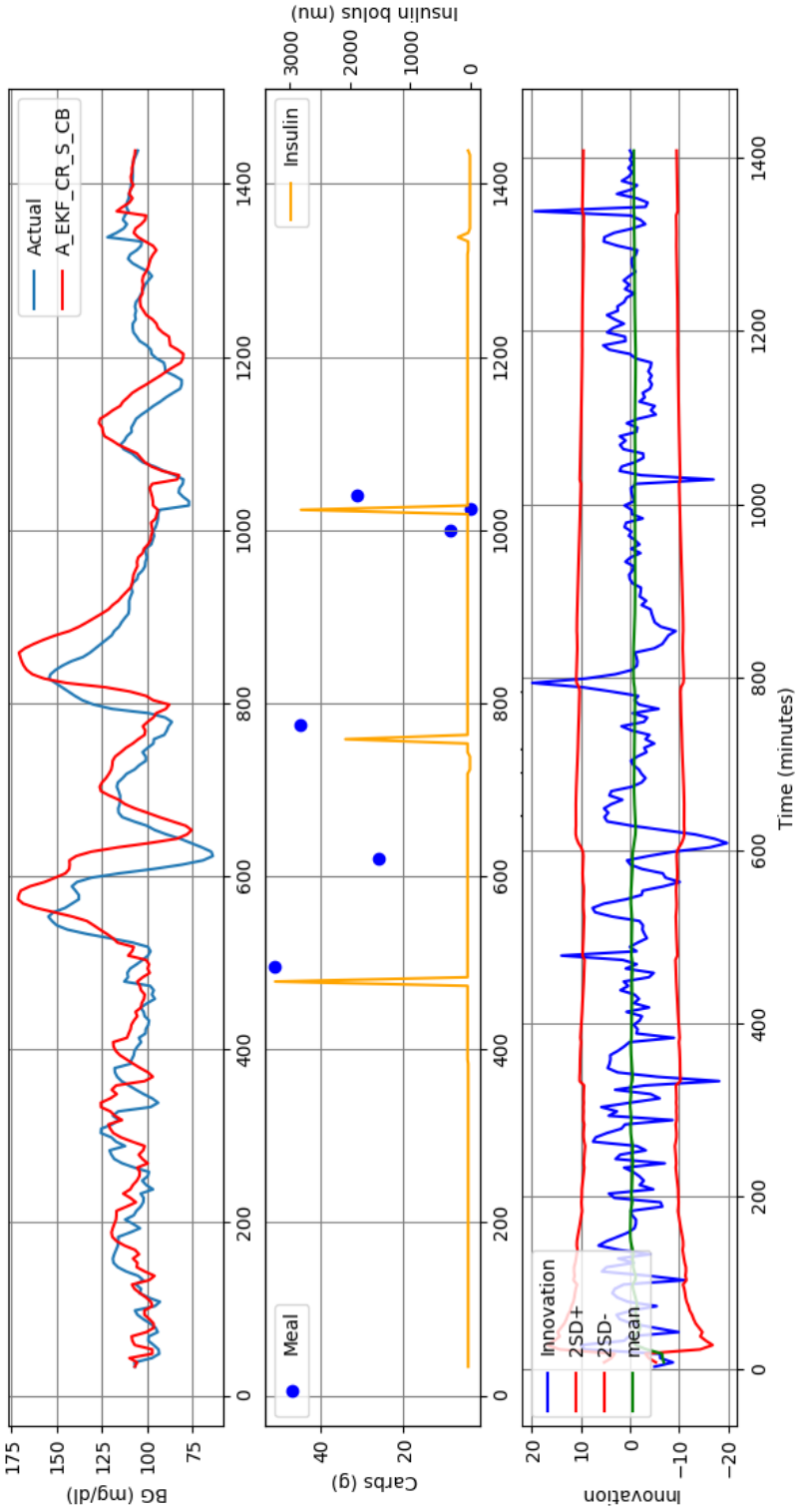

Figure 26; Filter performance evaluation

7.3. Dual Estimation

As was evident from Figure 26, the prediction system does not adapt to rapid drops in blood glucose. The innovation just preceding hypoglycemia episodes is outside the 2σ bounds. The filter fails to perform effectively as the underlying model fails to capture the system dynamics. In order to capture this behavior, the parameter α_{dep1} was varied as it captures the rate of blood glucose depletion due to physical activity. The new dual estimation and prediction scheme A_EKF_CR_SP_CB captures the hypoglycemic episodes better (as seen by reduction in zone D%). This is the ideal mix between the smoothness of physiological model and accuracy of an AR model. Evaluating the scheme on Dataset 1 for α=0.75 yields the comparison shown in Table 7.

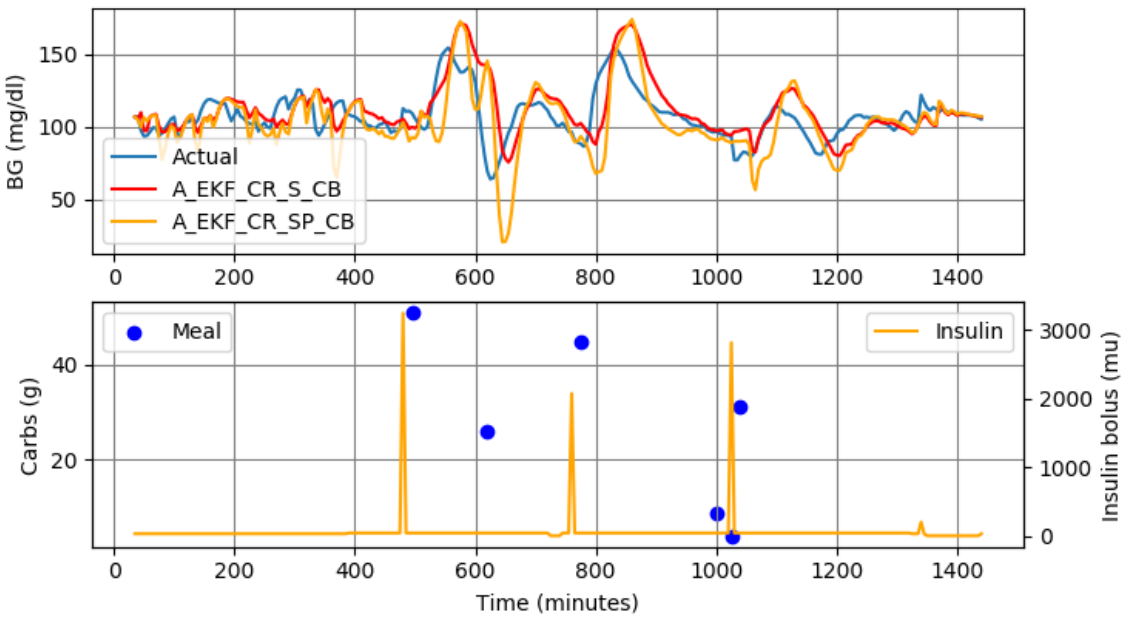

Figure 27: A comparison between state-estimation and dual-estimation prediction system at PH=30 minutes.

Table 4: Overall performance of various prediction schemes on Dataset 1 at PH=30 minutes.

Scheme	Zone A%	Zone B %	Zone C %	Zone D %	Zone E%	RMSE
C_KF_PAL_S_N	80.74	18.47	0	0.79	0	18.75
C_EKF_CR_S_CB	85.22	14.00	0.01	0.77	0	16.12
A_EKF_CR_S_CB	85.41	13.78	0.81	0.81	0	15.96
A_EKF_CR_SP_CB	83.06	16.61	0.02	0.29	0.01	18.28
C_EKF_CR_SP_CB	82.39	17.34	0.00	0.28	0.00	18.80
IIR	75.89	23.34	0.51	0.23	0.03	26.29

The CG_EGA for the A_EKF_CR_SP_CB scheme applied on the worst performing day in dataset 1 is shown in Figure 28 for a prediction horizon of 30 minutes.

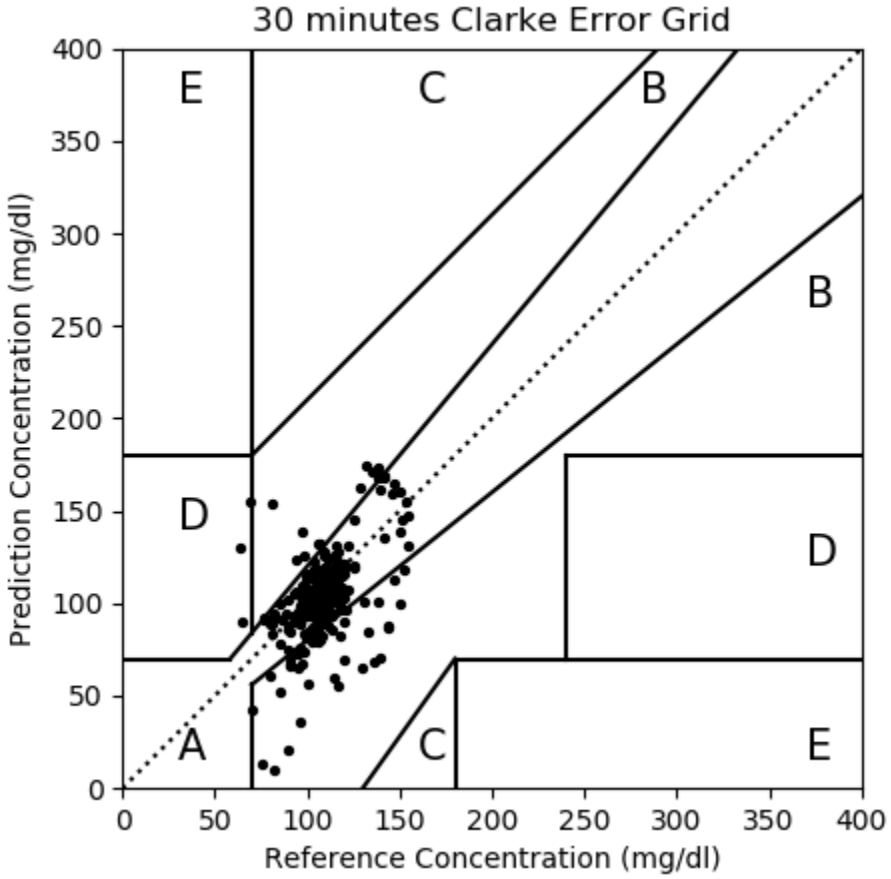

Figure 28: CG_EGA for a 30-minute prediction horizon - A_EKF_CR_SP_CB

7.4. Experiments with the Unscented Kalman filter

The Unscented Kalman filter was expected to perform better as it does not make gaussian approximations to noise. A comparison between a C_UKF_CR_S_CB and C_EKF_CR_S_CB shows minimal differences on most metrics but a slight improvement in zone D predictions for a 30-minute prediction horizon. A Q/R of 5 was found to be the necessary condition for the C_UKF_CR_S_CB scheme to pass the consistency tests. A comparison between the two (for a Q/R=5 in C_EKF_CR_S_CB) is shown in Figure 29.

An adaptive scheme is also attempted for an α of 0.9 however, this led to a decrease in performance in zone D predictions with a slight improvement in RMSE compared to the A_EKF_CR_S_CB scheme, the A_UKF_CR_S_CB scheme performed better on zone D predictions when both were initialized with the Q/R=1 and α=0.9. A comparison between the algorithms is presented in Table 8. The UKF scheme was also applied to simultaneous state and parameter estimation to gauge its effectiveness in reducing zone D errors. Both the A_UKF_CR_SP_CB and C_UKF_CR_SP_CB schemes were found to perform worse.

Table 5: A comparison between EKF and UKF schemes on Dataset 1 at α=0.9 and PH=30 minutes.

Scheme	Zone A%	Zone B %	Zone C %	Zone D %	Zone E%	RMSE
C_EKF_CR_S_CB	85.22	14.00	0.01	0.77	0	16.12
C_UKF_CR_S_CB	85.57	13.69	0.65	0.65	0.00	17.12
A_EKF_CR_S_CB	85.41	13.78	0.00	0.81	0.00	15.96
A_UKF_CR_S_CB	84.79	14.52	0.00	0.69	0.00	16.12
C_UKF_CR_SP_CB	82.88	16.16	0.00	0.96	0.00	17.05
A_UKF_CR_SP_CB	83.52	15.57	0.00	0.91	0.00	16.95

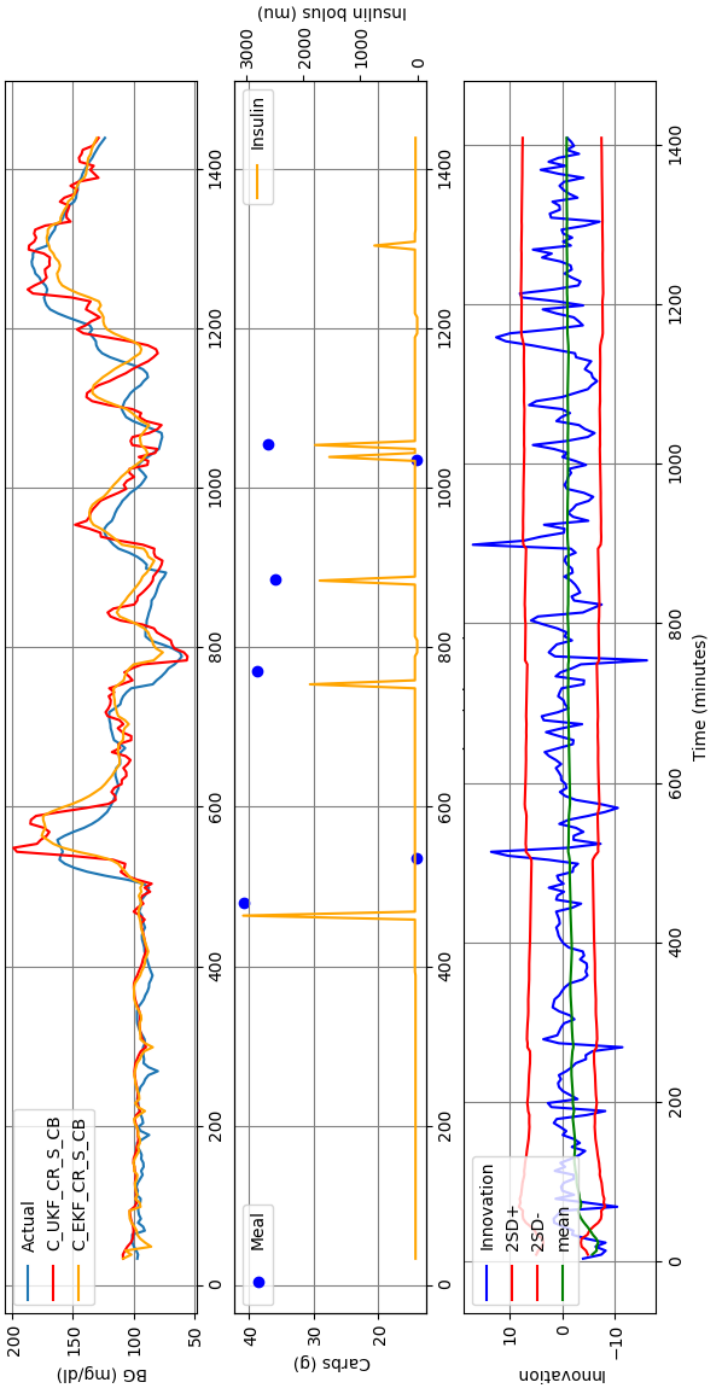

Figure 29: A comparison between the UKF AND EKF schemes at α=0.9 and PH=30 minutes

7.5. Experiments with a Mixed Algorithm

As evident from the sections above, no single scheme performs satisfactorily on all metrics. Adaptive noise schemes perform better in all cases when looking at RMSE results. Dual estimations schemes perform better on the zone D metric at the cost of other metrics. UKF schemes only perform better at state estimation. For the first experiment, the C_UKF_CR_S_CB scheme is taken as the baseline predictor and the C_EKF_CR_SP_CB scheme is taken as the failure-mode predictor. To better predict hypoglycemia episodes, a second mixed algorithm with a baseline C_EKF_CR_SP_CB scheme and an IIR scheme with a forgetting factor of 2 as the failure-mode was also tried. This algorithm is referred to as C_MIXED_IIR_EKF.

Table 6: A comparison of the mixed scheme on Dataset 1 at PH=30 minutes

Scheme	Zone A%	Zone B %	Zone C %	Zone D %	Zone E%	RMSE
C_EKF_CR_SP_CB	82.39	17.34	0.00	0.28	0.00	18.8
C_UKF_CR_S_CB	85.57	13.69	0.0.9	0.65	0.00	17.12
C_MIXED_CR_SP_CB	84.45	15.09	0.09	0.37	0.00	17.93
C_MIXED_IIR_EKF	83.70	16.05	0.00	0.23	0.01	18.39

The C_MIXED_CR_SP_CB scheme performs better on all metrics with exception to zone D when compared to C_EKF_CR_SP_CB. When compared to the C_UKF_CR_S_CB, the mixed scheme performs better only on the zone D metric. A comparison is now performed with the baseline schemes C_KF_PAL_S_N and IIR. The p-values for each performance metric are presented in Table 11. The null hypothesis being that the mixed scheme is not significantly better than IIR and C_KF_PAL_S_N. The C_MIXED_CR_SP_CB scheme does not perform significantly better than IIR or C_KF_PAL_S_N schemes with exception to benign errors (zone B) predictions. A

comparison between the schemes a different prediction horizon on dataset 1 is shown in Table 12. A comparison of the mixed scheme and C_EKF_CR_SP_CB at a prediction horizon of 30 minutes is provided in Figure 30. The C_MIXED_IIR_EKF shows the ideal tradeoff between RMSE of EKF/UKF schemes and the zone D prediction potential of the IIR schemes. Moreover, it also has reduced computational overhead in comparison to the C_MIXED_CR_SP_CB scheme.

Table 7: Testing the significance of results with the C_MIXED_EKF_IIR scheme on Dataset 1 at PH=30.

Scheme	Zone A%	Zone B %	Zone D %	RMSE
IIR	0.38	4.65e-6	0.99	1.8e-5
C_KF_PAL_N	0.99	0.45	0.24	0.985

When comparing overall performance, both the mixed schemes perform well similarly at prediction horizons of 25 minutes or less. The C_MIXED_EKF_IIR scheme is also more consistent at prediction horizons of 15 minutes or less.

Table 8: A comparison of zone A performance at different prediction horizons- mean (SD).

	PREDICTION HORIZON (minutes)					
	30	**25**	**20**	**15**	**10**	**5**
C_MIXED_CR_SP_CB	84.45 (3.73)	87.73 (3.49)	91.97 (2.92)	95.58 (2.20)	98.10 (1.40)	99.55 (0.45)
C_KF_PAL_N	80.74 (5.00)	84.12 (5.57)	88.47 (4.49)	92.52 (3.85)	96.18 (2.52)	98.97 (1.03)
IIR	75.89 (5.65)	83.19 (4.33)	89.21 (3.66)	94.81 (2.78)	98.43 (1.30)	99.73 (1.30)
C_MIXED_EKF_IIR	83.7 (4.8)	87.78 (4.18)	91.75 (3.06)	95.47 (2.08)	98.48 (1.27)	99.77 (0.23)

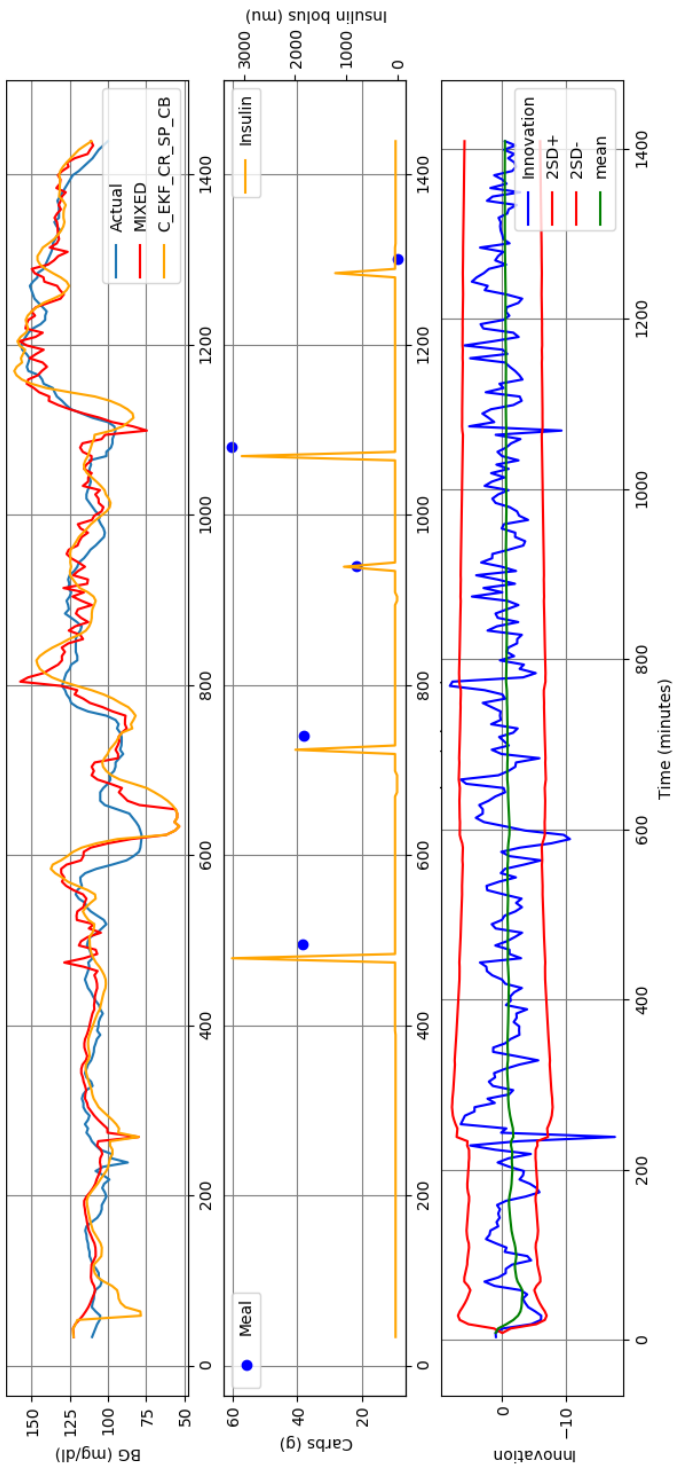

Figure 30: A comparison between the MIXED and C_EKF_CR_SP_CB schemes at a PH=30 minutes.

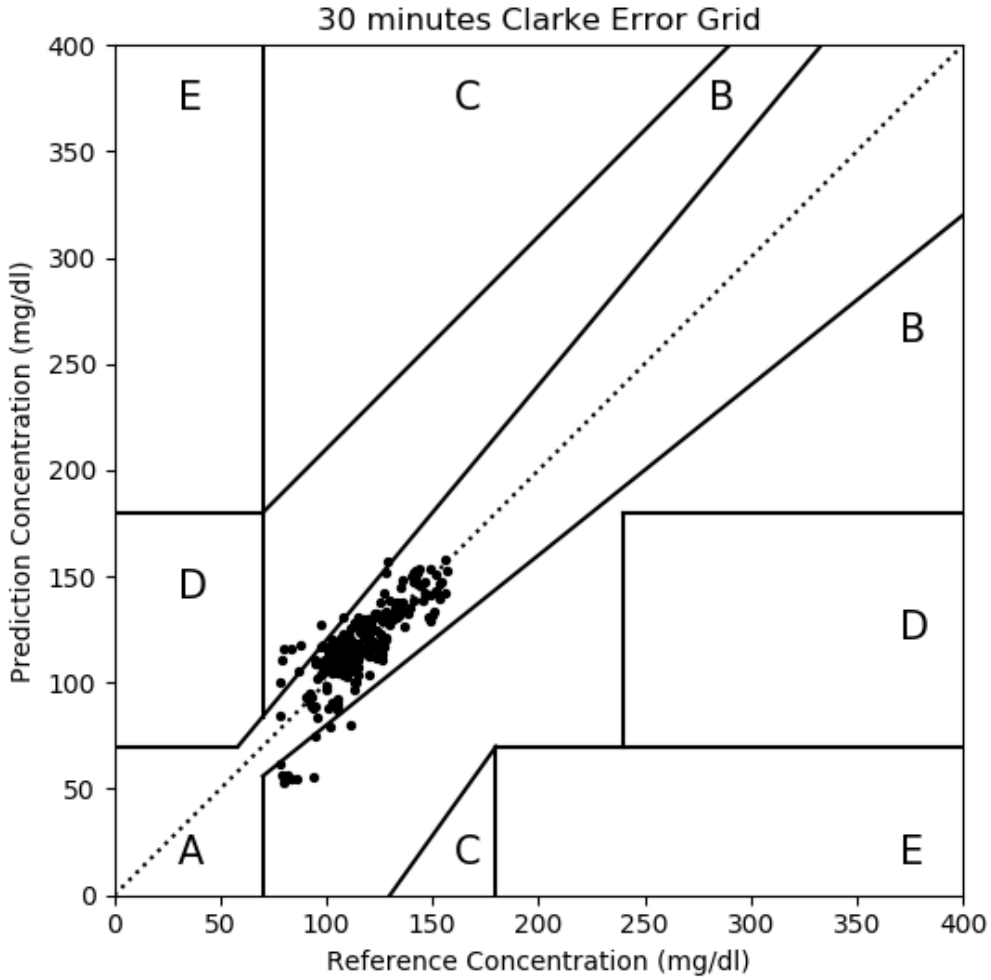

Figure 31: The CEGA for a single day on the mixed scheme at PH=30 minutes

7.6. Influence of Carbohydrate and Insulin Information.

The non-linear model relies on basal, bolus and meal input. Experiments were performed to identify the impact of these variables on the prediction accuracy.

1. Exclude Carb but include Insulin C_MIXED_CR_SP_B.

2. Exclude Insulin but include Carb C_MIXED_CR_SP_C.

3. Exclude both C_MIXED_CR_SP_N.

The results of the experiments are shown in Table 13. The scheme performed expectedly worse when no inputs were provided. When only the meal inputs were ignored, the zone D performance improved slightly. When basal insulin inputs were ignored, the scheme performed slightly better on all metrics with exception to zone D. While the trends in results were not surprising, overall, when only one input was ignored, the performance degraded by an insignificant amount. Exclusion of bolus inputs caused the UKF scheme to diverge. This exposes a weakness in the mixed scheme implementation. Further experiments relevant to this topic are covered in section 7.7.

Table 9: Comparison of performance with various inputs on Dataset 1. PH=30 minutes

Scheme	Zone A%	Zone B %	Zone C %	Zone D %	Zone E%	RMSE
C_MIXED_CR_SP_CB	84.45	15.09	0.09	0.37	0.00	17.93
C_MIXED_CR_SP_B	83.6	16.02	0.05	0.33	0.00	18.45
C_MIXED_CR_SP_C	83.87	15.59	0.04	0.38	0.01	17.78
C_MIXED_CR_SP_N	81.38	18.15	0.00	0.46	0.00	18.49

7.7. Influence of Heart rate on the prediction system

Dataset 1 contains heart rate information discretized to a 5-minute interval. To incorporate this input, the system Equation for insulin dependent utilization was modified to the form shown in Equation 50.

$$\Delta_{dep,k+1} = -I_{c,k}\alpha_{dep1}.\frac{HR}{baseline} + \left(\frac{gm_k}{2.2bm} + \alpha_{dep2}\right) \quad (50)$$

The baseline heartrate was kept constant at 70 bpm. The prediction algorithm also assumes that heart rate remains constant during the prediction horizon. A comparison is

provided in Table 14. Except for a slightly lower RMSE, the inclusion of heart rate in the prediction scheme did not have any significant effect.

Table 10: A comparison of performance with heart rate in a mixed scheme. PH=30 minutes

Scheme	Zone A%	Zone B %	Zone C %	Zone D %	Zone E%	RMSE
C_MIXED_CR_SP_CB	84.45	15.09	0.09	0.37	0.00	17.93
C_MIXED_CR_SP_CBH	84.05	15.42	0.08	0.43	0.01	17.83

7.8. Experiments on Dataset 2

Dataset 2 contains a larger pool subjects in free-living conditions. The dataset does contain missing bolus and carbohydrate information. The inputs were determined to be Missing Completely at Random and thus suiTable entries must be filtered out to ensure a balanced comparison between the schemes. The filtering algorithm sorted the data based on the following two criteria:

1. Continuous CGM data was available from 08:00 to 23:00.

2. At least 2 meal and insulin inputs were recorded.

A total of 37,436 candidate days were identified based on condition 1 alone. This was further filtered down to 15,263 candidate days after applying condition 2. 1000 days were selected at random to perform the experiments. The results are shown in Table 15.

Table 11; Performance of various schemes on Dataset 2 at PH=30 minutes. Mean (SD).

Scheme	Zone A%	Zone B %	Zone D %	RMSE	MAD %
IIR	74.35 (13.90)	23.40 (11.83)	1.04 (1.72)	31.07 (13.14)	15.82 (6.61)
C_EKF_CR_SP_CB	78.06 (11.08)	20.22 (9.86)	1.31 (1.73)	28.79 (10.60)	14.27 (4.70)
A_EKF_CR_SP_CB	77.21 (12.74)	20.46 (10.52)	1.62 (2.80)	29.95 (15.78)	14.71 (6.01)
C_MIXED_EKF_IIR	78.15 (11.28)	20.16 (10.13)	1.32 (1.8)	28.26 (10.53)	14.05 (4.65)
C_KF_PAL_SP_CB	71.95 (14.80)	25.25 (12.91)	2.53 (2.89)	29.11 (11.52)	16.18 (6.00)

Overall, all the prediction schemes performed lower on dataset 2 than on dataset 1. Moreover, the C_KF_PAL_SP_CB performed worse. The IIR scheme still boasts the best hypoglycemia prediction potential. However, the high overall RMSE makes it unsuiTable as an aid in bolus calculations. The C_EKF_CR_SP_CB scheme performed best on all metrics with exception to zone D predictions. The improvement in performance over the C_KF_PAL_SP_CB scheme shows the utility of patient specific modelling. The adaptive scheme performed similar relative to other schemes. The C_MIXED_EKF_IIR scheme showed the best overall performance.

Table 12: Tests of statistical significance on Dataset 2. PH=30 minutes.

Scheme	Zone A%	RMSE	MAD %
IIR	1	2.19e-22	1
C_EKF_CR_SP_CB	1	1	1
A_EKF_CR_SP_CB	4.53e-05	3.08e-14	1
C_KF_PAL_SP_CB	2.65e-30	1	1

8. SweetSpot

8.1. Introduction

The algorithm has been adapted to incorporate real-time visualization of data and predictions as shown in Figure 32. User-Interface testing and visualization was first performed in MATLAB. Both Medtronic and Dexcom have device specific applications on smartphones shown in Figure 33. To develop a similar application would impose costs on the developer to manage separate codebase's for Android, iOS and Windows. A reasonable approach would be to develop a web application capable of running in a web browser which handles device compatibility.

The web application was developed using the Django framework for Python. Django is a high-level web API that is optimized for speed even on handheld phones. It automatically manages scaling and graphics vectors based on device version. To improve security, the website can be hosted behind the user's own personal firewall. Off the shelf wireless LAN routers already have this functionality. An internet-enabled monitoring system also enables the user's treatment specialists and caregivers to monitor their BG in real time. The web service can be easily installed on a raspberry pi through the Apache framework unlike existing systems like Nightscout (NightScout, 2019) which require intermediate to advanced skill to setup on a 3rd party host.

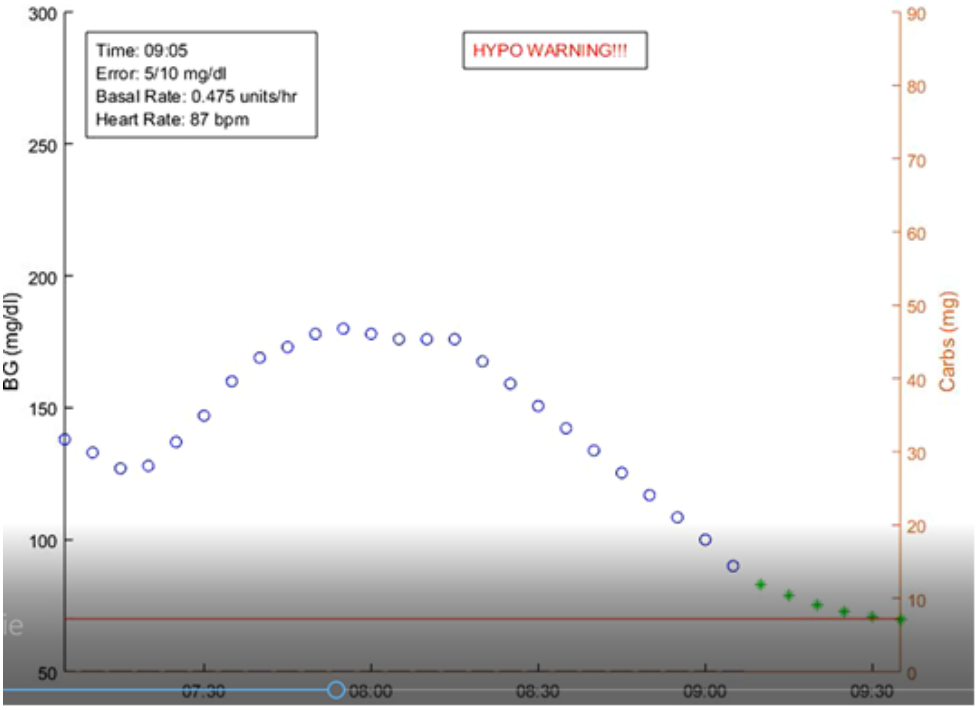

Figure 32: MATLAB UI testbed

Figure 33: handheld CGM apps developed by Medtronic and Dexcom

Figure 34 provides an overview of how the prediction framework functions. Figure 35 provides an overview of code organization and resource management for the SweetSpot website. Sections marked in blue run in their own virtual environment. Subject data is stored in a MySQLite database. The scripts to download Dexcom-share data and Kalman filtering run asynchronously.

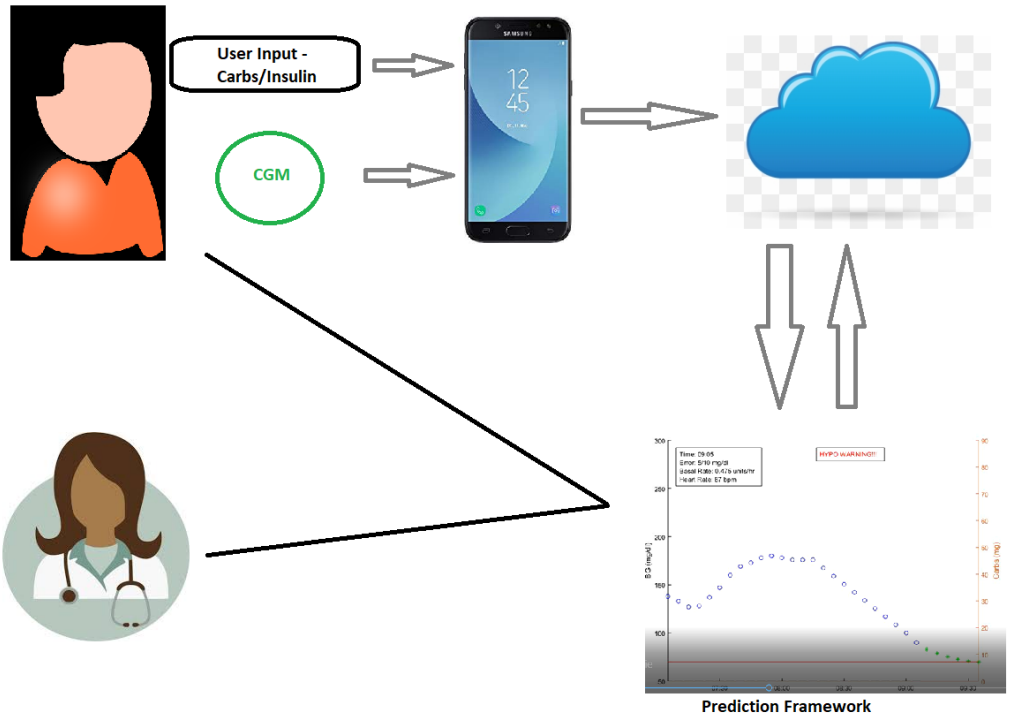

Figure 34: An overview of the prediction framework.

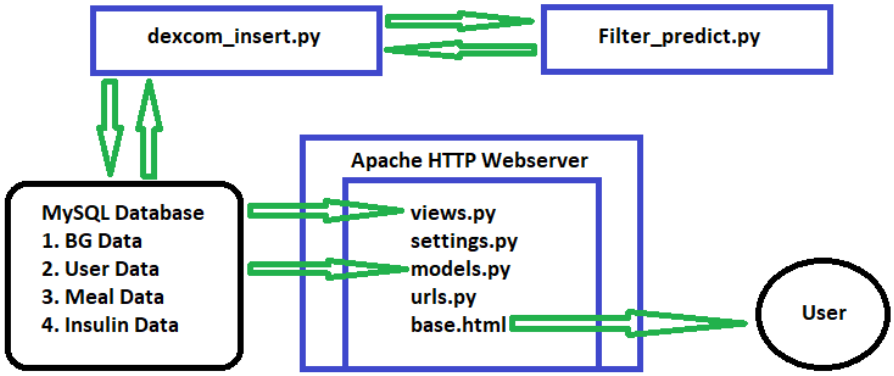

Figure 35: Code Organization for the SweetSpot system in the Django framework.

8.2. Functionality

Each user is expected to setup the web service on a cloud based virtual linux machine or personal computer. To aid in installation, a bash script is provided which installs the source code, apache server and wsgi on the host machine. This leaves the user to only conFigure the Dexcom settings and SSL certificates which is trivial. During installation, the installer guides the user to setup the administrator account for the website. Once completed, the service is active on the <domain name>/SweetSpot address and the user is greeted with the login screen as shown in Figure 36.

Figure 36: SweetSpot login screen.

Once logged in, the user can view the last 2 hours of BG behavior on the home page. Predictions are non-shaded circles. BG values above 180 appear orange and below 70 (hypoglycemia) appear as red. The main page is shown in Figure 37.

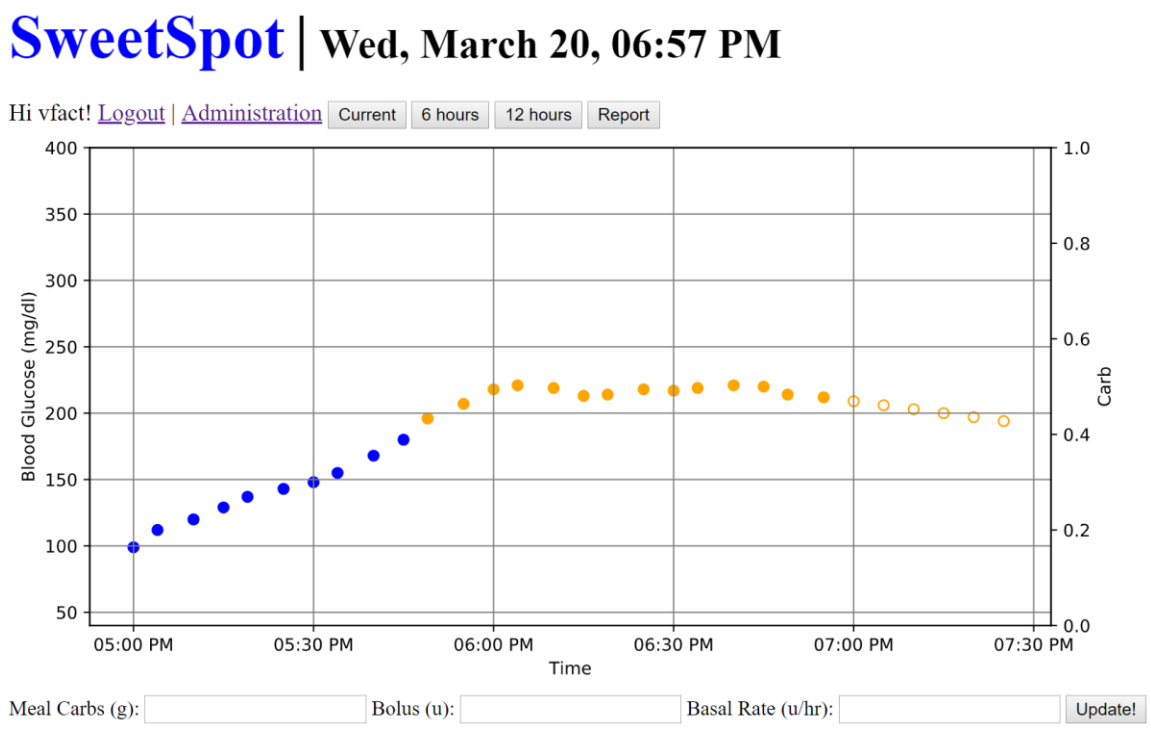

Figure 37: SweetSpot home page.

The administrator page shown in Figure 38 allows the user to enter bodyweight, time zone and Dexcom settings. Figure 39 shows the Report page which provides a pie chart overview of the number of points in the Clarke error grid over past 12 hours. It also plots a bar chart plotting blood glucose levels across the two clinical levels 90, and 150. The user can enter meal and bolus details into the input form below the page.

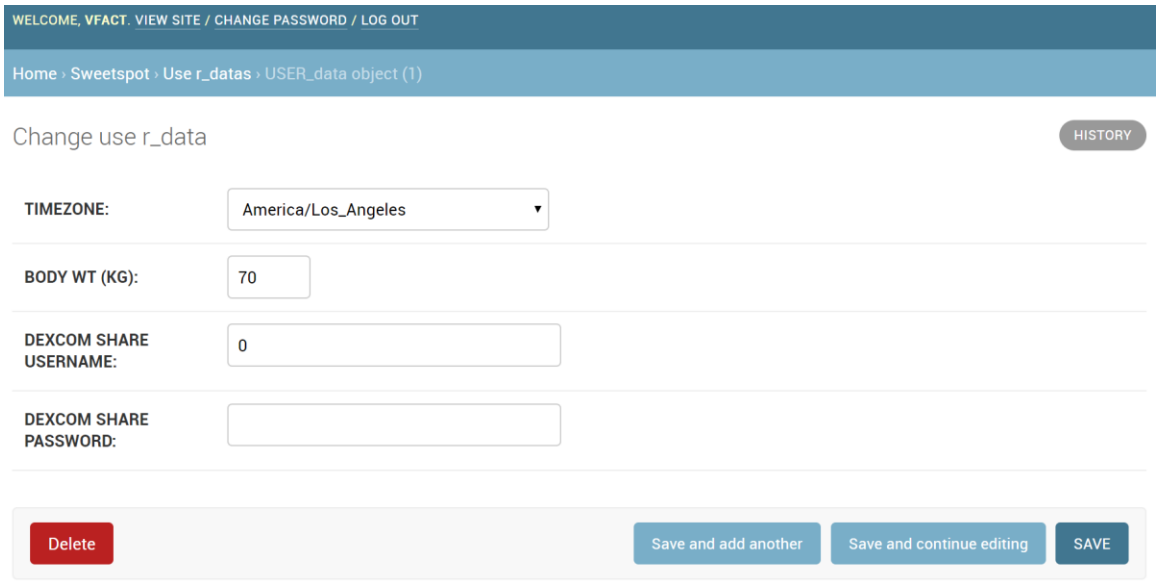

Figure 38: SweetSpot admin page.

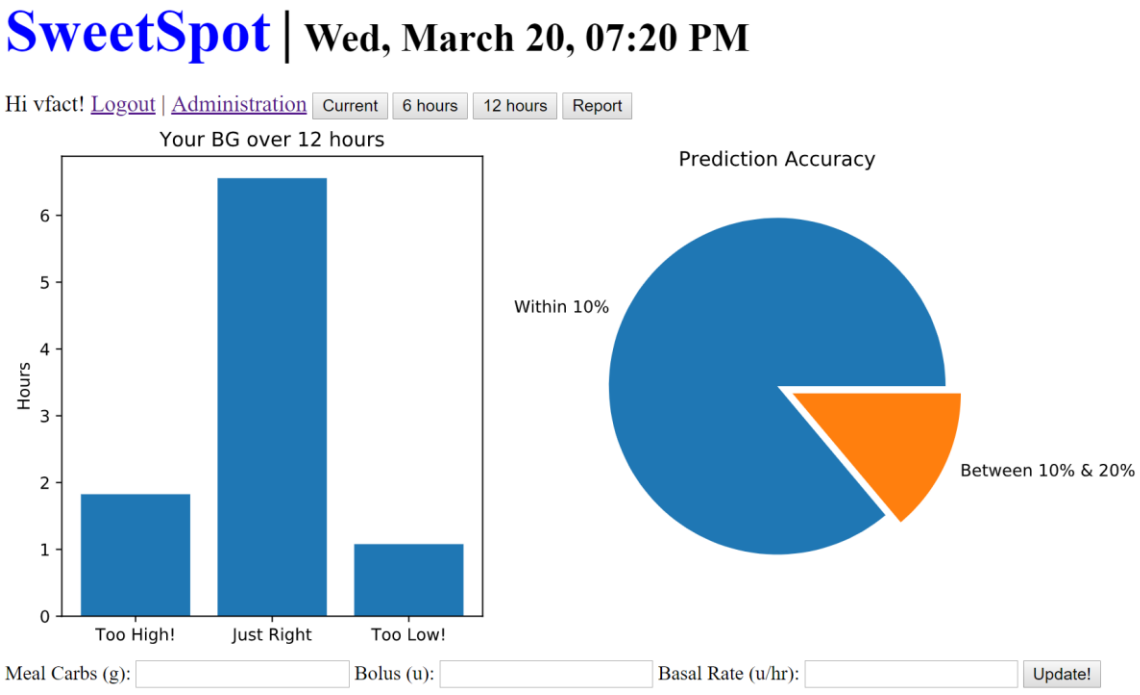

Figure 39: SweetSpot Report page.

The system also allows the user to receive email alerts to selected users at pre-determined thresholds. Moreover, the input forms are only available to the user/administrator of the site.

9. Concluding Notes

Hypoglycemia prediction from CGM data has seen increased research participation around the world since the last decade. This study attempts to compare existing systems and find possible alternatives in the signal processing domain. the project shows the positive impact of physiological modelling on a prediction scheme and demonstrated that patient specific models perform better than a one-size-fits all scheme. Different filtering strategies, extended, robust, sigma point were attempted to arrive at a scheme that shows improvement across all performance metrics. It was noted, that no single scheme performs on every measure. Compromises must be made depending on the end-user application. An IIR works well if hypoglycemia prediction is the only goal. However patient specific modeling through Kalman filters can yield better long-term insight into how behavior and bolus patterns affect glycemic control. The utility of sigma-point filters to this application was also noted. Moreover, having multiple datasets of varying veracity shows how robust the prediction schemes are and whether relative performance is consistent.

SweetSpot can be used as a data-gathering and performance evaluation tool given its scalability. The primary roadblock in developing machine learning algorithms is the lack of accurate data.

Finally, I must acknowledge the role and support of my advisor Dr. Thomas Stahovich through my graduate school journey at UC-Riverside. I also thank my colleagues working alongside at the Smart Tools Lab – Kevin Rawson, Amirali Darvisadeh and Justin Gyllen for support on navigating the grad-school life and some memorable moments.

10. References

Aleppo, G., Ruedy, K. J., Riddlesworth, T. D., Kruger, D. F., Hirsch, I., Peters, A. L., . . . Ahmann, A. J. (2017). REPLACE-BG: A Randomized Trial Comparing Continuous Glucose Monitoring with and Without Routine Blood Glucose Monitoring in Adults With Well-Controlled Type 1 Diabetes. *Diabetes Care*.

Berger, M., & Rodbard, D. (1989). Computer Simulation of Plasma Insulin and Glucose Dynamics After Subcutaneous Insulin Injection. *Diabetes Care*.

Chen, C., Zhao, X.-L., Hong, Z., Zhu, Z.-G., & Qian, S.-H. (2017). Current and Emerging Technology for Continuous Glucose Monitoring. *Sensors*.

Clarke, W. L., Frederick, L. A., Carter, W., & Pohl, S. L. (1987). Evaluating Clinical Accuracy of Systems for Self-Monitoring of Blood Glucose. *Diabetes Care*.

David Rodbard, M. (2016). *Continuous Glucose Monitoring: A review of Successes, Challenges, and Opportunities.* Diabetes Technology & Therapeutics.

Deutsh, T., & Lehmann, E. (1996). Computer asssited diabetes care: a 6-year retrospective. *Computer Methods and Programs in Biomedicine*.

Diabetes Forecast. (2014, May). *Anatomy of a CGM Sensor*. Retrieved from Diabetes Forecast: http://www.diabetesforecast.org/2014/05-may/anatomy-of-a-cgm-sensor.html

diaTribe Learn. (2018, March 3). *When You Don't Knoe You're Low - Hypoglycemia Unawareness 101*. Retrieved from diatribe.org: https://diatribe.org/hypoglycemia-unawareness-101

Dubosson, F., Beatriz, L., & Schumacher, M. (2017). Negative results for the prediction of postprandial hypoglycemia from insulin intakes and carbohydrates: analysis and comparison with simulated data. *Proceedings of the Second Workshop on Artificial Intelligence for Diabetes.* Vienna, Austria.

Duke, D. L. (2010). *Intelligent Diabetes Assistant.*

Facchinetti, A., Sparacino, G., & Cobelli, C. (2009). An Online Self-Tunable Method to Denoise CGM Sensor Data. *IEEE transactions on Biomedical Engineering*.

Geaghan, S. (2014). Infection transmission associated with point of care testing and the laboratory's role in risk reduction. *The Journal of the International Federation of Clinical Chemistry and Laboratory Medicine*.

Group, D. R. (1993). The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus. *The New England Journal of Medicine*.

Guerra, S., Facchinetti, A., Sparacino, G., & Cobelli, C. (2012). Enhancing the Accuracy of Subcutaneous Glucose Sensors: A Real-Time Deconvolution-Based Approach. *IEEE Transactions on Biomedical Engineering*.

health24. (2017, February 21). *A finger-stick free future for diabetics?* Retrieved from health24 Diabetes: https://www.health24.com/Medical/Diabetes/Technology/A-finger-stick-free-future-for-diabetics-20150317

Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory.* Upper Saddle River: Prentice Hall.

Klonoff, D. C. (2005). Continuous Glucose Monitoring: Roadmap for the 21st century diabetes therapy. *Diabetes Care*.

Knobbe, E. J., & Buckingham, B. (2005). The Extended Kalman Filter for Continuous Glucose Monitoring. *Diabetes Technology & Therapeutics*.

Lebech, S., Johansen, M., & Hejlesen, O. (2016). Towards Big Data Analytics: Review of Predictive Models in Management of Diabetes and its Complications. *Journal of Diabetes Science and Technology*.

Lehmann, E., & Deutsch, T. (1992). A physiological model of glucose-insulin interaction in type 1 diabetes mellitus. *Journal of Biomedical Engineering*.

Man, C. D., Camilleri, M., & Cobelli, C. (2006). A System Model of Oral Glucose Absorption: Validation on Glod Standard Data. *IEEE Transactions on Biomedical Engineering*.

Man, C. D., Micheletto, F., Lv, D., Brenton, M., Kovatchev, B., & Cobelli, C. (2014). The UVA/PADOVA Type 1 Diabetes Simulator: New Features. *Journal of Diabetes Science and Technology*.

Mayo Clinic. (2018, September 7). *Hypoglycemia*. Retrieved from mayoclinic.org: https://www.mayoclinic.org/diseases-conditions/hypoglycemia/symptoms-causes/syc-20373685

Merwe, R. v., & Wan, E. (2003). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models.* Beaverton: OGI School of Science & Engineering.

Mougiakakou, S. G., Prountzou, K., & Nikita, S. K. (2005). A Realtime Simulation Model of Glucose-Insulin Metabolism for Type 1 Diabetes Patients. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference.* Shanghai.

National Center for chronic Disease Prevention and Health Promotion. (2017). *National Diabetes Statistics Report.*

NightScout. (2019, February 1). *NightScout #WeAreNotWaiting*. Retrieved from http://www.nightscout.info/

Olansky, L., & Kennedy, L. (2010). Finger-Stick Glucose Monitoring. *Diabetes Care*.

Oruklu, M. E., Cinar, A., Quinn, A., & Smith, D. (2009). Estimation of Future Glucose Concentrations with Subject-Specific Recursive Models. *Diabetes Technology & Therapeutics*.

Palerm, C. C., & Bequette, B. W. (2007). Hypoglycemia Detection and Prediction Using Continuous Glucose Monitoring - A Study on Hypoglycemic Clamp Data. *Journal of Diabetes Science and Technology*.

Pappada, S. M., Cameron, B. D., & Rosman, P. M. (2008). Development of a Neural Network for Prediction of Glucose Concentration in Type 1 Diabetes Patients. *Journal of Diabetes Science and Technology*.

Perez-Gandia, C., Facchinetti, A., Sparacino, G., Cobelli, C., Gomez, E., Rigla, M., . . . Hernando, M. (2010). Artificial Neural Network Algorithm for Online Glucose Prediction from Continuous Glucose Monitoring. *Diabetes Technology & Therapeutics*.

Plis, K., Shubrook, J., & Schwartz, F. (2014). A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes Management. *Modern Artifical Intelligence for Health Analytics*.

Plis, K., Shubrook, J., & Schwartz, F. (2014). abcd. *Modern Artificial Intelligence for Health Analytics*.

Sparacino, G., Zanderigo, F., Corazza, S., Maran, A., Facchinetti, A., & Cobelli, C. (2007). Glucose Concentration can be Predicted Ahead in. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*.

Wang, Q., Molenaar, P., Harsh, S., Freeman, K., Xie, J., Gold, C., . . . Ulbrecht, J. (2014). Personalized State=space Modelling of Glucose Dynamics for Type 1 Diabetes Using Continuously Monitored Glucose, Insulin Dose, and Meal Intake: An Extended Kalman Filter Approach. *Journal of Diabetes Science and Technology*.

Wentholt, I. M., Hoekstra, J. B., & DeVries, J. H. (2006). A Critical Appraisal of the Continuous Glucose-Error Grid Analysis. *Emerging treatments and Technologies*.

Zhu, Z., Gancedo, L., Flewitt, A. J., Xie, H., & Moussy, F. (2012). A Criticla Review of Glucose Biosensors Based on Carbon Nanmaterials: Carbon Nanotube and Graphene. *Sensors*.